١٢

الجزء الأول

بسم الله الرحمن الرحيم

دولة فلسطين

وزارة التربية والتعليم

البرمجة والأتمتة

الفرع التكنولوجي

فريق التّأليف:

أ. إبراهيم قدح (منسقاً)

أ.مهند ابو الهيجا

م.سامي غنام

أ. حسين حمامدة

م. جهاد خلوف

أ.أيمن العكلوك

م. معاذ أبو سليقة

قررت وزارة التربية والتعليم في دولة فلسطين

تدريس هذا الكتاب في مدارسها بدءاً من العام الدراسي ٢٠١٧ / ٢٠١٨ م

الإشراف العام

رئيس لجنة المناهج	د. صبري صيدم
نائب رئيس لجنة المناهج	د. بصري صالح
رئيس مركز المناهج	أ. ثروت زيد
مراجعة تربوية	د. بشار طحاينة

الدائرة الفنية

الإشراف الفني	كمال فحماوي
التصميم الفني	أ.سمر عوض

التحرير اللغوي	أ. وفاء الجيوسي
متابعة المحافظات الجنوبية	د. سميّة النّخالَة

الطبعة الأولى

٢٠١٩ م/ ١٤٤٠ هـ

جميع حقوق الطبع محفوظة ©

دولة فلسطين

وزارة التربية والتعليم

مركز المناهج

حي الماصيون، شارع المعاهد

ص. ب 719 – رام الله – فلسطين

pcdc.mohe@gmail.com | pcdc.edu.ps

mohe.ps | mohe.pna.ps | moehe.gov.ps

.com/MinistryOfEducationWzartAltrbytWaltlym

فاكس +970-2-2983250 | هاتف +970-2-2983280

تـقـديـم

يتصف الإصلاح التربوي بأنه المدخل العقلاني العلمي النابع من ضرورات الحالة، المستند إلى واقعية النشأة، الأمر الذي انعكس على الرؤية الوطنية المطورة للنظام التعليمي الفلسطيني في محاكاة الخصوصية الفلسطينية والاحتياجات الاجتماعية، والعمل على إرساء قيم تعزز مفهوم المواطنة والمشاركة في بناء دولة القانون، من خلال عقد اجتماعي قائم على الحقوق والواجبات، يتفاعل المواطن معها، ويعي تراكيبها وأدواتها، ويسهم في صياغة برنامج إصلاح يحقق الآمال، ويلامس الأماني، ويرنو لتحقيق الغايات والأهداف.

ولما كانت المناهج أداة التربية في تطوير المشهد التربوي، بوصفها علماً له قواعده ومفاهيمه، فقد جاءت ضمن خطة متكاملة عالجت أركان العملية التعليمية التعلمية بجميع جوانبها، بما يسهم في تجاوز تحديات النوعية بكل اقتدار، والإعداد لجيل قادر على مواجهة متطلبات عصر المعرفة، دون التورط بإشكالية التشتت بين العولمة والبحث عن الأصالة والانتماء، والانتقال إلى المشاركة الفاعلة في عالم يكون العيش فيه أكثر إنسانية وعدالة، وينعم بالرفاهية في وطن نحمله ونعظمه.

ومن منطلق الحرص على تجاوز نمطية تلقّي المعرفة، وصولاً لما يجب أن يكون من إنتاجها، وباستحضار واعٍ لعديد المنطلقات التي تحكم رؤيتنا للطالب الذي نريد، وللبنية المعرفية والفكريّة المتوخّاة، جاء تطوير المناهج الفلسطينية وفق رؤية محكومة بإطار قوامه الوصول إلى مجتمع فلسطيني ممتلك للقيم، والعلم، والثقافة، والتكنولوجيا، وتلبية المتطلبات الكفيلة بجعل تحقيق هذه الرؤية حقيقة واقعة، وهو ما كان له ليكون لولا التناغم بين الأهداف والغايات والمنطلقات والمرجعيات، فقد تآلفت وتكاملت؛ ليكون النتاج تعبيراً عن توليفة تحقق المطلوب معرفياً وتربوياً وفكرياً.

ثمّة مرجعيات تؤطّر لهذا التطوير، بما يعزّز أخذ جزئية الكتب المقررّة من المنهاج دورها المأمول في التأسيس؛ لتوازن إبداعي خلّاق بين المطلوب معرفياً، وفكرياً، ووطنياً، وفي هذا الإطار جاءت المرجعيات التي تم الاستناد إليها، وفي طليعتها وثيقة الاستقلال والقانون الأساسي الفلسطيني، بالإضافة إلى وثيقة المنهاج الوطني الأول؛ لتوجّه الجهد، وتعكس ذاتها على مجمل المخرجات.

ومع إنجاز هذه المرحلة من الجهد، يغدو إزجاء الشكر للطواقم العاملة جميعها؛ من فرق التأليف والمراجعة، والتدقيق، والإشراف، والتصميم، وللجنة العليا أقل ما يمكن تقديمه، فقد تجاوزنا مرحلة الحديث عن التطوير، ونحن واثقون من تواصل هذه الحالة من العمل.

وزارة التربية والتعليم

مـركـز الـمـنـاهـج الفلسطينية

آب / ٢٠١٨ م

مـقـدمـة

تسعى وزارة التّربية والتّعليم الفِلسطينية للنّهوض بالعمليّة التعليميّة، ومجاراة الواقِع المهنيّ عالميًا، وفلسطينيًا، مُستندةً إلى متطلبات سوق العمل، ساعيةً لتطويرِ مناهجها؛ لبناء جيلٍ قادر على تحمّل مسؤولياته، والانخراط في سوق العمل بكفاءةٍ وفاعليّة، مَناهِجَ تُبنى على المعرفة العلميّة الصحيحة؛ لسد الحاجة المتزايدة في هذا المجال، وعليه جَاءت فكرة إيجاد فرع جديدٍ للطّلبة الفلسطينيين، وهو الفَرع التّكنولوجي، الذي يغطّي الحاجة الملحّة لِدراسة تكنولوجيا المَعلومات، والاتّصالات، والإلكترونيات، مدعمًا بمبحث الرّيادة في الأعمال، وذلكَ لتزويد الطّلبة وتمكينهم من الأساسيات العلميّة والعمليّة، وقَد تمّ بناء المِنهاج ضمن استراتيجيّة التعليم النّظري، والتّطبيق العَمليّ مدعمًا بالصّور والفيديوهات، وذلك في إطار مواكبة التّطورات العالمية، لتخريج طالبٍ يَستطيع المُنافسة في مجال البَرمجة والاتصالات.

جاءَ الكتاب مقسّمًا إلى خمسِ وحَدات، رأينا أنّها تشكل عونًا أساسيًا للطالب بعد إنهائِه للثانوية العامّة في رسِم وتحديدِ مُستقبله المِهني، سواء بِفتح مشروعه الخاص، أو الاستمرار في المجال العلميّ، والالتحاق بالتّخصص المناسب جامعيًا.

عنونت الوحدة الأولى بقواعد البيانات وبرمجتها، حيث تطرقت الى مفهوم قاعدة البيانات وكيفية تحليلها، وبناء قاعدة بيانات سليمة باستخدام برنامج SQL server ؛ فيما جاءت الوحدة الثانية لتتمة مفهوم البرمجة المتقدمة وعرضت موضوع تراتيب البيانات والبرمجة الهدفية لتكون عوناً للطالب في تحديد اتجاهاته المستقبلية، أما الوحدة الثالثة فحملت عنوان الذكاء الاصطناعي واقعه ومستقبله.

فيما كان الجزء الثاني من الكتاب والذي أتى بوحدتين، أتت الوحدة الأولى بعنوان الوسائط المتعددة لتقدم نظرة عامة لاستخدامات الوسائط المتعددة وتمكين الطلبة من خلال استخدام برامج خاصة لمعالجة الصوت والصورة وإنتاج الفيديو، فيما كانت الوحدة الأخيرة تحت عنوان برمجة المواقع الإلكترونية لتتناول دراسة سوق العمل وحاجته لمثل هذه المواضيع، ثم تطوير الطلبة لإمكانية ليصبح بإمكانهم تصميم موقع الكتروني وإدارة محتوى. تمّ إنجاز هذا الكِتاب بجهودِ فريق عمل متميّز، حيث يعتبر هذا الكِتاب في السّنة الأولى من إصداره نسخةً تجريبيّة بحاجة إلى التنقيح والتّعديل بناء على التّغذية الراجعة التي تردنا من الميدان، لذا نرجو من معلمينا، وأبنائِنا الطّلبة، وإخوتنا أولياءَ الأمور تزويدَنا في الإدارة العامة للمباحث العلميّة - مركز المناهج بكافّة المُلاحظات والاقتراحات؛ من أجل الإثراء، والتّعديل، والتّحديث.

فريق التأليف

المحتويات

قواعد البيانات وبرمجتها

الوحدة الأولى

- **الدرس الأول:** قواعد البيانات ٤
- **الدرس الثاني:** تحليل قواعد البيانات ١٧
- **الدرس الثالث:** بناء قاعدة البيانات ٣٥

البرمجة المتقدمة

الوحدة الثانية

- **الدرس الأول:** تركيب البيانات ٧٤
- **الدرس الثاني:** البرمجة الهدفية ١٠٤

الذكاء الإصطناعي

الوحدة الثالثة

- **الدرس الأول:** الذكاء الاصطناعي ١٣٤

الوحدة الأولى

قواعد البيانات وبرمجتها

أتأمل ثم أُناقش: خطوة نحو اقتصاد المعرفة

أهداف الوحدة

يتوقّع منك بعد دراستك لهذه الوحدة أن تكون قادراً على بناء قاعدة بيانات محوسبة ومتكاملة لتسهيل عمليّة اتّخاذ القرارات، من خلال تحقيق الأهداف الآتية:

١- تعريف المفاهيم الأساسيّة في أنظمة قواعد البيانات وإدارتها.
٢- تحليل الأنظمة المحوسبة لتخزين البيانات.
٣- إجراء عمليّات معالجة مختلفة لقاعدة البيانات ومكوّناتها من خلال لغة (SQL).
٤- بناء نظام قاعدة بيانات بصورة شاملة ومتكاملة.

الدرس (1)

قواعد البيانات (Data Base)

تعدّ البيانات في العصر الحالي المادّة الأساسيّة لبناء المعرفة والمعلومات اللّازمة لصِناعة القرار، وتوثيقاً حقيقيّا للمشاهدات والحقائق والأحداث، وعند تَزايد كمّ البيانات وأنواعها وتسمياتها لابُدَّ من وضعها في بوتقةٍ شاملةٍ لإتاحة الترابط بينها بطريقة يمكّننا الإستفادة منها، ويتمّ الربط بين تلك البيانات والتنسيق ما بينها عن طريق قاعدة بياناتٍ تختلف بُنيتها باختلاف فحوى البيانات الموجودة فيها ونوعها.

يَشهد العالم في القرن الحاليّ تقدّما سريعاً في التّعامل مع قواعد البيانات وإدارتها، من حيث: التّخزين والتّحليل و طرق التصنيف والترتيب والعمليّات عليها، من خلال مبادئ وقواعد نظمت لهذا الغرض، بحيث تضمن الوصول للمعلومات بشكلٍ دقيقٍ وسليم دون لُبس أو تكرار في البيانات المعروضة؛ فقواعد البيانات المحوسبة تملك المرونة اللّازمة في تخزين البيانات واسترجاعها والتّعديل عليها، وهي البديل الأفضل عن الطرق التقليدية في تخزين البيانات خاصّة الورقية منها. فما هي قواعد البيانات؟

قارن بين نظام الملفات القديم (الورقي) والنظام المحوسب في تخزين البيانات والتعامل معها من خلال المعايير الموضحة في الجدول أدناه.

المعيار	النظام الورقي	النظام المحوسب
معالجة البيانات		
استقلالية البرامج		
تكلفة النظام		
سرعة الوصول للمعلومة		

تذكّر أنّ:

- البيانات (Data): تمثل المادة الخام التي يستخرج منها أي معلومة تفيد في اتخاذ القرار.
- المعلومات (Information): مجموعة من البيانات تم تجميعها وتنظيمها ومعالجتها لتعطي دلالات بطرق معينة.

خصائص قواعد البيانات Database Properties

تمتاز قواعد البيانات بخصائص عديدة وضرورية؛ لتصبح أكثر تماسكاً وفاعليّة.
وللتعرُّف إلى تلك الخصائص قم بالنشاط الآتي:

نشاط (١) خصائص قواعد البيانات

نظامٌ جامعيّ مكوّن من عدّة أقسام: الماليّة، المستودع، المكتبة، التسجيل.
الطالب داخل الجامعة له علاقة بأغلب هذه الأقسام إنْ لم تكن جميعها، فإذا علمت أنّ الجامعة تستخدم نظام الملفات الورقي في معاملاتها، ناقش مع زملائك الحالات الآتية:

١- تسجيل بيانات الطالب في كلّ قسم على حدة.
٢- اختلاف كتابة اسم الطالب في الأقسام الأخرى.
٣- إطلاع إدارة الجامعة على عدد الطلبة المستعيرين الكتبَ بشكلٍ فوريّ ودقيق.
٤- تأثير حذف بيانات طالب من سجلّات قسم التسجيل لأمرٍ ما في الأقسام الأخرى.
٥- إطلاع بعض الأقسام على بيانات الطالب دون الأخرى.

إضاءة!
قاعدة البيانات: مجموعة من البيانات مرتّبة ومنظّمة بطريقةٍ يسهل الوصول إليها، وإدارتها،والتعديل عليها.

لعلك لاحظت في النّشاط السابق أنّ نظام الملفات الورقي يعاني مشكلاتٍ عدّةً في تنظيم وإدارة البيانات، إلّا أنّ استخدام قواعد البيانات المحوسبة في إدارة البيانات وتنظيمها يشكّل حلًّا جذريّاً لتلك المشكلات.
ولجعل قواعد البيانات أكثر فاعليّة، لا بدّ أن تتّصف بالخصائص الآتية:

- **تجنُّب التكرار (Redundancy):** عدم تكرار البيانات في أقسام المؤسسة المختلفة، بحيث يتم إدخالها في قاعدة بيانات مركزية، واستخدامها من الأقسام جميعها.

- **تناسق أو تطابق البيانات (Consistency):**

ويعني أنّ أيّ إجراءات تتمّ على قاعدة بيانات محدّدة يجب أن تغيّر البيانات المتأثرة بذلك الإجراء وبالطرق المسموح بها، والبيانات المكتوبة تكون صالحة وفقاً لقواعد محدّدة، فمثلاً: في حال الإدخال الخاطئ للبيانات بشكلٍ غير مقصود، فإنّ ذلك ينتج عنه بيانات غير متطابقة لبيانات الطالب نفسه.كقيام موظّف التسجيل بكتابة اسم طالب على النحو (احمد محمد علي سليمان)، وقيام موظف المكتبة بكتابة اسم الطالب نفسه على النحو (أحمد محمد علي سليمان)، فإنّنا سنواجه

مشكلة في تطابق البيانات؛ كونها اسماً لطالبين مختلفين من وجهة نظر قاعدة البيانات، وهذا ما تمّ التغلب عليه في قاعدة البيانات المركزية المحوسبة.

سؤال

ناقش الشكل أدناه على ضوء التكرار وتطابق البيانات في نظام إدارة الملفات ونظام إدارة قواعد البيانات.

رقم الطالب	اسم الطالب	التقدير	المادة
111	أحمد	B	CS201
111	أحمد	C	CS101
222	محمد	A	CS300

قسم المالية

رقم الطالب	اسم الطالب	القسط
111	احمد	90

- **مشاركة البيانات (Sharing):** تمكّننا قواعد البيانات من مشاركة البيانات بحيث يتمّ الوصول لقاعدة البيانات من قِبَل عدّة مستخدمين وبشكلٍ متزامن.

مثال قيام مدير المدرسة بالاطّلاع على علامات الطّلبة أثناء قيام المعلمين بإدخال تلك العلامات على قاعدة البيانات.

- **خصوصية وأمن البيانات (Privacy and Security):** توفّر قاعدة البيانات عدّة مستويات لحماية البيانات وخصوصيتها لكي لا تكون عرضة للفقد أو السرقة أو الاطلاع غير المعتمد أو غير المرخص به، نذكر منها:

- **المصادقة (Authentication):** عمليّة التأكد أنّ الشخص الذي يحاول الدّخول إلى قاعدة البيانات وإجراء بعض العمليات عليها مصرّح له بعمل ذلك. (مثال: اسم المستخدم وكلمة المرور).
- **التفويض (Authorization):** عملية تحديد الصّلاحيات (Permissions) للمستخدمين من قِبل «مدير النظام»، ونطاق البيانات المسموح للمستخدم التّعامل معها داخل قاعدة البيانات، فيمكن لمستخدم قراءة البيانات فقط، ويمكن لمستخدم آخر القيام بعملية التحديث والإضافة والحذف على البيانات.

استقلالية البيانات (Data Independence):

إحدى أهمّ الخصائص التي أدّت إلى تطوير وتطبيق نظم إدارة قواعد البيانات في المنظمات المختلفة، وتعني فصل البيانات عن البرامج المعالجة لها، والقدرة على إجراء تغييرات على خصائص البيانات دون الحاجة إلى إجراء تغييرات على البرامج التي تصل إلى البيانات بنظام إدارة قواعد البيانات (DBMS)، ويتمّ على مرحلتين:

1. الإستقلاليّة المنطقية للبيانات: فلو تمّ إضافة حقل أو جدول فإنّه لا يؤثّر على البيانات المعروضة (View) للمستخدم من خلال التطبيقات.
2. الإستقلاليّة الفيزيائية للبيانات: فلو تمّ تغيير في ترتيب الجداول المخزّنة فيزيائيا في قاعدة البيانات فإنّه لا يؤثر على نتائج الاستعلامات (Queries).

التّكامل المرجعي (Integrity Referential):

مجموعة قواعد وقيود يتمّ تطبيقها على الجداول المرتبطة بحيث تبقى العلاقات بين الجداول متناغمة والسجلّات متّسقة دون حدوث أخطاء أثناء ترابط البيانات بين الجداول (المرتبطة والأساسية) من خلال حقولها (الأساسي والأجنبي)؛ حيث لا يمكن القيام بعمليّة (إضافة أو تحديث أو حذف) على بيانات الجدول المرتبط دون الجدول الرئيسي، ويرتبط بهذا المفهوم تتالي التحديث (Cascading Update) وتتالي الحذف (Cascading Delete) والتي تضمن أنّ التغييرات في الجدول المرتبط ستنعكس على الجدول الرئيسي.

مثال حذف بيانات طالب من قسم "الطالب" يعني بالضرورة حذف بياناته من جميع الأقسام الأخرى (السجل الطبي، المكتبة،...).

ما الخطأ في الجداول أدناه؟ بيّن ذلك.

رقم التصنيف	اسم التصنيف
1	مواد غذائية
2	مواد صناعية
3	مواد اولية

جدول أساسي

رقم المنتج	اسم المنتج	رقم التصنيف
11	اطارات	2
33	حبوب	1
44	غاز	33

جدول مرتبط

قضية مناقشة:

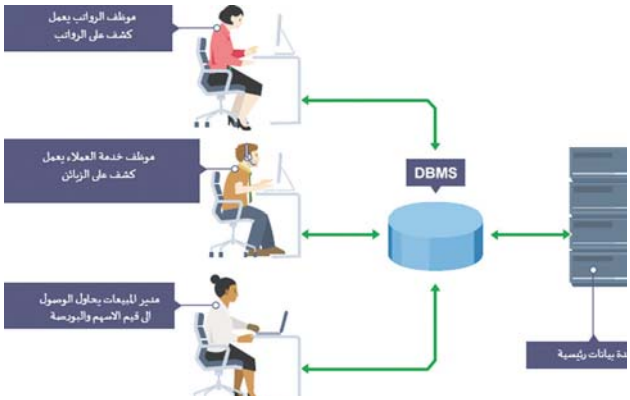
ناقش العبارة الآتية: "ساهمت الشبكات الحاسوبية في دعم وترسيخ قواعد البيانات".

إضافة إلى الخصائص المذكورة سابقاً، نذكر ما يأتي:

- المرونة في استخدام وتحديث البيانات.
- توفير نسخ احتياطية للبيانات ومعالجتها في حال حدوث أعطال لقواعد البيانات.
- اقتصادية الاستخدام من خلال التقليل من التكرار غير المبرّر للبيانات.
- تقليل زمن تطوير البرامج.

نظم إدارة قواعد البيانات Database Management Systems (DBMS)

تُعدّ الحاجة لتخزين البيانات في نظام أرشفة الملفات اليدوية وتصنيفها سبباً مهمّا لاسترجاع تلك المعلومات في أيّ وقت، وهنا تكمن العلاقة بين نظم إدارة قواعد البيانات (DBMS) ونظام الإرشيف؛ حيث أنّ نظم إدارة قواعد البيانات تقوم بمهمّة الأرشفة والتصنيف للبيانات بسرعة كبيرة ودقّة عالية بشكل محوسب، مما يضيف ميّزات أخرى مقارنةً بنظام أرشفة الملفات التقليدي؛ كزيادة إنتاجيّة المستخدمين، وسرعة استرجاع المعلومات، وزيادة أمن المعلومات، مع إمكانيّة مراقبة قاعدة البيانات طوال الوقت.



وتُعرّف إدارة قواعد البيانات أنّها مجموعة من البرامج والأنظمة الفرعية المسؤولة عن تنظيم البيانات للوصول لها بسرعة وسهولة، وتنفيذ جميع الإجراء والوظائف المختلفة على قاعدة البيانات، مثل: الإضافة والتّعديل والتخزين والحذف والاسترجاع والبحث والعرض لبيانات تلك القاعدة، بينما تعرّف نظم قواعد البيانات أنّها الجامعة بين قواعد البيانات وإدارة قواعد البيانات، وهذا ما جعلها تتغلّب على مشاكل نظام الملفات القديم باحتوائها خصائص كلّ من قواعد البيانات وإدارتها معاً واللّازمة لتحقيق استقلاليّة البيانات وإنشاء الرّوابط بينها بشكل أساسيّ.

من وظائف نظم ادارة قواعد البيانات نذكر ما يأتي:

١- تعريف قاعدة البيانات وإنشائها وعمل الاستعلامات وتحديثها وإدارتها.

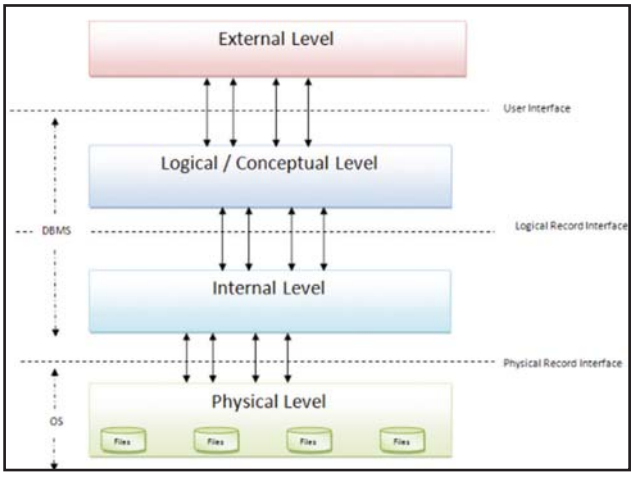
٢- تحديد قواعد التحقق من صحة البيانات، واعفاء المستخدمين من صياغة برامج صيانة البيانات.

٣- ترتيب وتنظيم البيانات داخل ملفات، وعرضها على شكل تقارير او نماذج منظمة.

عمارة نظم إدارة قواعد البيانات (Database Management Systems)

تشكّل نظم إدارة قواعد البيانات حلقة الوصل بين المستخدم وقاعدة البيانات مروراً بمدير الملفات (File Manager) وهو أحد برامج نظم التشغيل المسؤول عن توصيل قاعدة البيانات بنظم إدارة قواعد البيانات،

حيث تقوم باستقبال الطّلبات من المستخدمين وتنفيذ مجموعة عمليّات استجابة لهذه الطلبات لتزويدهم بالنتائج المطلوبة من خلال ثلاثة مستويات:

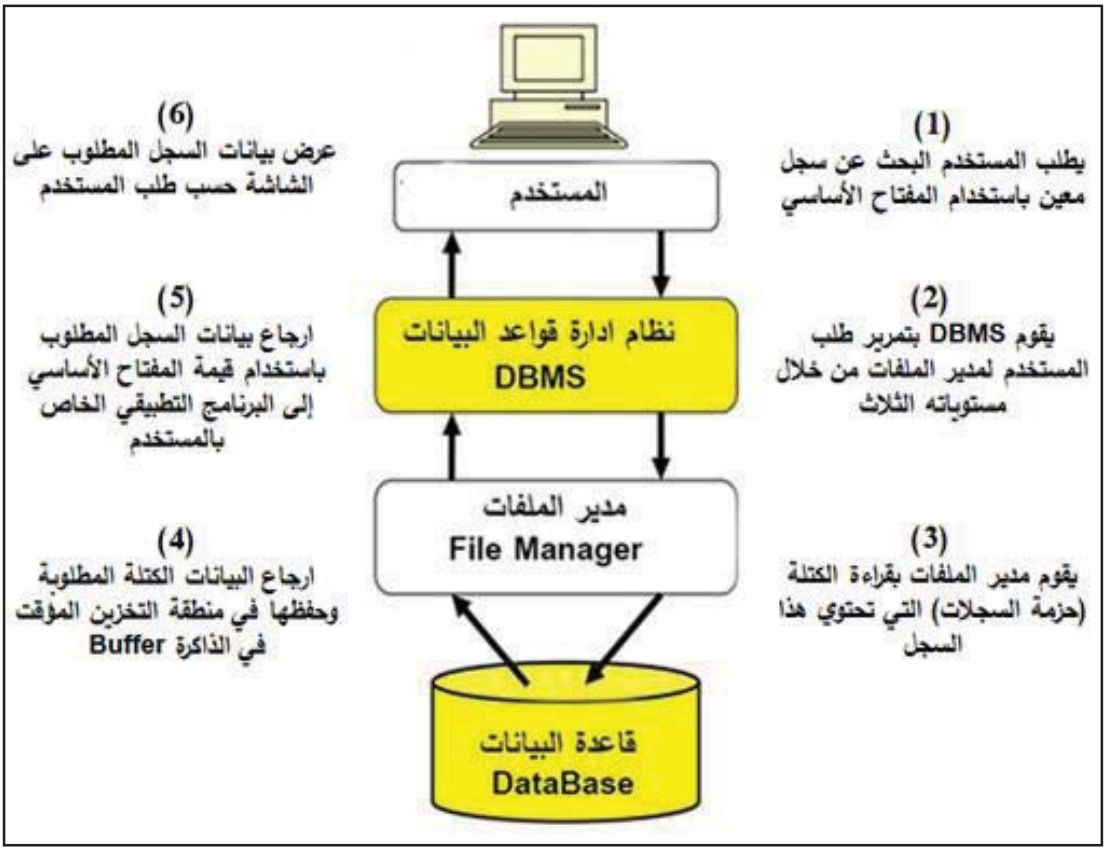


١- المستوى الداخلي (Internal Level):
وصف طريقة تخزين البيانات في وسائط التخزين المتعدّدة، ويتعلّق بشكلٍ مباشر بالتركيب الفيزيائي لقواعد البيانات.

٢- المستوى المنطقي (Logical Level):
شكل (Schema) البيانات العام وترابطها المنطقي بعضها مع بعض من خلال العلاقات (Relations).

٣- المستوى الخارجي (External Level):
تَعامُل المستخدم مع البيانات من خلال واجهات النظام (Interfaces) والعمليّات المختلفة.

الشكل أدناه يوضّح آليّة استرجاع سجل من ملف مخزّن داخل قاعدة البيانات:



مكوّنات أنظمة إدارة قواعد البيانات

يحتوي نظام إدارة قواعد البيانات (DBMS) على عدّة مكوّنات، يؤدي كلّ منها مهمّة محدّدة في بيئة DBMS، نذكر منها:

١. المكونات المادية (Hardware):

جميع المعدّات الملموسة وأجهزة الاتّصال في قواعد البيانات. انظر الشكل المجاور.

خوادم servers

الحاسبات computers

وسائط التخزين المختلفة
storage media

الطابعات printers

٢. المكونات البرمجية (Software):

مجموعة البرامج المستخدمة للتحكم في قاعدة البيانات وإدارتها؛ ومنها برمجيّات DBMS ذاته، وأنظمة التشغيل، وبرامج الشبكة لمشاركة البيانات بين المستخدمين، وبرامج التطبيقات للوصول إلى البيانات في (DBMS).

أ. أنظمة التشغيل (O.S.): مجموعة برمجيّات لإدارة وتشغيل أجهزة الحاسب، وتُعدّ كحاضنة للبرمجيّات الأخرى ضمن نظام التشغيل.

سؤال

اذكر ثلاثاً من أمثلة أنظمة التشغيل؟

ب. برمجيّات قواعد البيانات: مجموعة البرامج المسؤولة عن تصميم ومعالجة قواعد البيانات، مثل:

- معالج لغة الإستعلام (SQL).
- برامج تصميم نماذج الإدخال و الإخراج (Screen Formatter).
- برامج النسخ الإحتياطي (Backup).
- برامج توزيع قواعد البيانات (Distribution).
- برامج محرّك قاعدة البيانات (Database Engine).

ج. البرمجيات التطبيقية: مجموعة البرامج والواجهات التي يقوم ببنائها المبرمجون للشركات والمؤسسات والوزارات لحلّ مشاكل لديهم، مثل: برامج التّسجيل الجامعي وبرامج المحاسبة التّجارية، وبرامج الصيدليّات ومحطّات الوقود.

٣. الموارد البشرية:

أ. مدير قواعد بيانات (DataBase Administrator-DBA): الشخص المسؤول عن إدارة وتنظيم قاعدة البيانات من حيث: منح الصّلاحيّات (Permissions)، وتوفير الأمن والحماية (Security) لها.

ب. مصمّم ومحلّل النّظم (System Analyst): الشخص المسؤول عن تصميم وتحليل قاعدة البيانات.

ج. المبرمجون (Programmers): الأشخاص المسؤولون عن ترجمة قاعدة البيانات إلى لغة يفهمها الحاسوب، وكتابة البرامج التّطبيقية بإحدى لغات البرمجة، والتّأكد من سلامة هذه البرامج ودقّتها في استخراج النّتائج من قواعد البيانات.

د. المستخدمون (Users): الأشخاص المُناط إليهم التّعامل مع واجهات النظام وعمليّاته المختلفة التي توفّرها.

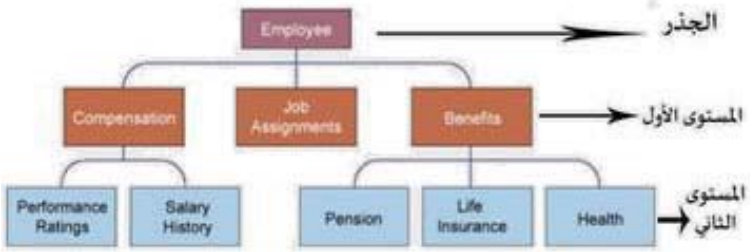
٤. البيانات (Data): ويعدّ المكوّن الأكثر أهمية، حيث تحتوي قاعدة البيانات على البيانات الفعلية والبيانات الوصفية، ويقوم نظام (DBMS) بجمع وتخزين ومعالجة والوصول إلى تلك البيانات.

نماذج قواعد البيانات

مرّت قواعد البيانات بعدّة مراحل لحين الوصول الى التّطور الحالي، فلا بدّ من الحديث عن الثّلاثة نماذج لقواعد البيانات، وهي: الهرميّ، والشّبكي، والعلائقي الذي رغم التطورات المتسارعة في نظم قواعد البيانات ما يزال النظام المعمول به في حزم (DBMS) المختلفة ولبنة الأساس للأنظمة الحديثة.

١ النّموذج الهرميّ Hierarchical Model

يعدّ النّموذج الأوّل لتمثيل قواعد البيانات، حيث اعتمد على الروابط للتنقل بين ملفاته، وتمّ بناء هذا النموذج من الأعلى للأسفل (Up-down)، فالمستوى الأعلى يعتبر كأب للمستوى في الأسفل كما بالشكل المجاور.

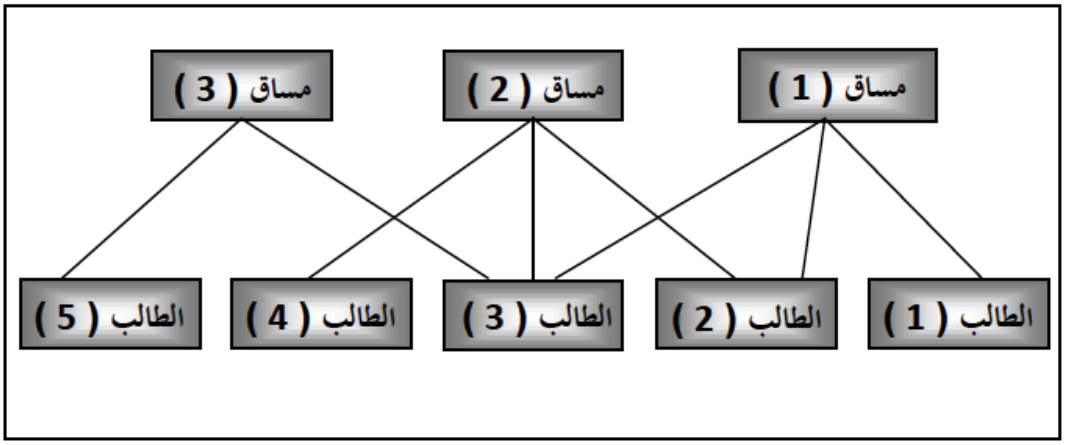


قضية بحث:

من خلال الانترنت، ابحث عن عيوب النموذج الهرمي في بناء قواعد البيانات.

٢ النّموذج الشّبكي (Network Model):

يتمّ تخزين البيانات في هذا النموذج بصورة سلاسل مترابطة من البيانات، حيث توصف العلاقات المنطقية بطريقة أكثر تعقيداً من سابقه، ويعتمد في بنائه على مفهوم المجموعات الجزئيّة والعناصر، ويطلق على المجموعات مصطلح «المالكين» (Owners) وعلى العناصر «المستخدمين» (Users)، حيث يناسب التركيب الحقيقي للبيانات.



٣ النموذج العلائقي (Relational Model):

يُعدّ الأكثر انتشاراً بعد ظهور حُزم نظم إدارة قواعد البيانات لسهولة الفهم والاستخدام، وفكرته الأساسية تعتمد على تحويل المفاهيم الطبيعية الموجودة في العالم الحقيقي إلى نموذج مصغّر يُمثّل قواعد البيانات المخزّنة في الحاسوب.

نموذج الكيان العلائقي

يعتمد بناء النموذج العلائقي على ثلاثة مكوّنات أساسيّة:

١- الكيان (Entity): وحدة أساسية مستقلّة محطّ اهتمام النّظام، يشير إلى شيء له وجود حقيقي، مثل: (شخص، منزل، سيارة،...)، أو وجود منطقي، مثل: (شركة، مقرر، دورة...)، تمثّل بقواعد البيانات على شكل جدول (مِلف).

تشكّل مجموعة الكيانات (نظاماً) يُمثَّل بقاعدة البيانات؛ ففي الشّكل المجاور، نجد نظام (قاعدة البيانات) المستشفى يحوي الكيانات الآتية: (مريض، طبيب، دواء،....)، حدّد كيانات أخرى؟

٢- الصفة (Attribute): سمات أو خصائص لوصف الكائن، ولكل صفة من صفات الكيان مجال (Domain) من القيم التي من الممكن أن يحتويها، فمثلاً: يمكن وصف كيان (طالب) حسب: رقمه، واسمه، وتاريخ ميلاده، وتخصصه،... حدّد صفات أخرى؟ وفي قواعد البيانات، يطلق على الصّفات حقولاً (Fields) وتمثّل (أعمدة) في الجدول، وقيم مجموعة الحقول لصف واحد في الجدول يسمّى سجلّاً (Record).

٣- العلاقة (Relation): هي الروابط بين كيانين أو أكثر في النّظام، ولها درجة ونوع وشروط محدّدة، انظر الشكل أدناه.

مقارنة بين النماذج الثلاث: يَستخدم النموذجان (الهرمي والشبكي) روابط (links) لوصل السجلات (Records) ببعضها البعض في النظام، لأن السجلات فيها مرتبطة ببعضها البعض فيزيائيّاً، وتتميّز بدرجة صعوبتها في العمل والتعديل، إلّا أن سرعة الوصول للبيانات فيها تغطّي عيوبها. أما الأنظمة العلائقية فالربط بين السجلات يكون عن طريق أسماء الحقول، فالسجلات في هذا النظام قابلة للعنونة بالمحتوى بحيث يجري الوصول إليها بمطابقة قيم البيانات المخزنة مع بعضها.

سؤال مستعيناً بالشكل المجاور، أجب عن الأسئلة الآتية:

- ما اسم النظام (قاعدة البيانات)؟
- ما اسم الكيان (الجدول)؟
- ما أسماء الصفات (الحقول)؟
- ما مجال القيم الممكنة لكلّ حقلٍ من الحقول؟
- بعد إجابتك على الأسئلة أعلاه، قم بتعبئة الجدول أدناه.

الدواء					

أسئلة الدرس

1- ما المقصود بالمفاهيم الآتية: قاعدة البيانات، إدارة قواعد البيانات، استقلالية البيانات، النّموذج الشبكي، التكامل المرجعي.
2- عدّد مكوّنات نظم إدارة قواعد البيانات، مع إعطاء مثالٍ لكلّ منها.
3- اذكر اثنين من أساليب الحماية في قاعدة البيانات، مع التّوضيح.
4- ما الفرق بين تكرار البيانات وتناسق البيانات؟
5- اشرح مستويات عمارة نظم إدارة قاعدة البيانات.
6- وضّح آلية استرجاع سجل من قاعدة البيانات، ممثّلا ذلك بمثال واقعيّ بمدرستك؟
7- حدّد ثلاثاً من الكيانات مع الصفات المناسبة لكلّ منها لنظام مركز تعليمي.
8- عدّد بنقاط أهمية قواعد البيانات المحوسبة.
9- اذكر اثنين من سلبيات النّموذج الهرمي والنّموذج الشبكي.

10- علّل ما يأتي:

١- تتماثل البيانات المخزّنة في الحقل وتختلف عنها في السّجل.

٢- يفضّل استخدام الملفات المحوسبة بدلاً من الملفات اليدويّة في محل تجاري.

- دراسة حالة:

ادرس الحالة الآتية، ثمّ أجب عن الأسئلة التي تليها:

لدى محمد شركة مقاولات عامّة تعمل داخل الأراضي الفلسطينية منذ زمن بعيد، ولدى الشركة العديد من الأفرع موزعة على مدن الضفة الغربية، وتعتمد الشركة النظام التقليدي لتخزين ملفّاتها، سمع «محمد» بتخصص تكنولوجيا المعلومات وأراد أن يحوسب عمله ويحول نظام الملفات القديم إلى نظام محوسب كقاعدة بيانات. لكنّه غير مقتنع بهذا النظام.

أ- كيف تقنع محمد أنّ هذا النظام أفضل من النظام التقليدي من حيث: الترتيب والدقة.

ب- اذا سألك محمد عن الوظائف التي سيقوم بها نظام قاعدة البيانات الجديد كيف ستجيبه.

ج- في حال اقتنع محمد بعمل النظام وأوكل إليك مهمّة عمل النظام، ما هي المكونات المادّية والبرمجيّة التي ستفكر فيها. ولماذا؟

د- ما هو نموذج قواعد البيانات الذي سوف تعتمده؟ ولماذا؟

الدرس (٢)

تحليل قواعد البيانات

يُعدّ تحليل مشكلة قاعدة البيانات من الخطوات الأولى التي تعتمد عليها شمولية وتكاملية قاعدة البيانات والتي تسبق عمليّة التصميم والتنفيذ، حيث تمثّل حجر الأساس لبناء الأنظمة وحلّ المشكلات على اختلاف طبيعتها، ويقصد بتحليل النظام تحديد كيانات النظام، وصفاتها والعلاقات فيما بينها، وهذه العمليّة تسبق تصميم وتنفيذ قاعدة البيانات عمليّا.

في هذا الدرس سيتمّ العمل على مشكلة ما للوصول إلى مخطط ERD (Entity Relationship Diagram) والذي يُظهِر الجداول والحقول والمفاتيح والعلاقات وأنواعها.

جمع البيانات وتحليل النظام.

تُعدّ عملية جمع البيانات جزءاً لا يتجزّأ من عمليّة تحليل النظام والتي من شأنها إعطاء وصف شامل أو جزئي لكيانات وحقول وعلاقات قاعدة البيانات، المراد منها تلبية احتياجات ومتطلبات الشخص الذي يتطلّع لحلّ مشكلته أو تطوير أو تيسير عمله، ويتمّ ذلك من خلال المقابلة الشخصية أو أيّة طرق أخرى، ولهذا، قد تختلف كيانات وصفات وعلاقات قاعدة البيانات ذاتها من شخص لآخر اعتماداً على الغرض منها.

تأتي خطوات تحليل نظام يحوي قاعدة بيانات في النّموذج العلائقي وفق التّسلسل الآتي:

أوّلاً: تحديد كيانات النّظام وصفاته.

ثانياً: رسم نموذج ERD (Entity Relationship Diagram) أوّلي.

ثالثاً: تسوية البيانات في القاعدة بالصيغة المعيارية الثالثة (3NF).

رابعاً: تحديد العلاقات بين كيانات النّظام.

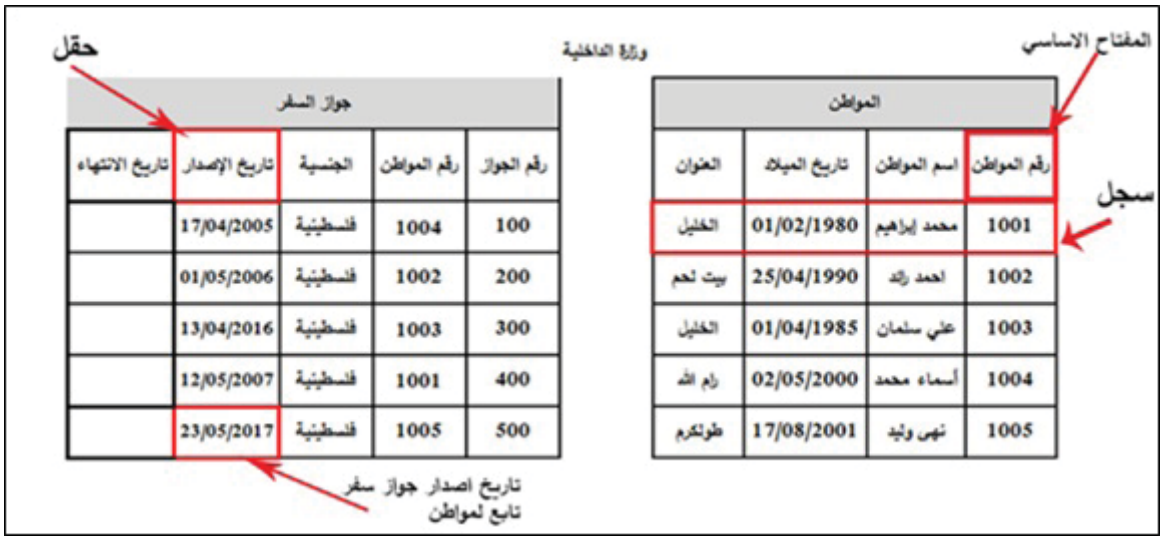
خامساً: تنفيذ النظام باستخدام أحد برامج معالجة قواعد البيانات.

✷ أوّلاً: تحديد كيانات النّظام وصفاته (Entities & Attributes)

تتلخّص فكرة قاعدة البيانات العلائقية بمجموعة من الكيانات (جداول - Tables) والعلاقات (Relations)، ومنها جاءت تسمية النّموذج بالعلائقي، ولكلّ كيان داخل قاعدة البيانات اسم فريد (يفضّل أن يكون بصيغة المفرد)، يتكوّن من مجموعة صفوف (سجلات) وأعمدة (حقول) حيث تمثّل صفات البيانات ونوعها، مثل: بيانات نصية، ورقميّة وغيرها.

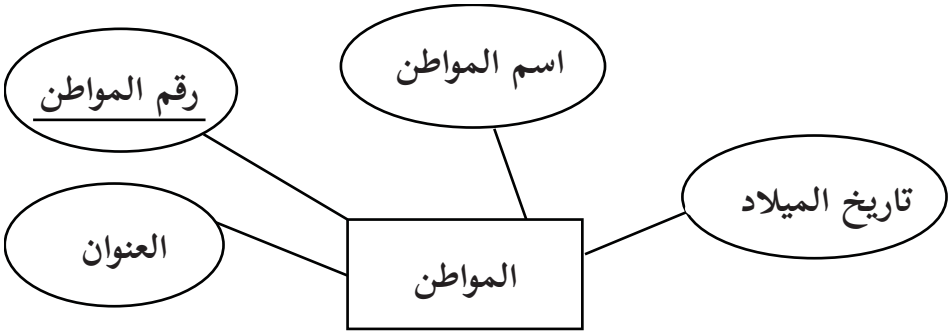
نشاط (١) دراسة قاعدة بيانات

تمعّن قاعدة البيانات أدناه، ثمّ أجب عن الأسئلة التي تليها:



١. ما اسم قاعدة البيانات؟
٢. عدّد أسماء الكيانات المكوّنة لقاعدة البيانات.
٣. ما الصّفات المكوّنة لكلّ جدول من الجداول في قاعدة البيانات؟
٤. اقترح نوع بيانات مناسب لكلّ صفة من الصفات أعلاه.
٥. اكتب قيم السجل الرابع في جدول «جواز السفر»؟
٦. عمليّاً، هل يمكن احتواء السجل الواحد في أيّ جدول نوعاً محدّداً من البيانات؟ لماذا؟
٧. ما تاريخ إصدار جواز السفر للمواطن «محمد ابراهيم»؟ كيف عرفت ذلك؟

لاحظت من النشاط السابق أنّ الكيان له مجموعة صفات (حقول) تميّزه عن غيره من الكيانات, وأنّ السجلات داخل الكيان الواحد يتمّ تمييزها من خلال المفتاح الأساسي (Primary Key) الذي يُعبّر عنه بإحدى الصّفات أو مجموعة الصّفات الفريدة (لا تتكرّر قيمها، ولا تقبل بالقيمة الفارغة (NULL))، حيث يُستخدم للربط بين جدولين أحدهما يحوي حقل المفتاح الأساسي والآخر يحوي نفس الحقل مع عدم تحقيقه لشروط المفتاح الأساسي، ويطلق عليه المفتاح الأجنبي (Foreign Key).
يُعبّر عن الكيان داخل مخطّطات ERD -الذي سنتناوله لاحقاً- بشكل مربع/مستطيل، بينما يُعبّر عن الصفات بالشكل البيضاوي. لاحظ المخطّط الآتي:

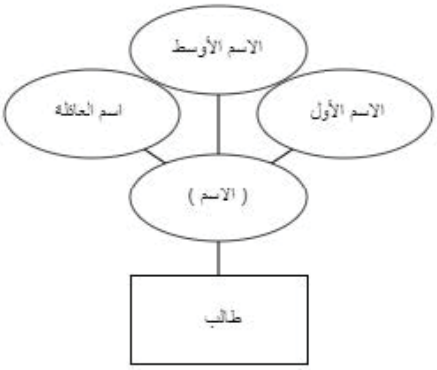


- اقترح صفاتٍ أخرى لهذا الكيان، ثمّ أضفها للشّكل السابق.
- لماذا تمّ وضع خطّ سفليّ للصفة (رقم المواطن) ولم توضع لبقيّة الصفات؟
- ارسم مخطّط الكيان (جواز السفر) للنّشاط السّابق؟

تُصنّف الصّفات المكوّنة للكيانات -من حيث تكوينها- إلى:

أ- صفة بسيطة أو مركّبة (Simple or Composite Attribute)

يُطلق على الصفة التي لا يمكن تجزئتها بالصفة البسيطة، مثل: عمر الطالب؛ حيث لا يمكن تقسيمها، بينما يُطلق مفهوم الصّفة المركّبة على الصفة التي يمكن تقسيمها إلى أجزاء ذات دلالة، مثل: اسم الطالب؛ حيث يمكن أن يكون وحدة واحدة، أو يتمّ تجزئته إلى: الاسم الأول، واسم الأب، واسم الجد، واسم العائلة.



تُمثّل الصفات المركبة في مخططات ERD بشكل بيضاويّ للصفة العامّة، وأشكالٍ بيضاوية فرعية متّصلة بالصفة العامة تُعبّر عن أجزاء هذه الصّفة، والشكل المجاور يوضّح ذلك.

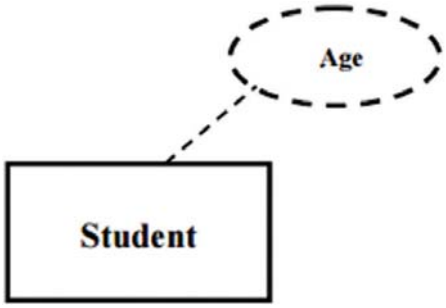
ب- صفة وحيدة أو متعدّدة القيم (Single-valued or Multi-Valued Attribute)

تحتوي الصفة الوحيدة على قيمة واحدة فقط، مثل: رقم السيارة أو رقم الهوية، بينما تحتوي الصّفة متعدّدة القيم على أكثر من قيمة، مثل: البريد الالكتروني أو رقم الهاتف؛ فمن الممكن أن يملك الشخص أكثر من هاتف أو أكثر من بريد الكتروني، وتمثّل الصفة متعدّدة القيم في مخطّطات ERD كما في الشكل المجاور.

ملاحظة:

الصفة الوحيدة ليس بالضرورة أن تكون صفة بسيطة.

ج- الصفة المشتقة (Derived Attribute)



تعتمد قيمتها على قيم صفات أخرى؛ أي يتمّ اشتقاقها من صفات أخرى داخل قاعدة البيانات، مثل: عمر الطالب، يتمّ اشتقاقه من تاريخ الميلاد والتاريخ الحالي (عمر الطالب = التاريخ الحالي – تاريخ الميلاد)، ويرمز للصفة المشتقة بشكل بيضاوي متقطع كما في الشكل المجاور. كما أن الصفات المشتقة لا تُخزّن، وإنما توضع طريقة اشتقاقها (مثل: المعادلة اللّازمة لحسابها).

مثال: نظام البقالة (Market)

يُعدّ الوصف التفصيلي لعمل ما عنصراً هامّا في خطوات بناء قاعدة البيانات قويّة، حيث يتمّ تحديد الكثير من الأمور وأبرزها: كيانات وصفات قاعدة البيانات، ويكون ذلك بإجراء مقابلة شخصيّة مع الأطراف المعنيين، ليتمّ تدوين وصف شامل للنظام وعمليّاته.

في نظام البقالة، وُصف العمل المطلوب كما يأتي:

- تحتوي البقالة على العديد من البضائع والمنتجات المصنّفة لأنواع مختلفة، مثل: المنظّفات، والبسكويت واللحوم والمعلبات وغيرها.
- يدخل المتسوّقون من الزبائن يوميّا إلى البقالة، ويتسوّقون ثمّ يدفعون بطرق مختلفة (نقدي، بطاقة آليّة وغيرها).

- يتمّ عمل فاتورة بيع لكلّ زبون.
- تدخل بضاعة جديدة كلّ فترة من الزمن للبقالة.
- أسعار السلع تكون حسب الباركود لكلّ منتج، لذلك يجب إدخال سعر البضاعة من خلال جهاز الباركود.

هل يمكن إدارة هذه البقالة يدويّا؟ إذا كانت الإجابة نعم؛ فكم عدد الموظفين اللّازم توظيفهم لإنجاز هذا العمل؟ هل يمكن إصدار الكشوفات والحسابات في الوقت المحدّد؟

ملاحظة:

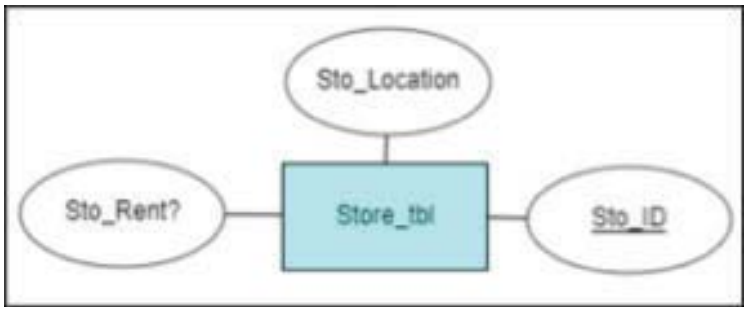
الباركود: هي صور مربعة او مستطيلة تحتوي على خطوط لونية سوداء وبيضاء بمساحات مختلفة، يمكن قراءتها عن طريق الحاسوب، تستخدم للتعريف بالمنتجات داخل المحال التجارية، ولمتابعة المنتجات داخل المصانع وغيرها.

اعتماداً على معطيات نظام البقالة السّابق، يظهر الحاجة إلى عدّة كيانات رئيسة، وهي:

- كيان "مخزن البضائع" (Store_tbl).
- كيان "المنتج" (Product_tbl).
- كيان "الزبون" (Customer_tbl).
- كيان "دفعات الزبون" (CustomerPayment_tbl).
- كيان "فاتورة المبيعات" (BillSale_tbl).

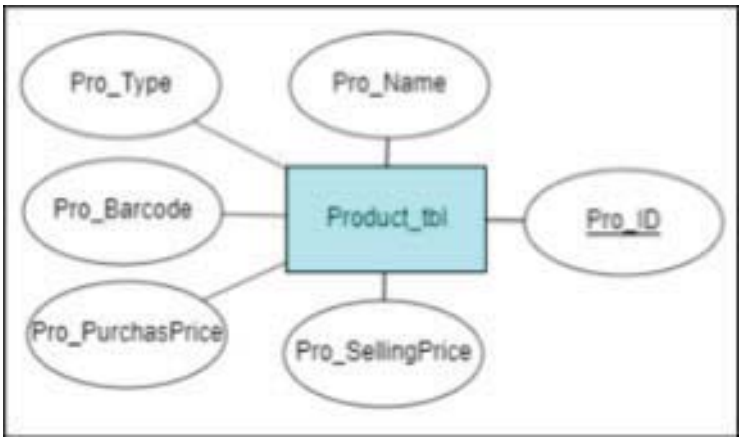
ويمكن أن يشتمل كلّ كيان منها على الصّفات الآتية:

أ) كيان المخزن (Store_tbl): توضع فيه كمّيات المنتجات المتواجدة داخل مخزن البقالة، بحيث يتمّ التعديل عليها في حال تمّ البيع أو الإضافة، ويتضمّن الصفات المقترحة الآتية:



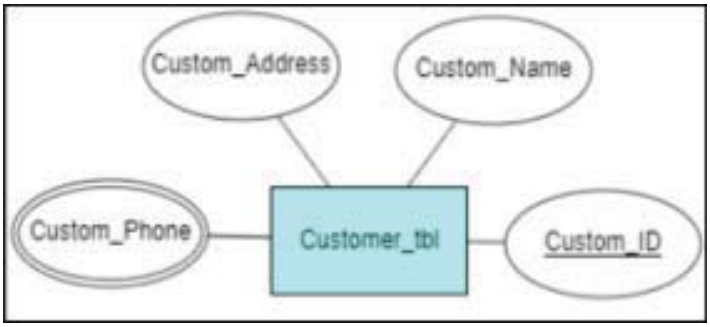
- Sto_ID: مفتاح أساسي، رقم المخزن.
- Sto_Location: مكان المخزن.
- Sto_Rent?: هل المخزن مستأجر؟

● ب) كيان المنتجات (Product_tbl): يستخدم لتخزين المنتجات، ويتضمّن الصفات المقترحة الآتية:



- Pro_ID: المفتاح الأساسي، رقم المنتج.
- Pro_Name: اسم المنتج.
- Pro_Type: نوع المنتج.
- Pro_Barcode: رقم الباركود للمنتج.
- Pro_PurchasePrice: سعر الشراء
- Pro_SellingPrice: سعر البيع.

● ج) كيان الزبون (Customer_tbl): يستخدم لتخزين بيانات الزبائن، ويتضمّن الصفات المقترحة الآتية:



- Custom_ID: المفتاح الأساسي، رقم الزبون.
- Custom_Name: اسم الزبون.
- Custom_Phone: رقم هاتف للزبون.
- Custom_Address: عنوان الزبون.

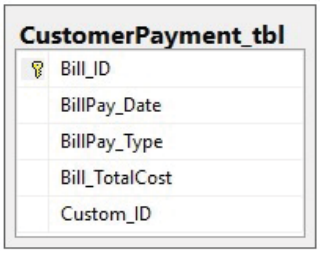
ملاحظة:

قد تظهر حاجة ملحّة لبناء كيانات إضافيّة أثناء تحليل النظام.

نشاط (٢) كيانات نظام البقالة.

◄ من خلال دراستك لمتطلبات نظام البقالة، قم بما يأتي:

١. حدّد صفات الكيانات المتبقّية، اعتماداً على البيانات الآتية:



٢. اقترح كيانات جديدة غير المذكورة سابقاً ويحتاجها النّظام.

المفاتيح

يُعّد المفتاح الأساسيّ مهماً لتمييز السّجلات داخل الكيان الواحد وإجراء عمليّات الإسترجاع، والتحديث، والحذف، وعمل الروابط بين الكيانات المختلفة.

نشاط (٣) المفاتيح في قواعد البيانات.

تمعّن الكيانات بقاعدة البيانات أدناه، ثم أجب عن الأسئلة التي تليها:

المساقات

رقم المساق	اسم المساق	ساعات معتمدة	ساعات نظرية	ساعات عملية
220	مقدمة حاسوب	3	2	1
230	جبر خطي	3	2	1
240	رياضيات متقطعة	3	2	1
250	شبكات	2	1	1
260	نحو1	2	1	1
270	نحو2	2	1	1

الطلبة

رقم الطالب	رقم الهوية	اسم الطالب	الجنس	مكان الميلاد	تاريخ الميلاد	المعدل	التخصص
1000	94956901	محمد	ذكر	الخليل	01/02/2000	55	تكنولوجيا
1001	58456254	عفاف	انثى	الخليل	15/05/2001	69	رياضيات
1002	88754215	سليمان	ذكر	رام الله	13/04/2001	85	تكنولوجيا
1003	88754211	إبراهيم	ذكر	جنين	17/05/2002	47	لغة عربية
1004	25658598	رقية	انثى	طولكرم	14/08/2001	59	لغة عربية
1005	25874136	فاطمة	انثى	بيت لحم	01/01/1999	96	تكنولوجيا

التسجيل الفصلي

رقم الطالب	رقم المساق	تاريخ التسجيل
1000	250	14/09/2016
1005	220	15/08/2016
1001	230	14/10/2015
1003	260	11/11/2016
1000	220	01/05/2016
1005	250	01/05/2016

١. اقترح اسماً مناسباً لقاعدة البيانات؟

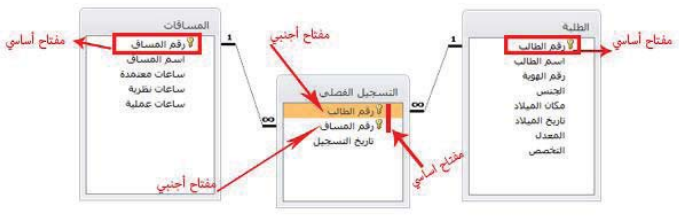
٢. أيّ الصفات تُمثّل مفتاحاً أساسيّا في كل كيان من الكيانات السابقة؟

٣. ما الصفة المناسبة لربط كيان "الطلبة" بكيان "التسجيل الفصليّ"؟

٤. لماذا لم تُستخَدم صفة "رقم الهوية" كمفتاحٍ أساسيّ؟

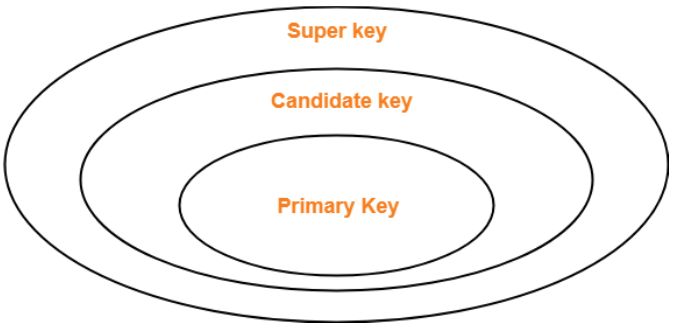
نستنتج من خلال النشاط السّابق أنّ هناك عدّة أنواع من المفاتيح، نذكر منها ما يأتي:

أ. المفتاح الأساسي (Primary Key): صفة أو أكثر فريدة، لها قيمة غير خالية ولا تتكرّر في الصفة، فمثلاً؛ صفة (رقم الطالب) في كيان (الطلبة) مفتاح أساسي بسيط لأنه مُؤلّفٌ من صفةٍ واحدة بينما الصفتان (رقم الطالب) و(رقم المساق) في كيان (التسجيل الفصلي) يؤلّفان معاً مفتاحاً أساسيّا مركّباً.



● **ب. المفتاح الأجنبي (Foreign Key):** صفة في كيان مرتبطة بصفة أخرى في كيان آخر تمثّل مفتاحاً أساسيّا، مثل: صفة (رقم الطالب) في كيان (التسجيل الفصلي)؛ كونه أساسيّ في كيان (الطلبة).

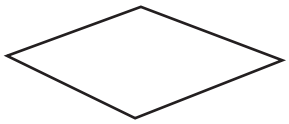
● **ج. المفتاح المرشح (Candidate Key):** صفة فريدة لا تتكرّر تصلح أن تكون مفتاحاً أساسيّا، ولكن، لم يتمّ اختيارها، مثل: صفة (رقم الهوية) في كيان (الطلبة).



● **د. المفتاح الأعظم (Super Key):**
أقلّ مجموعة من الصّفات التي يتمّ من خلالها تمييز السجل عن بقيّة السجلات في الكيان. هل يعتبر المفتاح الأساسيّ مفتاحاً أعظم ؟

الروابط والعلاقات (Relationships)

عمليّة ربط الكيانات مع بعضها البعض، تضمن عمليّة تناسق البيانات في الكيانات، وإمكانيّة استرجاع البيانات من أكثر من كيان في الوقت عينه، ونعبّر عن هذا الترابط بعبارات، مثل:(ينتمي إلى، يحتوي، يسجل، يتكون من، له)، ويتمّ التعبير عن العلاقة باستخدام شكل المعين، حيث يكتب داخله اسم العلاقة كما في الشّكل:



نشاط (٤) العلاقات بين الكيانات.

◀ تمعّن الكيانات الآتية، ثمّ أجب عن الأسئلة التي تليها:

القسم		
رقم القسم	اسم القسم	الدرجة العلمية
110	الهندسة	بكالوريوس
120	الزراعة	دبلوم
130	الآداب	دبلوم
140	العلوم الادارية	بكالوريوس

رئيس القسم			
الرقم	اسم رئيس القسم	العنوان	رقم القسم
1	رضوان	الخليل	120
2	مروان	رام الله	110
3	سلاف	أريحا	140
4	حياة	جنين	130

الطالب				
رقم الطالب	اسم الطالب	العنوان	الجنس	رقم القسم
1000	ابراهيم	رام الله	ذكر	110
1001	ساجدة	جنوب الخليل	أنثى	120
1002	سامي	نابلس	ذكر	110
1003	فاطمة	دورا	أنثى	140

المادة		
رقم المادة	اسم المادة	عدد الساعات
CS100	معادلات تفاضلية	3
CS200	تركيب بيانات	4
CS300	ذكاء اصطناعي	3
CS400	مختبر الكترونيات1	1

١. ما علاقة صفة (رئيس القسم) بكيان (القسم)؟

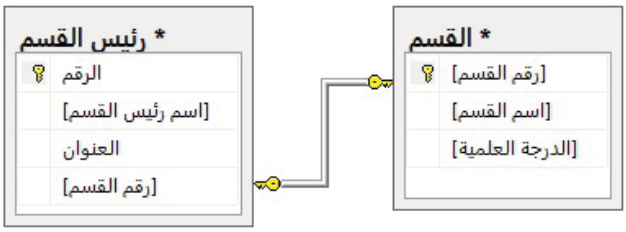
٢. هل هناك علاقة بين كيان (القسم) وكيان (الطالب)؟ استنتج تلك العلاقة إن وجدت.

٣. هل يمكن أن يدرس الطالب في أكثر من قسم؟ وضّح إجابتك ؟

٤. ما الرابط بين كيان (الطالب) وكيان (المادة) التي يدرسها من حيث: عدد المواد التي يدرسها الطالب.

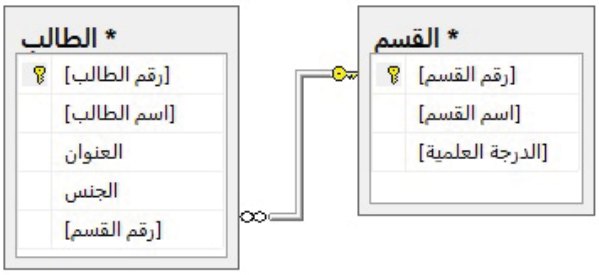
لعلّك لاحظت من خلال النّشاط السّابق أنّ هناك عدّة أنواع من العلاقات بين الكيانات، تهدف لتكامل البيانات مع إمكانية استرجاعها من أكثر من كيان بشكلٍ دقيقٍ وسليم، تصنّف العلاقات بين الكيانات كما يأتي:

١ علاقة واحد إلى واحد (One To One):



علاقة تربط كلّ سجل في كيان ما بسجل واحد من كيان آخر، والعكس صحيح، مثال ذلك: العلاقة بين كيان (القسم) وكيان (رئيس القسم)؛ وتقرأ: القسم الواحد له رئيس قسم واحد، ورئيس القسم الواحد يرأس قسماً واحداً. ويرمز لها (1:1).

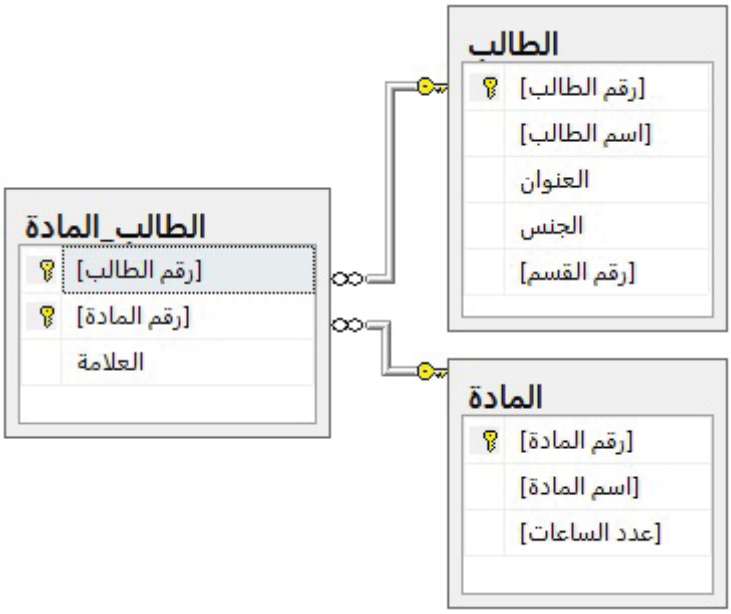
٢ علاقة واحد إلى متعدد (One To Many):



علاقة تربط سجلاً في كيان بعدّة سجّلات من الكيان الآخر، والعكس غير صحيح، ومثال ذلك: العلاقة بين كيانَي "القسم" و"الطلبة"؛ وتقرأ: القسم الواحد يحوي عدّة طلبة، والطالب الواحد موجود في قسم واحد. أعطِ أمثلة أخرى؟ ويرمز لها بالرمز (1:∞) مع مراعاة الترتيب؛ حيث أنّ علاقة: (1:∞) ≠ (∞:1)، والارتباط بين الكيانين هنا يكون باستخدام المفتاح الأجنبي.

سؤال لماذا لم يتمّ ايجاد الصفة (رقم الطالب) في كيان (القسم) بدلاً من العلاقة السّابقة؟

٣ علاقة متعدّد إلى متعدّد (Many To Many):



يرتبط السجل في الكيان بعدة سجلات في الكيان الآخر والعكس صحيح، وهذه العلاقة غير مقبولة داخل قاعدة البيانات؛ لصعوبة استرجاع البيانات بشكلٍ دقيق، لذلك، نلجأ إلى بناء كيان جديد يسمّى كيان (الوصلة أو الوسيط) وغالباً ما يتشكّل اسمه من اسم الجدولين المرتبطين معاً، فمثلاً: العلاقة بين كيان (الطالب) وكيان (المادة)، حيث أنّ الطالب الواحد يدرس عدّة مواد، والمادة الواحدة يدرسها عدّة طلاب، ولربط هذين الكيانين فإنّنا ننشئ كياناً وسيطاً يُسمّى (الطالب_المادة) أو (الدراسة)، بحيث يحتوي الكيان الجديد على حقلَين كحدّ أدنى، هما: المفتاحين الأساسيين في الكيانين (رقم الطالب، رقم المادة) فيصبحان معاً مفتاحاً أساسيّا مركّباً للكيان الوسيط مع إمكانيّة إنشاء أيّة صفات أخرى مناسبة للكيان الجديد.

نشاط (٥) كيانات نظام البقالة.

◄ من خلال دراستك لمتطلبات نظام البقالة، قم بما يأتي:

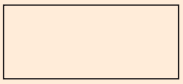
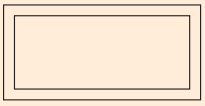
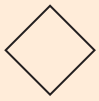
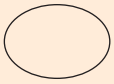
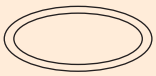
١. بيّن العلاقات لكيانات نظام البقالة السابق.

- كيان المنتج (Product_tbl)، كيان الزبون (Customer_tbl).
- كيان المنتج (Product_tbl) وكيان المخزن (Store_tbl).
- كيان دفعات الزبون (CustomerPayment_tbl) وكيان الزبون (Customer_tbl).

٢. اقترح نوع العلاقة بين كيانات النظام والكيانات التي اقترحتها في نشاط سابق.

نموذج ERD (Entity Relationship Diagram)

تقنية تحويل البيانات إلى نماذج، حيث يتمّ تمثيل محتويات النظام بشكل رسوميّ مفاهيميّ، تظهر من خلاله عناصر النظام الثلاثة: الكيان، الصفات، العلاقات. فيُعبّر عن تلك العناصر بأشكالٍ رسوميّة كما في الجدول الآتي:

الرمز	اسم الرمز	الوصف
	مربع او مستطيل	يكتب داخله اسم الكيان
	مربع او مستطيل مزدوج	يكتب داخله اسم الكيان الضعيف
	المعين	يكتب داخله اسم العلاقة بين الكيانين
	الشكل البيضاوي	يكتب داخله الصفة (الحقل) للكيان
	الشكل البيضاوي بخطوط متقطعة	يكتب داخل اسم الحقل المشتق.
	شكل بيضاوي مزدوج	يكتب داخله اسم الحقل الذي يحتوي على اكثر من قيمة.
	الخطوط المستقيمة	للربط بين الكيانات بعضها ببعض، والصفات الكيانات، وعند الربط بين الجداول توضع على اطرافه بالقرب من الجداول نوع العلاقة (ظ1-1) او (∞-1) او (∞-∞).

ولفهم طريقة التعامل مع هذه الرموز، تأمّل المثال الآتي:

مثال

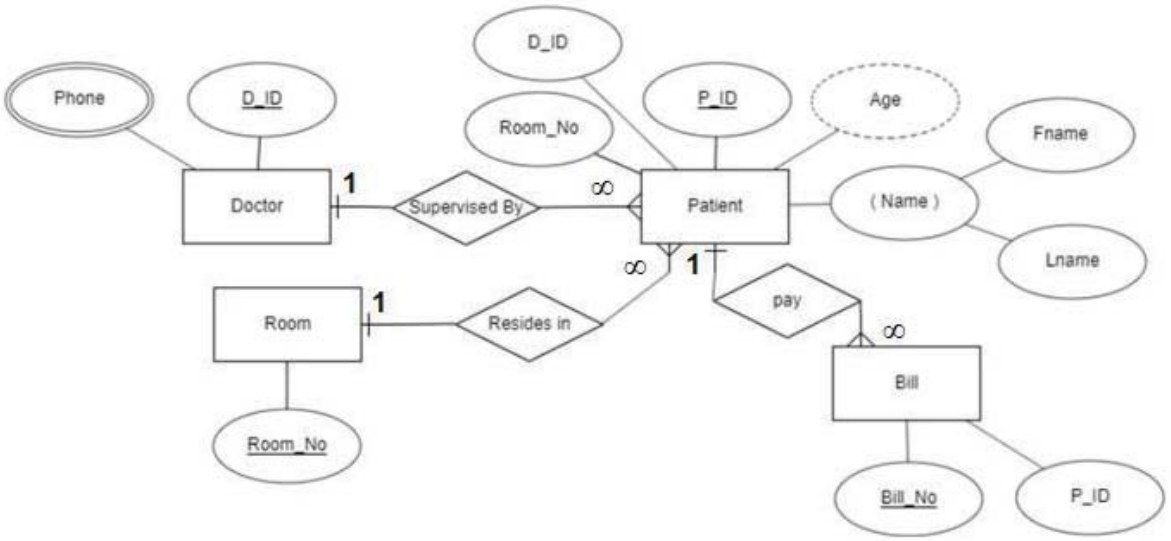
ارسم مخطط (ERD) لنظام المستشفى الآتي: في المستشفى مجموعة أطباء أخصّائيّين، كلّ طبيب يعالج عدّة مرضى في غرفة واحدة، وكلّ مريض يُصرف له عدّة فواتير علاجيّة ويعالج من مختصّ واحد، وينام المريض في غرفة داخل المستشفى دون التّنقل بين الغرف المستشفى لحين خروجه.

الحل:

الكيانات والصفات:

- كيان طبيب (**Doctor**): وحقوله (**D_ID**, Name, Phone, Department).
- كيان مريض (**Patient**): وحقوله (**P_ID**, Age, Name(Fname, Lname), Gender).
- كيان فاتورة (**Bill**): وحقوله (**Bill_No**, Doc_charge ,Room_Charge).
- كيان غرفة (**Room**): وحقوله (**Room_No**, Type).

رسم (ERD) يكون بالشكل الآتي:



سؤال:

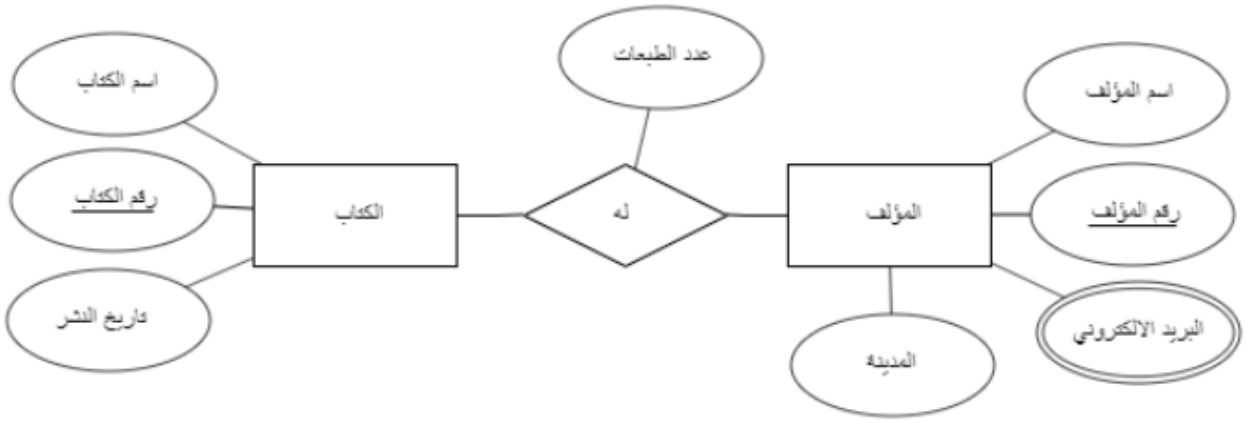
أكمل ربط بقيّة الصفات لكلّ كيان منها كما هو في وصف النظام.

نشاط (٦) نموذج ERD لنظام البقالة

بالعودة إلى نظام البقالة سابقاً، ارسم نموذج (ERD) له بشكلٍ كاملٍ.

نشاط (٧) مخطط ERD

نفّذ عملية التحويل المناسبة على مخطط (ERD)، أدناه ليصبح سليماً.



التسوية (Normalization)

نشاط (٨) تسوية البيانات.

لاحظ الكيان أدناه والذي يمثّل جدول الزّبائن في نظام البقالة، ثمّ ناقش الأسئلة التي تليه.

Customer

Custom_Name	Custom_phone	Custom_ID	Bill_TotalCost	BillPay_Type	BillPay_Date	Bill_ID
ابراهيم محمود	11112222	2	211	نقدي	2019-05-22	100
فادي فرح	12341234	4	500	شيك	2020-04-03	200
جهاد نايف	44332211	6	800	شيك	2020-09-10	300
سامي محمد	11111111	1	1200	كمبيالة	2020-12-11	400
مهند خليل	11223344	3	50	بطاقة آلية	2019-11-07	500
مهند خليل	11223344	3	70	نقدي	2019-03-04	600

- إذا أراد الموظف تعديل اسم الزبون صاحب الرقم (3)، كم من المرات سيتمّ التّعديل؟
- عند قيام زبون بعملية شراء، ما هي البيانات التي سيتم تخزينها؟ وما هي المشكلة المتوقّعة ؟
- عند إجراء حذف عملية شراء لزبون ما، هي المخاطر المتوقّعة؟

لاحظتَ بعد تنفيذ النّشاط ظهور بعض المشكلات عند تنفيذ العمليّات على الكيان؛ كالإضافة والحذف والتّعديل، وهذا ما تهدف إليه عمليّة تسوية الكيانات لحلّه دون التأثير على البيانات الأساسيّة، وهذا يتطلّب المرور على نماذج التّسوية.

نماذج التسوية Normalization Forms

نموذج التسوية الأوّل First Normal Form-(1NF)

نموذج تتحدّد فيه قيمة واحدة غير مركّبة لتقاطع السجل مع الصفة داخل الجدول، وهذا يعني إزالة الكيانات المركّبة بحيث لا يتواجد كيان داخل كيان آخر.

مثال لاحظ كيان المورّدين الآتي:

Supp_ID	Supp_Name	Supp_Phone
100	Ahmad	125487,225656
200	mohammad	252525,363636

يظهر من خلال الجدول أنّ رَقَمَيْ هاتف أُدخِلا في نفس الصفة، وهذا غير مقبول في بنية قواعد البيانات، وبالتالي؛ يعدّ الكيان في حالة النموذج الصفري (0NF)، حيث يجب فصل أرقام الهاتف كما في الجدول الآتي:

Supp_ID	Supp_Name	Sup_Phone1	Supp_Phone2
100	Ahmad	125487	225656
200	mohammad	252525	363636

سؤال اقترح حلّا آخر لمشكلة رقمَيْ الهاتف في الحقل الواحد.

نموذج التسوية الثاني (Second Normal Form -2NF)

تأتي خطوة التّحقق من شروط نموذج التسوية الثاني بعد التأكّد من شكل التسوية الأوّل؛ أي أنّ العلاقة تحقّق الشكل (1NF)، فنموذج التسوية الثاني عبارة عن علاقة لنموذج التسوية الأوّل، بحيث ترتبط كلّ خاصّية من خصائص الكيانات -غير المفتاح الأساسي- وظيفيّا مع المفتاح الأساسي؛ أي يجب اعتماد كلّ خاصّية من خصائص الكيانات بشكلٍ كاملٍ على المفتاح الأساسي، أمّا إذا اعتمدت الخاصّية بشكلٍ جزئي غير كامل فهذا يعني بالضرورة أنّه لا توجد علاقة في صيغة التسوية الثاني، ويكون غالباً ناتج عن وجود مفتاح أساسي مركّب تمّ الإعتماد عليه.

مثال ادرس الكيان الآتي:

Customer

Custom_Name	Custom_phone	Custom_ID	Bill_TotalCost	BillPay_Date	BillPay_Type	Bill_ID
ابراهيم محمود	11112222	2	211	2019-05-22	نقدي	100
فادي فرح	12341234	4	500	2020-04-03	شيك	200
جهاد تايف	44332211	6	800	2020-09-10	شيك	300
سامي محمد	11111111	1	1200	2020-12-11	كمبيالة	400
مهند خليل	11223344	3	50	2019-11-07	بطاقة آلية	500
مهند خليل	11223344	3	70	2019-03-04	نقدي	600

يُعدّ الكيان السابق في الصيغة المعيارية الأوّلى (1NF)، وليس في الصيغة المعياريّة الثانية (2NF)، حيث أنّ هناك اعتماديّة كلّية داخل الكيان؛ بحيث أنّ رقم الزبون (Custom_ID) يحدّد اسم الزبون(Custom_Name)، ورقم هاتفه (Custom_Phone)، وأنّ هناك اعتماديّة جزئيّة حيث أنّ رقم الزبون (Custom_ID) مع رقم الفاتورة (Bill_ID) يحدّدان الكميّة المُشتراة (Bill_TotalCost)، وطريقة الدفع (BillPay_Type) وتاريخ الدفع (BillPay_Date).

ولتحويل هذا الجدول إلى الصيغة المعيارية الثانية لا بدّ من فصله إلى كيانين، كما في الشكل الآتي:

Custom_ID	Custom_Name	Custom_Phone
1	سامي محمد	11111111
2	ابراهيم محمود	11112222
3	مهند خليل	11223344
4	فادي فرح	12341234
6	جهاد نايف	44332211

Bill_ID	Custom_ID	BillPay_Type	BillPay_Date	Bill_TotalCost
100	2	نقدي	22-05-2019	211
200	4	شيك	03-04-2020	500
300	6	شيك	10-09-2020	800
400	1	كمبيالة	11-12-2020	1200
500	3	بطاقة آلية	07-11-2019	50
600	3	نقدي	04-03-2019	70

نشاط (٩) نموذج التسوية الثاني (2NF).

ادرس الجدول أدناه، وقم بتحويله إلى الصيغة المعيارية الثانية (2NF):

رقم الموظف	اسم الموظف	المؤهل العلمي	الحالة الاجتماعية	اسم الابن	تاريخ ميلاد الابن
111	محمد	ماجستير	متزوج	عمر	1990/1/12
111	محمد	ماجستير	متزوج	سارة	1993/5/18
222	أحمد	دكتوراه	أعزب		
333	فاطمة	بكالوريوس	متزوج	علي	1987/7/8
444	أسامة	ماجستير	متزوج	عائشة	1995/2/5
444	أسامة	ماجستير	متزوج	محمد	1998/6/3
444	أسامة	ماجستير	متزوج	عمر	2001/1/5

نموذج التسوية الثالث (3NF -Third Normal Form)

تهدف التسوية الثالثة إلى انهاء أيّ اعتماديّة متعدّية وتحويل العلاقة إلى اعتماديّة كاملة بعد التأكّد من شكل العلاقة في التسوية الثاني، ويتمّ ذلك بفصل أعمدة الإعتمادية المتعدّية عن بقيّة صفات الكيان.

ملاحظة:

الاعتمادية الوظيفية: إذا كانت الصفة A تحدّد الصفة B، وكانت الصفة B تحدّد الصفة C؛ فإنّ الصفة A تحدّد الصفة C كونها ناتج العلاقة الآتية:

(A→B و B→C) فإنّ (A→C)

يجب التأكّد من أنّ جميع الصّفات داخل الكيان تعتمد بشكل مباشر على المفتاح الأساسي، ولا تعتمد على أيّ صفة أخرى غير مفتاحيّة.

مثال في كيان المنتجات (Product_tbl)، كانت البيانات على الشّكل الآتي:

Pro_SellingPrice	Pro_PurchasePrice	Pro_Barcode	Pro_Type	Pro_Name	Pro_ID
100	50	1111111111	لحوم	لحم خاروف	1000
20	12	5555555555	ألبان	لبن رايب	2000
3	1	4444444444	مثلجات	بوظة علب	3000
3	1	8888888888	معلبات	فول مدمس	4000
4	2	3333333333	بهار	فلفل أسود	5000
10	6	7777777777	ألبان	لبنة	6000

بعد دراسة الكيان السابق، يلاحظ أنّ هناك اعتمادية متعدّية داخل الكيان بحيث أنّ صفة اسم المنتج (Pro_Name) يعتمد على المفتاح الأساسي (Pro_ID)، وأنّ صفة نوع المنتج (Pro_Type) التّابع لها المنتج يعتمد على صفة (اسم المنتج)، وبالتالي:

Pro_ID → Pro_Name

Pro_Name → Pro_Type

وبالتالي: Pro_ID → Pro_Type يُعدّ ذلك اعتماديّة متعدّية يجب إزالتها في شكل التسوية الثالث، ويتمّ إزالتها عن طريق فصلها إلى كيانَين، ليكون الناتج كما يأتي:

Pro_SellingPrice	Pro_PurchasePrice	Pro_Barcode	Category_ID	Pro_Name	Pro_ID
100	50	1111111111	3	لحم خاروف	1000
20	12	5555555555	1	لبن رايب	2000
3	1	4444444444	2	بوظة علب	3000
3	1	8888888888	5	فول مدمس	4000
4	2	3333333333	4	فلفل أسود	5000
10	6	7777777777	1	لبنة	6000

Category_Name	Category_ID
ألبان	1
مثلجات	2
لحوم	3
بهار	4
معلبات	5
مواد تنظيف	6
بقوليات	7

أسئلة الدرس

1. ضع دائرة حول رمز الإجابة الصحيحة فيما يلي:

- ما المفتاح المُستخدم للرّبط المنطقي بين الكيانات؟

أ) المفتاح الأساسي ب) المفتاح الأجنبي ج) المفتاح الأعظم د) المفتاح الثانوي

- ما نوع نموذج التسوية عند وجود أكثر من قيمة داخل الصفة؟

أ) 0NF ب)1NF ج)2NF د)3NF

2. وضّح المقصود بالمصطلحات الآتية: التّسوية، السّجل، الصفة.
3. اذكر خطوات تحليل النظام.
4. علّل: تسمية النّموذج العلائقي بهذا الإسم.
5. في نظام مكتبة، يوجد عدّة كتب لعدّة مؤلفين، وتقوم المكتبة بإعارة الكتب للطلبة، ارسم (ERD) لذلك النظام، مع العلم أنّ:

- المكتبة يسجّل فيها عدّة مشتركين.
- يمكن للمشترك إعارة عدّة كتب.
- للكتاب عدّة مؤلّفين، والمؤلّف يمكن أن يشترك بتأليف عدّة كتب.
- هناك مدّة للإعارة لا تتعدّى 15 يوماً للكتاب الواحد.

الدرس (3)

بناء قاعدة بيانات (Sql Server)

قاعدة بيانات "SQL Server"

قواعد البيانات من اللّبنات المهمّة في حياة كل مبرمج حقيقي، حيث يوجد الكثير من البرامج والتّطبيقات لإدارتها والتّعامل معها. وتختلف البرامج المستخدمة حاليّا من حيث حجم البيانات المخزّنة ودرجة الأمان للبيانات وسعرها وسهولة استخدامها ومدى انتشارها بين أوساط المبرمجين وعدد المستخدمين إضافة الى مجموعة الأدوات التي يوفّرها البرنامج.

في هذا الدرس، سنتعرّف على برنامج (Microsoft SQL Server) كأحد برامج شركة مايكروسوفت الرائدة في ادارة قواعد البيانات والمستخدمة في كثير من الشركات والمؤسسات الكبيرة نسبيا؛ حيث يتمتّع البرنامج بقدرات ومميّزات ساهمت في انتشاره بين أوساط المبرمجين، منها:

- استيعاب حجم تخزيني للبيانات يتعدّى 2 جيجابايت بكثير.
- درجة الأمان العالية لحفظ البيانات من العبث والتلاعب بها.
- التّعامل مع عدد هائل من المستخدمين.
- سرعة معالجة البيانات مقارنة مع كثير من البرامج والتطبيقات (كبرنامج Access).

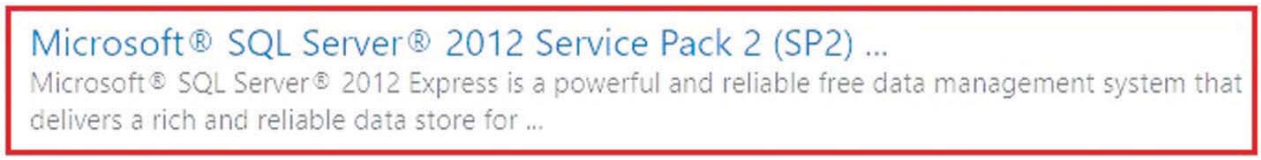
تنصيب برنامج (Microsoft SQL Server 2012 Express)

نشاط (١) تنصيب البرنامج.

- قم بالدخول الى موقع تحميل البرنامج من الرّابط الآتي:

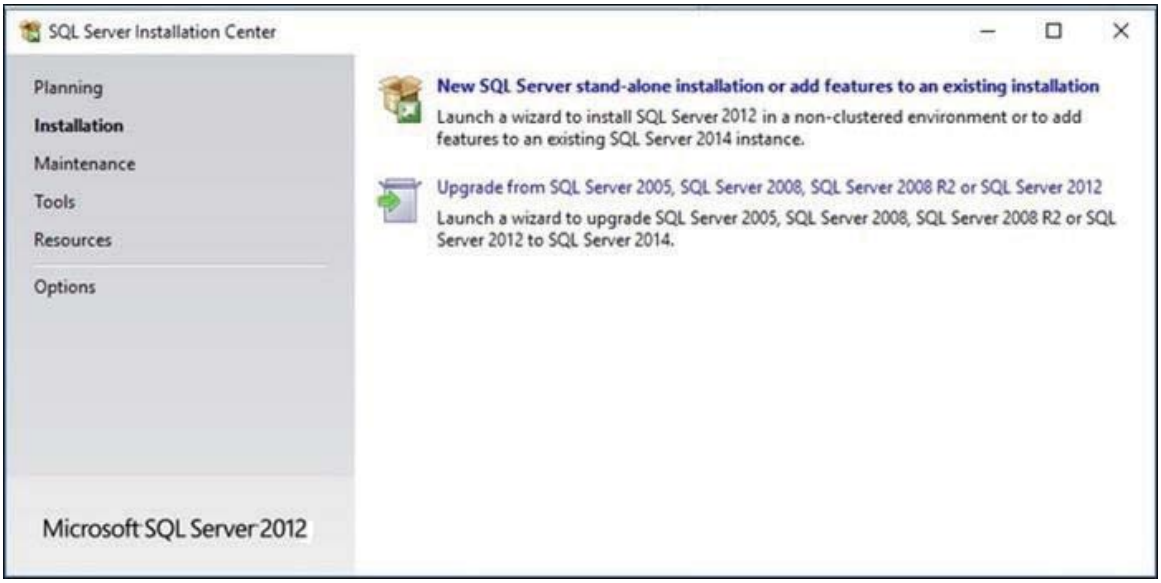
https://www.microsoft.com/en-us/download

- كتابة (SQL Server 2012 Express download) في صندوق البحث للموقع.
- اختيار ما يأتي:

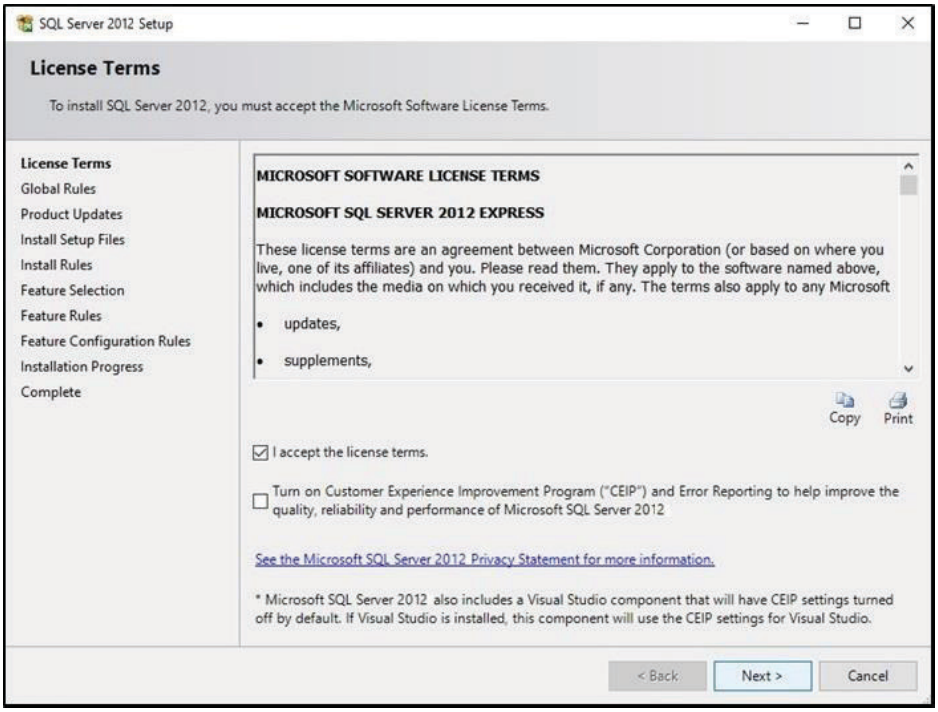


- تحديد نسخة SQL Server 2012 Express حسب نوع نظام التشغيل لديك، وانتظار التّحميل كما يأتي:

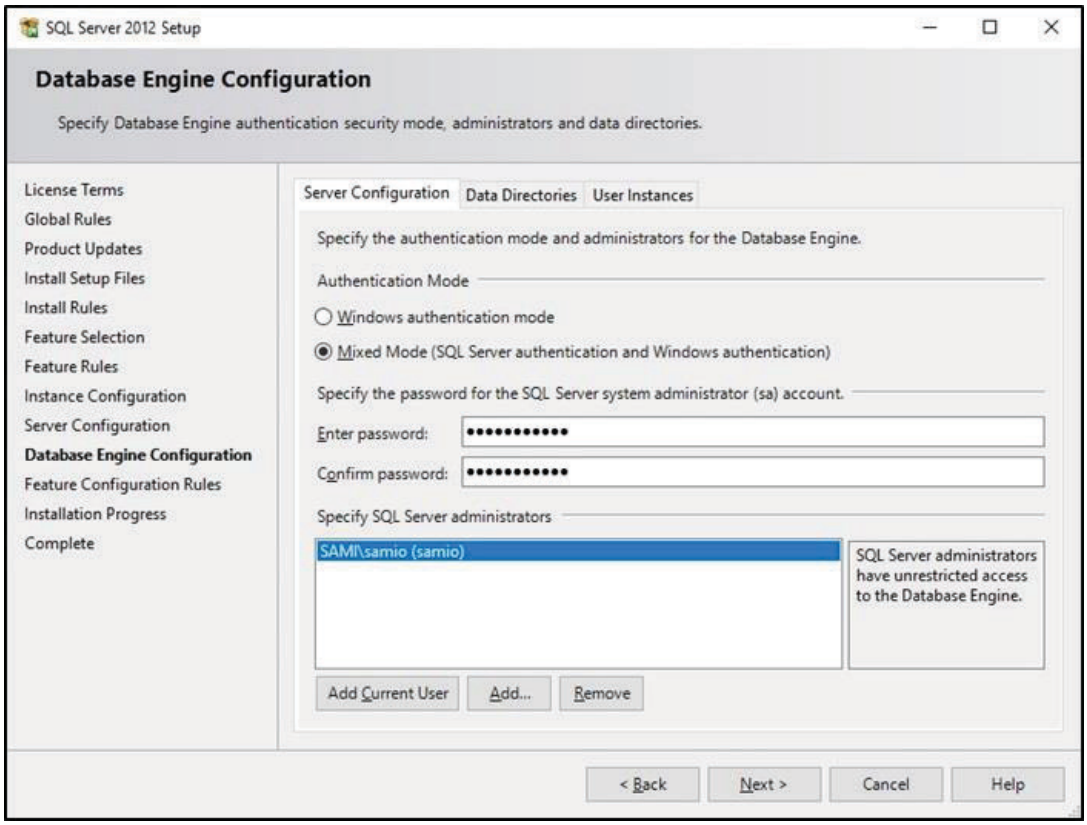
- الضّغط بزر الفأرة مرتين متتاليتين لفك ضغط البرنامج وظهور الواجهة الآتية:



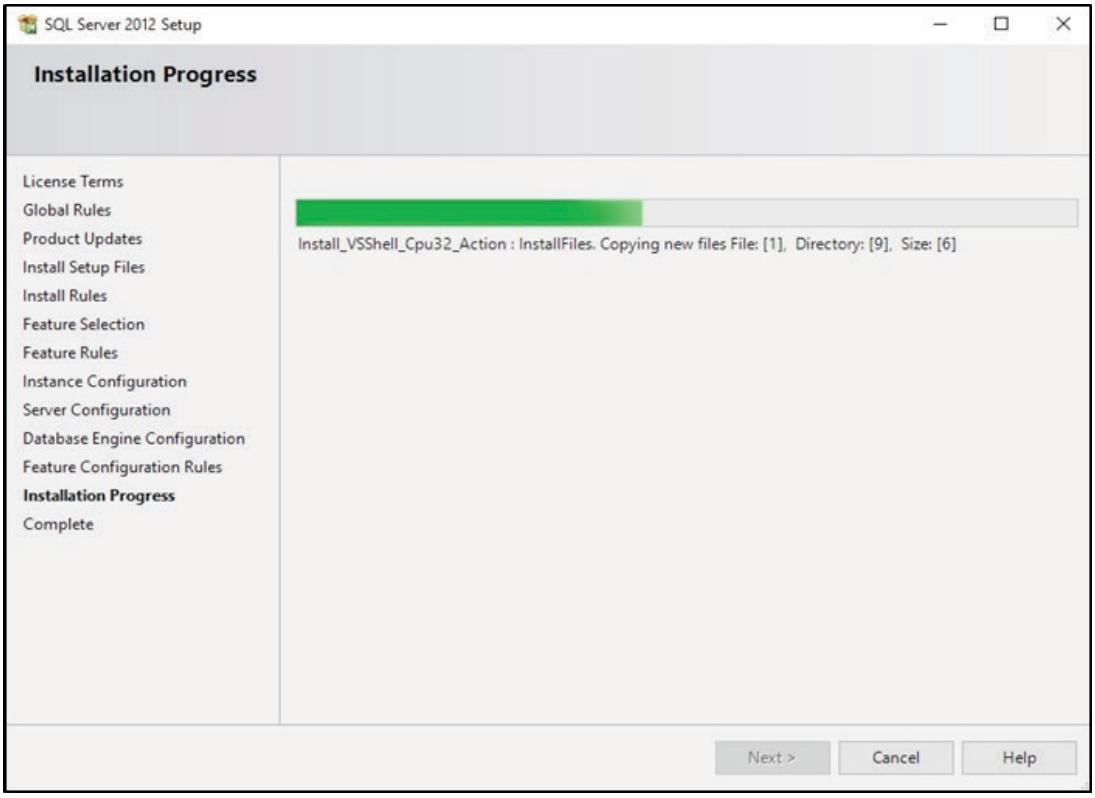
- اختار الخيار الأول (Stand-alone Installation)، والانتقال للواجهة الاتية:

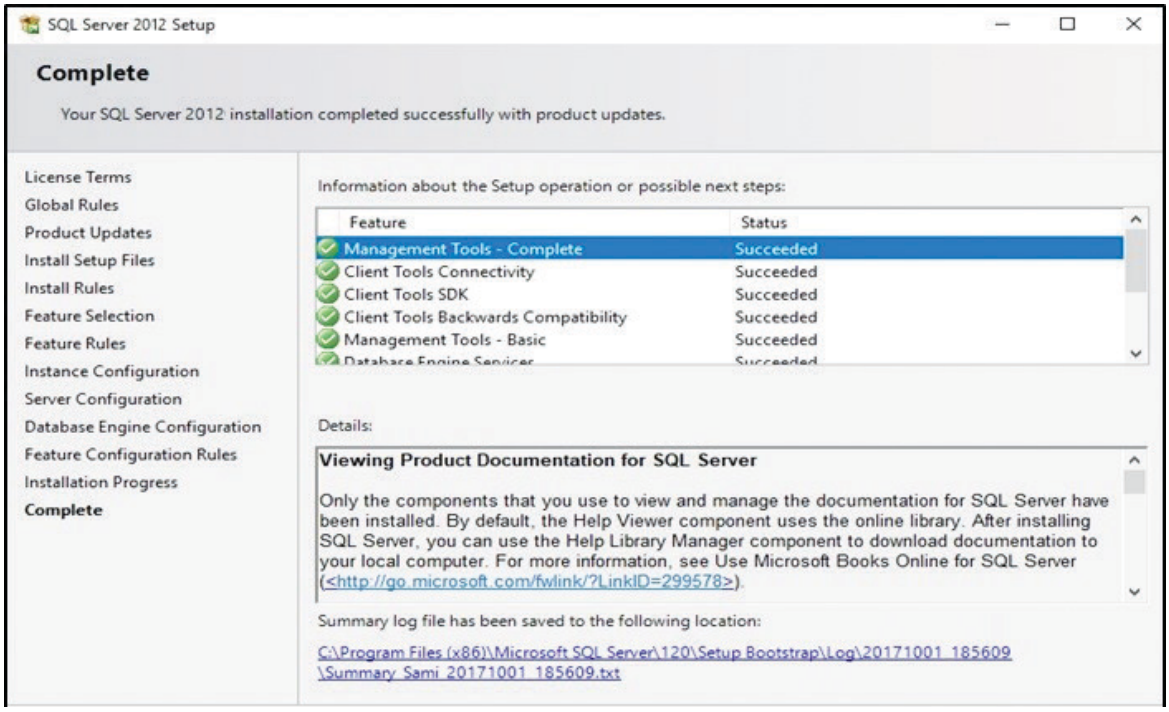


- الاستمرار بضغط زر التّالي (Next)، حتى ظهور الشاشة الآتية:



- الاستمرار في ضغط زر التالي حتى انهاء تنصيب البرنامج على جهاز الحاسوب.





سؤال: ما الفرق بين مستوى الحماية: (Windows Authentication) و (SQL Server Authentication)؟

- البحث من قائمة (ابدأ) عن التطبيق (Microsoft SQL Server 2012 Management Studio) في مجلد (Microsoft SQL Server 2012) كما في الشكل:

- الاتّصال بجهاز السيرفر (جهاز الحاسوب الحالي) مع تحديد نوع المصادقة (authentication) كما في الشكل أدناه.

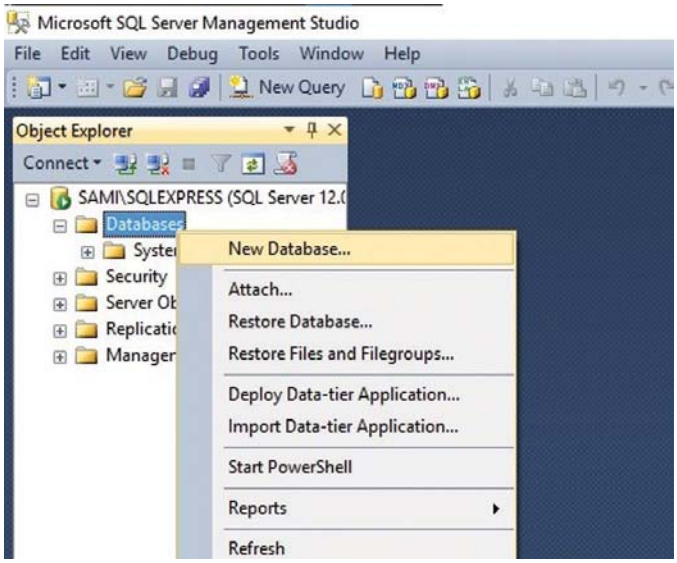
إنشاء قاعدة بيانات جديدة.

هناك طريقتان لإنشاء قاعدة بيانات في برنامج (SQL Server) كما يأتي:

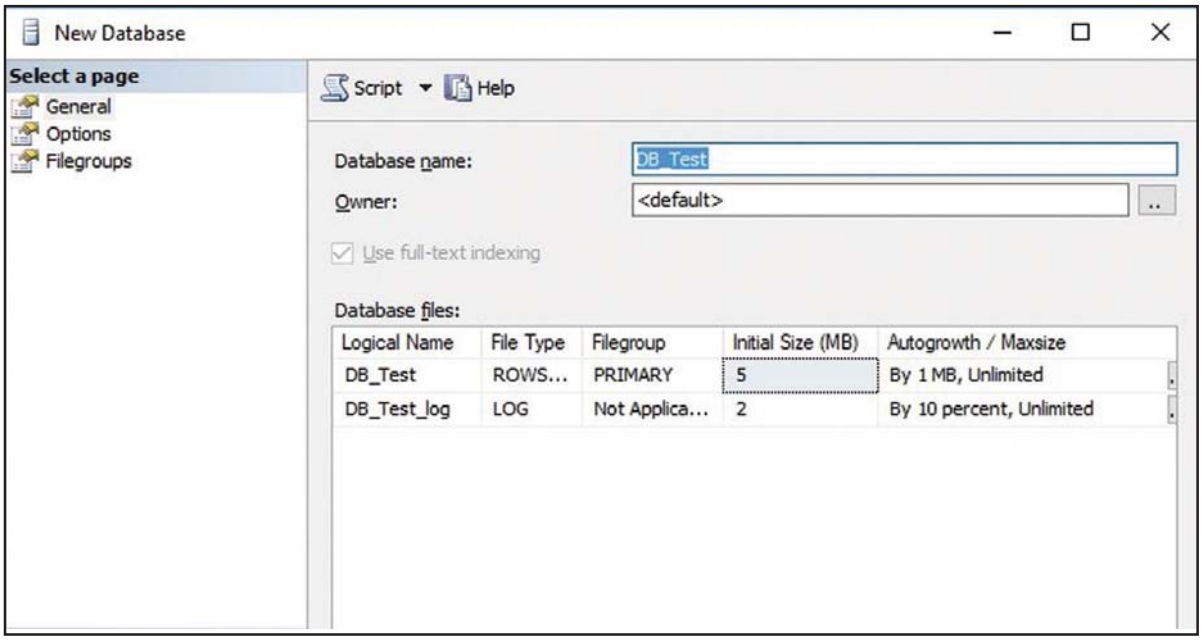
أوّلاً: المعالج Wizard

ويتم من خلال اتباع الخطوات الاتية:

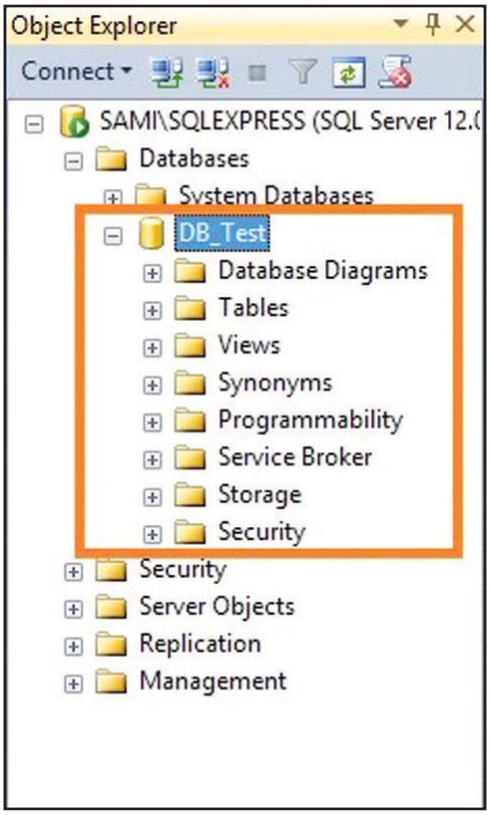
١ من خلال ضغط زر الماوس الأيمن على مجلد Database في شاشة (Object Explorer) ثم اختيار (New Database...) كما في الشكل ادناه.



٢ تسمية قاعدة البيانات المراد انشائها كما في الشكل:



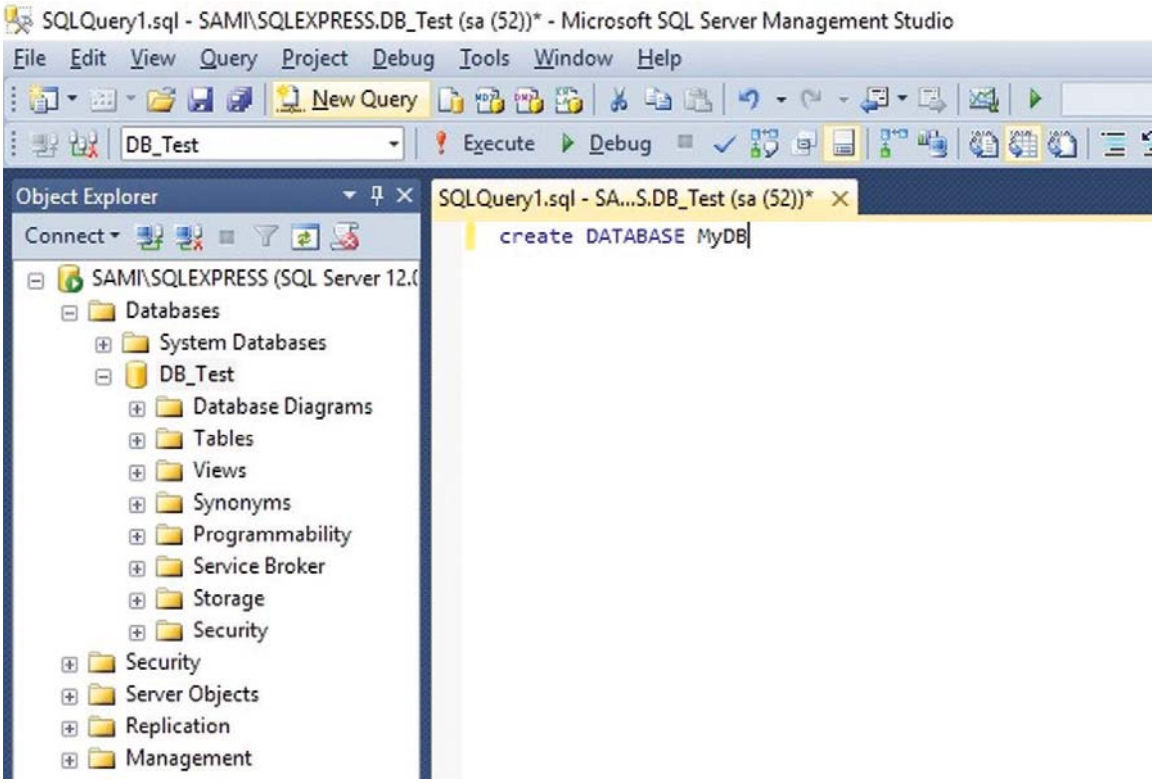
٣ ظهور قاعدة البيانات في شاشة (Object Explorer) بالتسمية السابقة كما في الشكل أدناه.

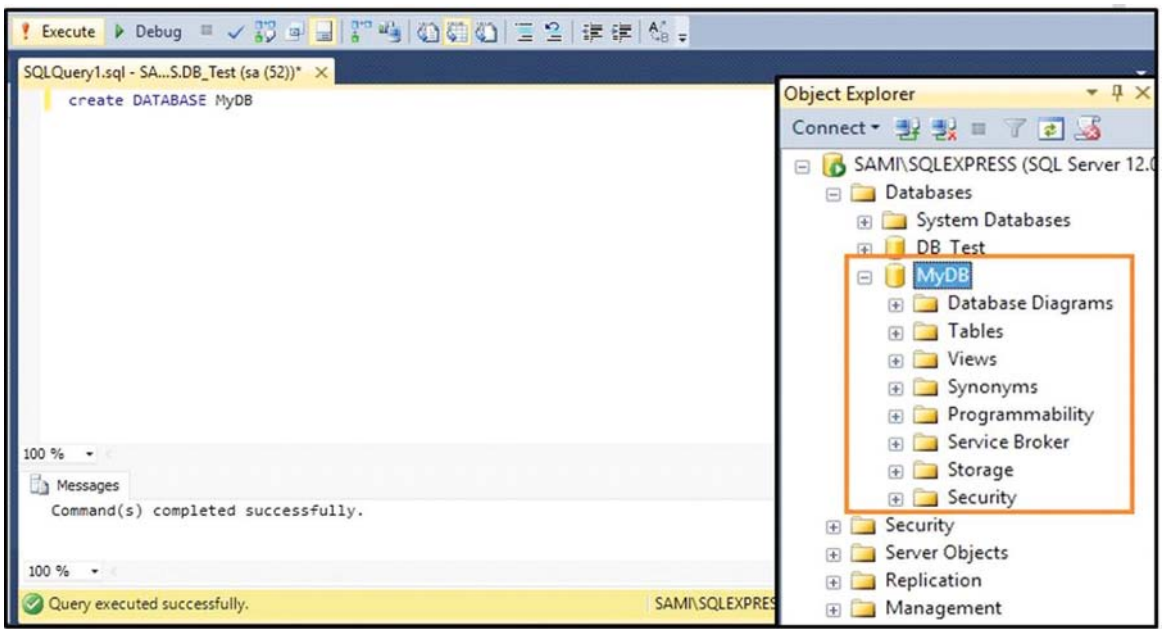


فكّر: كيف يمكن حذف قاعدة البيانات من خلال المعالج (Wizard)؟

ثانياً: طريقة الأوامر (SQL Transaction):

من خلال اختيار (New Query) وكتابة الأمر كما في الشكل:





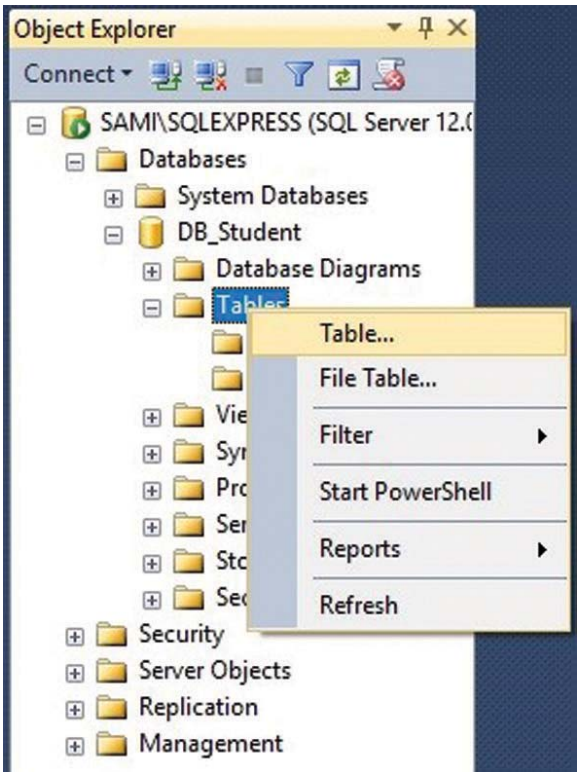
قم بإنشاء قاعدة بيانات باسم (Market) باستخدام SQL Server 2012 Express.

• إدراج كيان إلى قاعدة البيانات.

الكيان يشكل إحدى مكوّنات قاعدة البيانات الهامّة، لأنّه المستودع الرئيس للبيانات داخله، وكما تعلّمت في الدّرس السّابق من رسم (ERD) وتمثيل العلاقات بين الكيانات، وتحديد المفاتيح الرئيسية والأجنبية، جاء دور الجانب العملي في استخدام أحد برامج نظم إدارة قواعد البيانات في بناء الكيانات على الحاسوب.
هناك طريقتان لإدراج الكيان في قاعدة البيانات:

أولاً: عن طريق المعالج Wizard

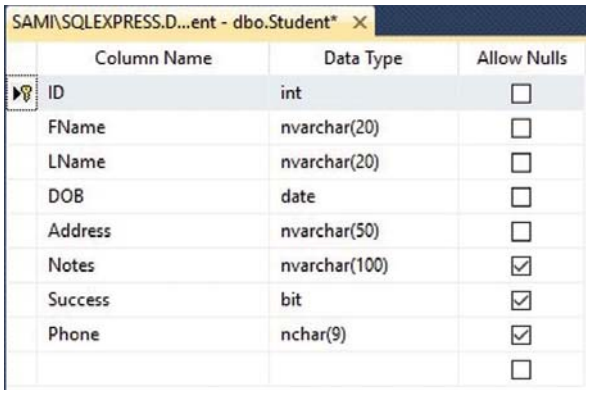
ويكون من خلال ضغط زر الماوس الأيمن على مجلد (Tables) الموجود داخل قاعدة البيانات (DB_Student) كما في الشكل المجاور.

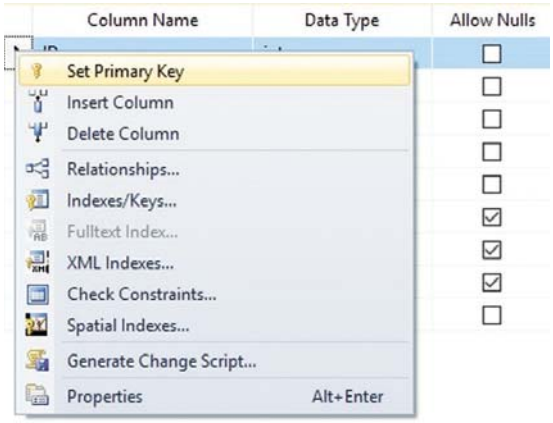


• إدراج صفات للكيان.

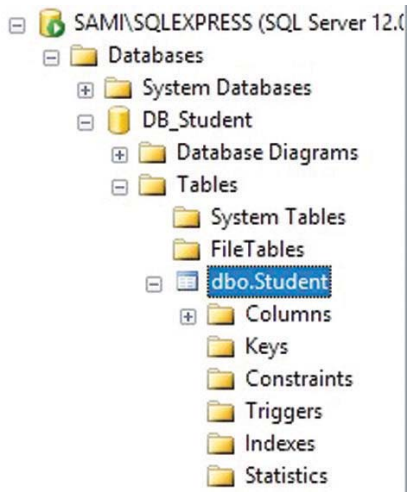
وذلك باتباع الخطوات الآتية:

- كتابة أسماء الصفات في عمود Column Name كما في الشكل المجاور.
- تحديد نوع البيانات لكل صفة من الصفات.
- تأكيد وجود قيم (Null) من عدمها من خلال وضع علامة بجانب الصفة المراد في عمود Allow Nulls كما في الشكل المجاور.





- تحديد المفتاح الأساسي للكيان من خلال ضغط زر الفأرة الأيمن على الصفة، ثمّ اختيار (Set Primary Key) كما في الشكل.

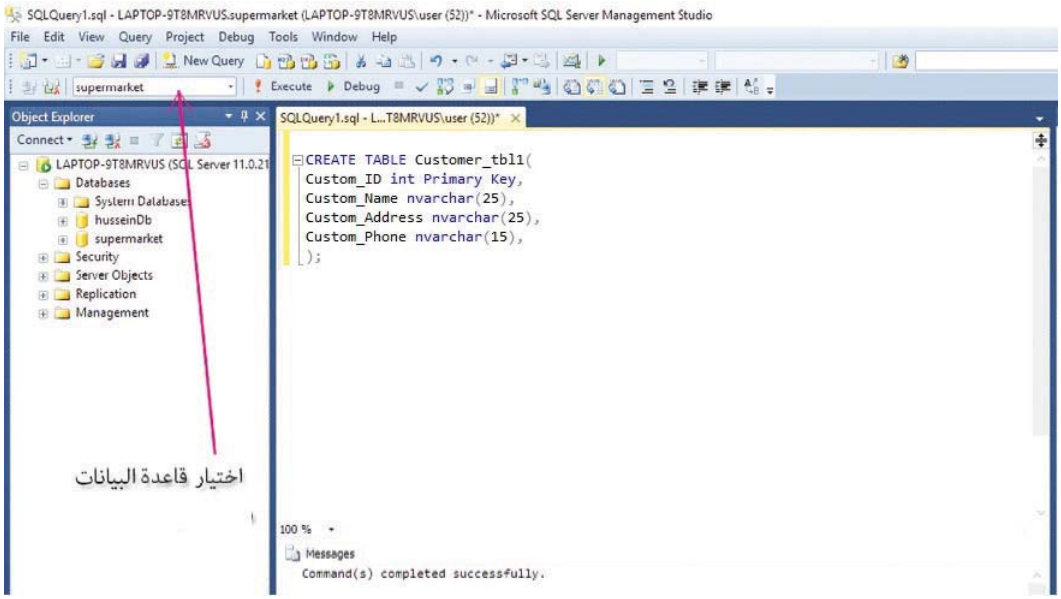


- حفظ الكيان في قاعدة البيانات باسم (Student)، فيظهر في قائمة (Object Browser) كما في الشكل المجاور.

ثانيا: عن طريق الأوامر بلغة (SQL)

ويكون من خلال اختيار زر (NewQuery)، وكتابة الأمر كما في الشاشة التالية، والتي يتمّ فيها كتابة جملة بناء الكيان وتحديد صفاته، مع ملاحظة أنّ كلّ سطرٍ ينتهي بـ(,) وآخر الجملة تنتهي بـ(;)، والصّيغة العامة على النحو الآتي:

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
  ....
);
```

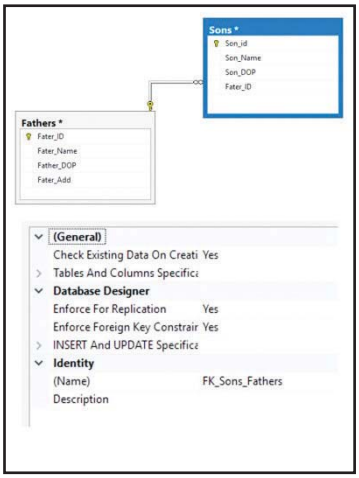


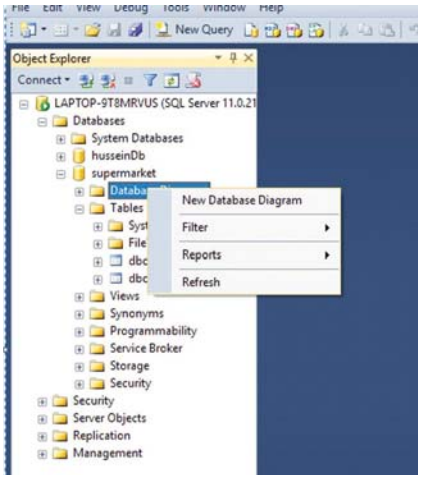
نشاط (٣) بناء كيانات نظام البقالة.

◄ مُستعيناً بما تعلّمته في الدرس السّابق، أنشئ جميع كيانات نظام (البقالة).

- **إنشاء العلاقات بين الكيانات.**

تعلّمت سابقاً أنواع العلاقات بين الكيانات، وتعرّفت أنّ علاقة (متعدّد إلى متعدّد) غير مقبولة في قاعدة البيانات؛ فيجب تحويلها. يتمّ إنشاء العلاقات على نظام (SQL server 2012) من خلال اختيار (DataBase Diagrams) والضغط على زر الماوس الأيمن، ثمّ اختيار (New DataBase Diagram)، ومن ثمّ يتمّ اختيار الكيانات لبدء إنشاء العلاقات؛ بسحب الصفة من الكيان الأول الى الصفة في الكيان الآخر مع تحديد ما اذا كانت هذه العلاقة لها تتالي الحذف والتّعديل أم لا. كما في الشّكل المجاور.





- **أنواع البيانات في SQL Server 2012**

تقسم البيانات في قواعد بيانات

SQL Server إلى أربعة مجموعات رئيسة، هي:

١ بيانات رقميّة.

نوع البيانات	الوصف	السعة التخزينية
Bigint	أرقام صحيحة	8 بايت
Int	أرقام صحيحة	4 بايت
Smallint	أرقام صحيحة	2 بايت
TinyInt	أرقام صحيحة	1 بايت
Decimal	أرقام عشرية	تعتمد على عدد المنازل العشرية 5 بايت – 17 بايت
Money	أرقام عشرية	8 بايت

٢ بيانات نصّية.

نوع البيانات	الوصف	السعة التخزينية
char [(n)]	مجموعة حروف أو أرقام أو رموز أو خليط بينهم.	n بايت
varchar [(n | Max)]		حسب المدخلات حتى 2 جيجا بايت
Nchar	مجموعة حروف أو أرقام أو رموز أو خليط بينهم إضافة الى الحروف المحلية Unicode.	n بايت
nvarchar [(n | Max)]		حسب المدخلات حتى 2 جيجا بايت

٣ بيانات تاريخ/وقت.

نوع البيانات	المدى	السعة التخزينية
Date	31/12/9999 – 1/1/0001	3 بايت
Time	23:59:59 – 00:00:00	5 بايت

٤ بيانات منطقيّة.

نوع البيانات	المدى	السعة التخزينية
Bit	Yes/No – 1/0 – True/False	1 بايت

نشاط (٤) تصميم الكيانات حاسوبيّا.

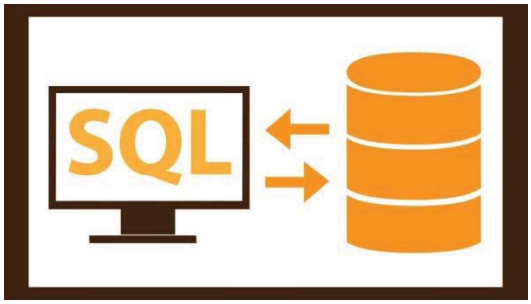
في مختبر الحاسوب، وبمساعدة معلمك، باستخدام أحد برامج نظم قواعد البيانات، نفّذ الآتي:

- أنشئ قاعدة البيانات أدناه بجميع كياناتها وصفاتها ومفاتيحها الأساسية.
- أضف أيّ صفات مناسبة الى الكيانات.
- أنشئ علاقات بين الكيانات.
- قم بإدخال عينة من البيانات إلى الكيانات.

الإستعلامات (Queries)

تستخدم الإستعلامات لإجراء البحث على الكيانات بناء على تطبيق معايير محددة ولاستعراض سجلات تطابق تلك المعايير، وذلك لإجراء عمليات محددة على تلك السجلات، مثل: العمليات الحسابية و التأكد من التغيرات التي طرأت على البيانات قبل اعتمادها والإلتزام بها.

ويمكن بناء الإستعلامات بطريقتين:



أولاً: طريقة المعالج: تقدم بعض برامج نظم قواعد البيانات واجهة رسومية، يمكن استخدامها لتكوين الإستعلام بكتابة جمل (SQL) نيابة عن المبرمج.

ثانياً: عن طريق كتابة جمل لغة الإستعلام البنيوية (SQL: Structure Query Language)

تمثل مجموعة من الأوامر تستخدم للبحث والإستعلام عن سجلات مجددة داخل قاعدة البيانات.وتنقسم لغة الإستعلام البنيوية إلى ثلاثة اقسام رئيسية:

١ لغة تعريف البيانات (DDL: Data Definition Language)

مجموعة من الأوامر لإنشاء قاعدة البيانات وحذفها، انشاء الكيانات وتعديلها وحذفها، تكوين العلاقات بين الكيانات، وقد كانت هذه العمليات في برامج نظم إدارة قواعد البيانات القديمة تتم من خلال كتابة أوامر اللغة؛ بينما في الحديثة أصبحت العمليّة اكثر سهولة حيث يستخدم واجهات رسومية.

٢ لغة التحكم بالبيانات (DCL: Data Control Language)

مجموعة من الأوامر لتحديد صلاحيات مستخدمي قاعدة البيانات، من حيث وصول مكوناتها، واستخدامها.

٣ لغة معالجة البيانات (DML: Data Manipulation Language)

مجموعة من الأوامر لإجراء عمليات على الكيانات (ادخال سجلات، تحديث سجلات، حذف سجلات، استرجاع سجلات) بناء على معيار معين، ولتلك العمليات اثر مباشر على البيانات، واسهامها في انشاء التقارير المطلوبة.

- **استعلامات لغة معالجة البيانات (DML)**

وتتكون من الأوامر الآتية:

الأمر	الوصف
الإدخال INSERT INTO	إضافة سجلّ أو أكثر إلى الكيان
الحذف DELETE	حذف سجلّ أو أكثر من الكيان
التّحديث UPDATE	تحديث البيانات داخل الكيان
الإختيار/التّحديد SELECT	إرجاع نتائج مجموعة سجلّات من كيان أو أكثر

إضافة قيم للكيان (INSERT INTO).

أوّلاً: طريقة المعالج.

ضغط زر الفأرة الأيمن على كيان قاعدة البيانات، ثمّ اختيار (Edit Top 200 Rows) لإظهار شاشة الإدخال، ثمّ الشروع في إدخال البيانات.

ثانياً: طريقة الأوامر بلغة SQL

وتتم من خلال اختيار (New Query) من الشريط العلوي في برنامج (Sql Server 2012) ثمّ الشروع في كتابة الأوامر الخاصة بلغة (SQL)، وهذا ينطبق على جميع الأوامر اللّاحقة.

ولإضافة البيانات إلى كيان محدّد نكتب الجملة الآتية:

الصيغة العامة	الوصف
INSERT INTO table_name (column1, column1,...) VALUES (value1, value2,...);	تحديد الصفات المكوّنة للسجل في الكيان. القيم المراد إدخالها بالترتيب.

مثال أدخل لكيان الزبون (Customer_tbl) البيانات الآتية: اسم الزبون (Custom_Name) أحمد، وعنوانه (Custom_Address) رام الله، ورقم هاتفه (Custo_Phone) 056805993 ، ورقمه (Custom_ID) 2809.

الحل: لإضافة بيانات الزبون في الكيان نكتب الأمر الآتي:

```
INSERT INTO Customer_tbl(Custom_ID,Custom_Name,Custom_Address,Custom_Phone)
VALUES (2809,'احمد','رام الله','056805993');
```

نشاط (٥) عملية الإضافة.

أَعِد تنفيذ المثال السابق بقيمٍ أخرى مع إبقاء (Custom_ID) بنفس القيمة السابقة، هل تمّت الإضافة؟ لماذا؟

لعلك لاحظت بعد تنفيذ النشاط السابق أنّ عمليّة الإضافة قد لا تتمّ بسبب التّكرار في قيم المفتاح الأساسي للكيان.

حذف البيانات من الكيان (DELETE)

• **أولاً: طريقة المعالج:** ضغط زر الفأرة الأيمن على كيان قاعدة البيانات، ثم اختيار الايقونة DELETE لحذف الكيان.

يستخدم هذا الامر لحذف سجل أو جميع السّجلات من الكيان، والصّيغة العامّة له كالآتي:

الصيغة العامة	الوصف
DELETE FROM table_name WHERE [condition/s];	تحديد الكيان المراد حذف السجلات منه. الشرط/الشروط اللّازم تحقيقها لإجراء الحذف.

مثال اكتب جملة SQL لحذف الفاتورة التي رقمها (311).

الحل:

```
DELETE FROM BillSale_tbl
WHERE  Bill _ID = 311;
```

إذا تمّ إلغاء الشرط (WHERE) فان جميع السجلات سوف تحذف من الكيان كما يأتي:

```
DELETE  From BillSale_tbl;
```

• **ثانياً: طريقة الاوامر بلغة SQL:**

يُستخدم هذا الاستعلام لتغيير قيمة سجل معين أو عدّة سجلّات داخل كيان في قاعدة البيانات، ولفهم آليّة عمل استعلام التحديث قم بالنشاط الآتي:

نشاط (٦) استعلام التحديث

في مختبر الحاسوب، وبمساعدة معلمك، قم بفتح قاعدة بيانات البقالة(Market) السابقة ونفّذ ما يلي:

- اقترح طريقة لتغيير اسم زبون الى آخر.
- لو طلب منك إضافة نسبة معينة على جميع الفواتير المتواجدة في قاعدة البيانات، ما هي الآلية التي ستتبعها؟

لعلك لاحظت بعد تنفيذ النشاط السابق أنّ هذه المهمة مرهقة وتستنزف الوقت، والنظام المحوسب وُجد لحلّ مثل هذه المشاكل بالسّرعة المطلوبة، لذلك سنقوم بحلّ هذه المشكلة من خلال استعلام التّعديل، وصيغته العامة كما يلي:

الصيغة العامة	الوصف
UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE [condition];	تحديد الكيان المراد تعديل بياناته. القيمة/القيم المراد تعديلها للصفات. شرط/شروط يجب تحقّقها لإجراء التعديل.

هنا، يتمّ تحديث السجلات المتوافقة مع الشرط (column_name = some_value) في الكيان (table_name) بحيث يتمّ تغيير قيمة الصفة (column1_name) إلى القيمة الجديدة (value1)...، ويمكن أن يكون حقل الشرط ضمن الصفات المراد تحديثها.

إذا تمّ إلغاء الشرط فإنّ التحديث سيطبق على جميع سجلات الكيان.

مثال اكتب جملة SQL لتغيير أسماء جميع الزبائن إلى الاسم (محمد سليمان) في كيان (Customer_tbl).

الحل:

```
UPDATE Customer_tbl SET Custom_Name='محمد سليمان';
```

نشاط (٧) تحديث بيانات كيان.

في مختبر الحاسوب، بمساعدة معلمك، افتح قاعدة بيانات البقالة (Market)، ثم حدّث بيانات الصفة سعر البيع للمنتج (Pro_SellingPrice) في الكيان منتج (Product_tbl) إلى القيمة 17 عندما تكون قيمة رقم المنتج (Pro_ID) تساوي 4.

- **استعلام الاختيار/التحديد (SELECT)**

نستطيع من خلال هذا الاستعلام عرض جميع الصفات أو تصفيتها بناءً على شرطٍ معيّن، ولفهم آليّة عمل استعلام الاختيار، قم بالنشاط الآتي:

نشاط (٨) استعلام الاختيار

في مختبر الحاسوب، وبمساعدة معلمك، نفّذ الآتي:

- افتح الكيان زبون (Customer_tbl) من نظام البقالة، ثم أجب عن الأسئلة التي تليه.

Custom_Phone	Custom_Address	Custom_Name	Custom_ID
11111111	جنوب الخليل	سامي محمد	1
11112222	رام الله	ابراهيم محمود	2
11223344	جنين	مهند خليل	3
12341234	الخليل	فادي فرح	4
43214321	الخليل	أمجد حسني	5
44332211	رام الله	جهاد نايف	6
44443333	جنين	تهاني عبود	7
44444444	القدس	فاطمة حسام	8
55555555	القدس	ولاء عمر	9

- ما عدد الزبائن داخل البقالة؟
- ما أسماء الزبائن الذين يقطنون مدينة "الخليل"؟

لعلك لاحظت بعد تنفيذ النشاط السابق أنّ عملية البحث مرهقة وتستنزف وقتاً طويلاً، وحلّ هذه المشكلة يكون من خلال استعلام الإختيار حسب الصيغة العامة الآتية:

الصيغة العامة	الوصف
SELECT column1,column2,... FROM TABLE1,TABLE2,... WHERE [condition/s] Order BY column1[Ascending | Descending], column2[Ascending | Descending] , ... ;	اختيار صفة أو أكثر. اختيار من كيان أو أكثر. شرط/شروط اختيار السجلات من الكيان. ترتيب الصفوف المختارة، بناءً على صفة أو أكثر تصاعدياً أو تنازلياً.

ملاحظة:

في عبارة (Order By)، وجود الاقواس [ ] يعني أنّ تحديد النمط اختياري، والنمط التلقائي يكون (Ascending) تصاعديّ، ويمكن اختصار كلمة التصاعدي ب (Asc) والتنازلي بـ (Desc).

من الجدير بالذكر أنّ نتيجة هذا الأمر تعرض بيانات الصفات على الشاشة بشكل كيان مؤقت، أعمدته الصفات المختارة في الأمر من الصفات الأصلية للكيان، وسجلاته يحدّدها الشرط في عبارة (WHERE).

مثال اختر جميع الصفات في كيان (Product_tbl)

الحل: تكتب الصيغة على الشكل الآتي:

```
SELECT *
FROM Product_tbl
ORDER BY Pro_ID desc;
```

يمكن اختيار صفات محدّدة لعرضها دون الأخرى، كما في المثال الآتي:

مثال اكتب امر SQL لعرض سجل رقم صاحب السيارة مع اسمه فقط.

الحل: تكتب الصيغة على الشكل الآتي:

```
SELECT Pro_ID, Pro_Name
FROM Product_tbl;
```

يمكن تغيير عنوان صفة ناتج الاستعلام بدلاً من العنوان الأصلي في صفة الكيان، مثل عرض (اسم المنتج) بدلاً من عرض (Pro_Name) في كيان (Product_tbl) من خلال كتابة جملة الأمر الآتية:

```
SELECT Pro_Name AS "اسم المنتج"
FROM Product_tbl;
```

سؤال هل تغيّر الاسم الأصلي للصفة (Pro_Name) في كيان (Product_tbl) بعد استخدام المعامل (AS)؟

- **عمليات حسابية على الصفات.**

يمكن استخدام المعاملات الحسابية (+، -، *، /) لإجراء بعض العمليات الحسابيّة على قيم الصفة داخل الكيان، ولكن، لا تنسَ استخدام المعامل (AS) لإعطاء الحقل الجديد اسماً مختلفاً.

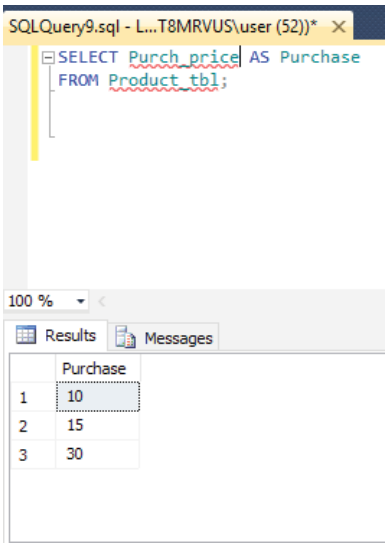
مثال اكتب استعلام لعرض سعر الشراء (Pro_PurchasePrice) مضروباً بالعدد 10، وإظهاره بمسمّى (سعر الشراء بعد التعديل).

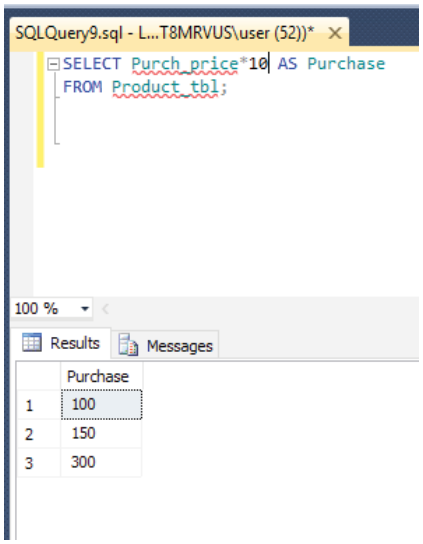
الحل:

يتم تنفيذ الامر السابق باستخدام الصيغة الآتية:

```
SELECT ((Pro_PurchasePrice) *10) AS "سعر الشراء بعد التعديل"
FROM Product_tbl;
```

تكون النتيجة كما يلي: قبل بعد





سؤال

قم بعمل النتيجة السابقة باستخدام استعلام (UPDATE) واستعلام (SELECT).
ما الفرق بين الحالتين؟

- **اختيار مجموعة من السّجلات بشرطٍ معيّن.**

لاختيار مجموعة من السّجلات بشرط معين نستخدم المعامل (WHERE). هناك مجموعة من المعاملات تستخدم مع المعامل في الشرط، هي:

العملية (Operator)	وصفها Description
=	يساوي Equal
< >	لا يساوي Not Equal
<	أصغر من less than
>	أكبر من Greater than
<=	أصغر من أو يساوي less than or equal
>=	أكبر من أو يساوي Greater than or equal

مثال لعرض سجل بيانات الزبون الذي دفعاته بتاريخ (2019-05-22).

الحل: نكتب الأمر الآتي:

```
SELECT *
FROM CustomerPayment_tbl
WHERE BillPay_Date = '2019-05-22';
```

نشاط (٩) استخراج بيانات مشروطة.

في مختبر الحاسوب وبمساعدة معلمك، افتح قاعدة البيانات السابقة، ثمّ نفّذ الآتي:
- صمّم استعلاماً بكتابة جملة SQL، بحيث تسترجع بيانات المنتجات من كيان المنتج.

(Product_tbl)، والتي تتراوح أسعار الشراء بين (5 و 17).

يمكن أن يتم جمع أكثر من شرط بالمؤثرات المنطقية (AND ,OR):

AND (و): يجب أن يتحقق الشرطين معا لتكون النتيجة إيجابية.

OR (أو): يكفي أن يتحقق احد الشرطين لتكون النتيجة إيجابية.

مثال استخرج بيانات الزبائن الذين تزيد ارقامهم عن (5) ويسكنون مدينة بيت لحم.

الحل:

```
SELECT *
FROM   Customer_tbl
WHERE Custom_ID > 5 AND Custom_Address  = 'بيت لحم';
```

نشاط (١٠) استخدام OR

في مختر الحاسوب وبمساعدة معلمك، قم بالآتي:
افتح قاعدة البيانات السابقة، وأنشئ استعلاماً يُظهر جميع الزبائن الذين يسكنون رام الله او القدس.

- **إظهار الصفات بترتيب معين.**

يمكن عرض السجلات بترتيب اما تصاعدي او تنازلي، كما في المثال الآتي

مثال اعرض بيانات المنتجات مرتبة حسب سعر الشراء بشكل تنازلي:

الحل: باستخدام جملة ORDER BY

```
SELECT *
FROM Product_tbl
ORDER BY Pro_PurchasePrice desc;
```

هناك معامل (LIKE) للبحث داخل الصفات النصية، ويستخدم في البحث عندما لا يمكن معرفة النصّ بشكلٍ كاملٍ؛ وإنما يُعرف جزء منه، فنستعيض عن الجزء الباقي كما في المثال الآتي:

مثال اكتب استعلام لطباعة أسماء الزبائن الذين تبدا أسماؤهم بحرف (ح):

الحل: نستخدم الصيغة الآتية:

```
SELECT *
FROM Customer_tbl
WHERE Custom_Name LIKE 'ح*';
```

رموز الاستعاضة المستخدمة مع المعامل (LIKE):

الرمز	ما يستعيض عنه
؟	للاستعاضة عن حرف واحد فقط.
*	للتعويض مكان أكثر من حرف.

- **دوال التلخيص والتّجميع**

تستخدم هذه الدوال لتلخيص البيانات بأكثر من طريقة، كعرض مجموع قيم حقل ما، او عدد سجلات كيان في قاعدة البيانات.

مثال اكتب جملة (SQL) لعرض عدد سجلات كيان (BillSale_tbl).

الحل: لعرض عدد السجلات نستخدم الدالة (COUNT) بالصيغة الآتية:

```
SELECT COUNT(Bill_ID) AS ‘‘RECORDS_NUMBER’’
FROM BillSale_tbl ;
```

ومجموعة الدوال المستخدمة في التلخيص هي:

اسم الدالة	معنى الدالة
SUM	المجموع
MAX	اكبر قيمة
MIN	اقل قيمة
AVG	المعدل

- **استعلامات من أكثر من كيان والربط بين الكيانات**

كما لاحظت في جميع جمل الاستعلامات السابقة أنّها كانت من كيان واحد، ولكنّ، في قواعد البيانات الحقيقية يكون الاستعلام من أكثر من كيان هو الأساس، فكيف يتمّ هذا الاستعلام؟

الصيغة العامة للاستعلام من اكثر من كيان هي:

```
SELECT column1,column2,….
FROM TABLE1,TABLE2,…
WHERE CONDITIONS;
```

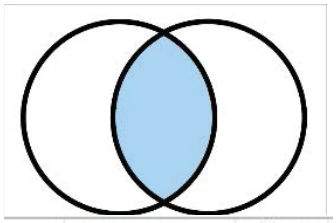
مثال اكتب استعلام لعرض أسماء الزبائن، وأرقام هواتفهم وفواتيرهم الخاصة بهم.

الحل: من الملاحظ أنّ المطلوب عرضه بيانات صفات تنتمي لكيّانين مختلفين، فتكون الصيغة بالشكل الآتي:

```
SELECT Customer_tbl.Custom_ID, Customer_tbl.Custom_phone, BillSale_tbl. Bill_ID
FROM Customer_tbl, BillSale_tbl
WHERE Customer_tbl.Custom_ID=BillSale_tbl.Bill_ID;
```

من الملاحظ في المثال السابق أنّه عندما يتمّ الاستعلام عن صفات تنتمي لأكثر من كيان، يفضّل في كتابة أسماء الصفات أن يسبقها اسم الكيان متبوعة بنقطة (.). برأيك لماذا ؟

توجد عدّة أنواع من الربط بين كيانَين (أو أكثر)، وهنا يكون الكيان الأول ناحية اليسار، والكيان الثاني ناحية اليمين.



١ استخدام المعامل (INNER JOIN).

تحديد السجلات من الكيان الاول والكيان الثاني بحيث تحقّق شرط الرّبط.

مثال أعد كتابة المثال السابق باستخدام (INNER JOIN)، مع إظهار الناتج.

الحل:

```
SELECT dbo.Customer_tbl.Custom_Name, dbo.CustomerPayment_tbl.Bill_ID
FROM dbo.Customer_tbl INNER JOIN CustomerPayment_tbl
ON
dbo.Customer_tbl.Custom_ID = dbo.CustomerPayment_tbl.Custom_ID;
```

	Custom_Name	Bill_ID
1	ابراهيم محمود	100
2	فادي فرح	200
3	جهاد نايف	300
4	سامي محمد	400
5	مهند خليل	500
6	مهند خليل	600

٢ استخدام المعامل (CROSS JOIN):

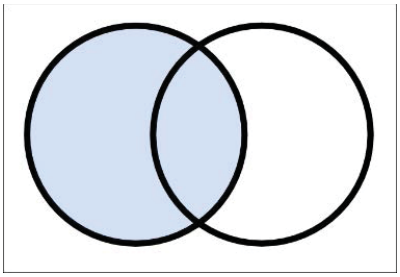
يكون الناتج كيان يتكوّن من: (عدد سجلّات الكيان الأول) مضروبًا في (عدد سجلّات الكيان الثاني).

مثال أعد كتابة المثال السابق باستخدام (CROSS JOIN).

الحل:

```
SELECT Customer_tbl.Custom_Name, CustomerPayment_tbl.Bill_ID
FROM Customer_tbl CROSS JOIN CustomerPayment_tbl
```

٣ استخدام المعامل (LEFT OUTER JOIN)



تحديد كافة السجلات من كيان الاول وبعض السجلات من الكيان الثاني والتي تحقق شرط الربط.

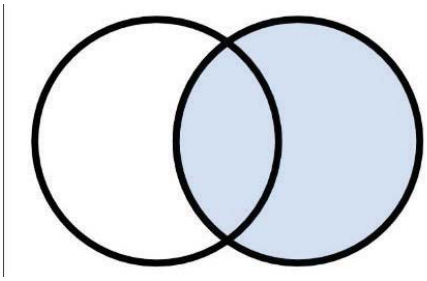
مثال أعد كتابة المثال السابق باستخدام (LEFT OUTER JOIN) مع إظهار الناتج.

	Custom_Name	Bill_ID
1	سامي محمد	400
2	ابراهيم محمود	100
3	مهند خليل	500
4	مهند خليل	600
5	فادي فرح	200
6	أمجد حسني	NULL
7	جهاد نايف	300
8	تهاني عبود	NULL
9	فاطمة حسام	NULL
10	ولاء عمر	NULL

الحل:

```
SELECT Customer_tbl.Custom_Name, CustomerPayment_tbl.Bill_ID
FROM Customer_tbl RIGHT OUTER JOIN CustomerPayment_tbl
 ON
Customer_tbl.Custom_ID = CustomerPayment_tbL.Custom_ID;
```

٤ استخدام المعامل (RIGHT OUTER JOIN)



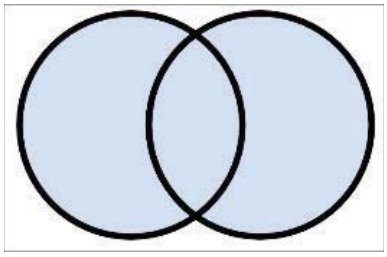
تحديد بعض السّجلات التي تحقّق شرط الرّبط من الكيان الاول، وكافّة السّجلات من الكيان الثاني كما في المثال التالي:

مثال أعد كتابة المثال السابق باستخدام (RIGHT OUTER JOIN) مع إظهار النتائج.

	Custom_Name	Bill_ID
1	ابراهيم محمود	100
2	فادي فرح	200
3	جهاد نايف	300
4	سامي محمد	400
5	مهند خليل	500
6	مهند خليل	600

الحل:

```
SELECT Customer_tbl.Custom_Name, CustomerPayment_tbl.Bill_ID
FROM Customer_tbl RIGHT OUTER JOIN CustomerPayment_tbl
 ON
Customer_tbl.Custom_ID = CustomerPayment_tbL.Custom_ID;
```



٥ استخدام المعامل (FULL OUTER JOIN)

تحديد كافة السجلات من الكيانَين الاول والثاني بغضّ النّظر عن تحقيقها للشّرط.

مثال أعد كتابة المثال السابق باستخدام (FULL OUTER JOIN).

الحل:

```
SELECT Customer_tbl.Custom_Name, CustomerPayment_tbl.Bill_ID
FROM Customer_tbl FULL OUTER JOIN CustomerPayment_tbl
 ON
Customer_tbl.Custom_ID = CustomerPayment_tbL.Custom_ID;
```

الإجراء المخزن (Stored Procedure)

جملة استعلام يتمّ انشاؤها وتخزينها داخل قواعد البيانات، بحيث يمكن استخدامه عوضاً عن العديد من الاستعلامات، والشكل المجاور يبين الشكل العام لهذا الاجراء.

```
CREATE PROCEDURE
AnyName -- اسم الاجراء
        متغيرات الاجراء
AS
BEGIN
الجمل البرمجية
END
```

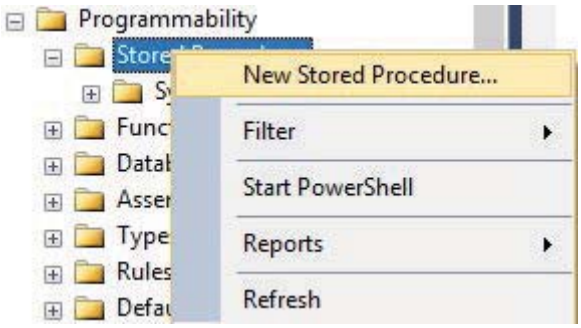
يُعرّف المتغيّر داخل الإجراء بكتابة اسم المتغير مسبوقا بـــ @ يليه نوع المتغير

```
DECLARE @ Userid INT
```

مثال قم بإنشاء اجراء مخزن لتخزين بيانات في كيان مخزن (Store_tbl).

الحل:

١ من المستكشف (Object Browser)، يتمّ اختيار قاعدة البيانات بقالة(Market)، ثمّ اختيار مجلّد (Programmability) ليظهر مجلّد الإجراء المخزّن (Stored Procedure).

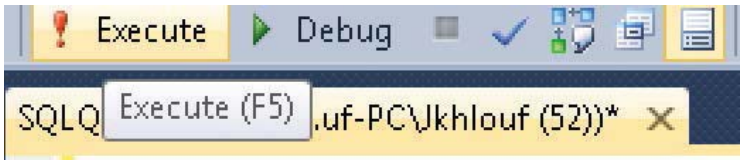


٢ الضغط بزر الماوس الأيمن على المجلّد كما في الشكل المجاور.

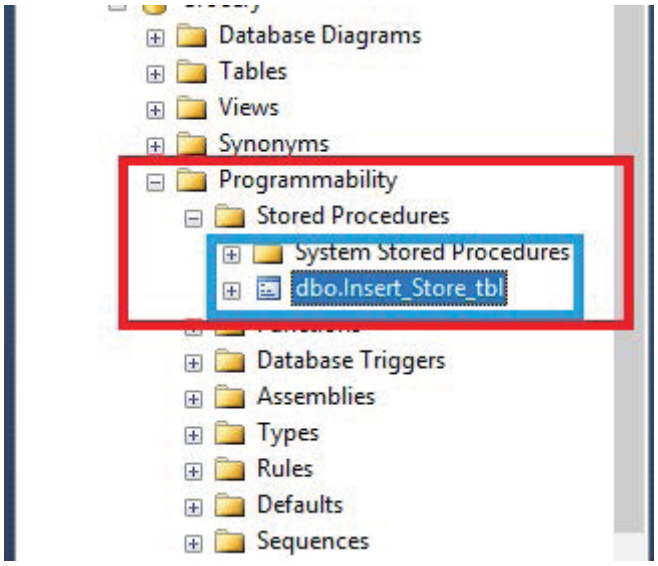
٣ البدء بكتابة الكود البرمجي التالي:

```
CREATE PROCEDURE [dbo].[Insert_Store_tbl]
    @Sto_ID INT, @Sto_Location NCHAR(15),
    @Sto_Rent BIT
AS
BEGIN
    Insert Into Store_tbl(Sto_ID,Sto_Location,[Sto_Rent?])
    Values
    (@Sto_ID,@Sto_Location,@Sto_Rent)
END
```

ثم قم بتنفيذ الجمل البرمجيّة أعلاه كما في الشكل أدناه.



وعند التنفيذ ينشئ اجراء مخزن ويظهر كمّا في الشكل الآتي:



ويمكن استخدام الاستعلام المخزن كمّا يلي:

```
Insert_Store_tbl '111','جنوب الخليل','True'
```

100 %

Messages

(1 row(s) affected)

نشاط (١١) إنشاء الإجراء المخزّن لمشروع البقالة.

قم بإنشاء إجراء مخزن لجميع الكيانات كمّا سبق ذكره في المثال

العرض (Views)

عبارة عن بناء كيان وهميّ لا يحوي على بيانات حقيقيّة؛ وإنّما فقط للعرض، وبذلك تتحقّق عدّة مميّزات لاستخدام (Views)، نذكر منها:

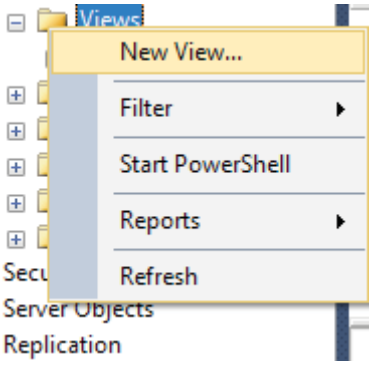
- صفات (View)؛ عبارة عن مزيج من صفات كيان أو عدّة كيانات.
- سهولة الاستعلام عن البيانات المعقدة.

- تحسين أداء قاعدة البيانات.
- تحقيق أعلى درجات الحماية لقواعد البيانات عن طريق منح الصّلاحيات لعدد معيّن من المستخدمين.

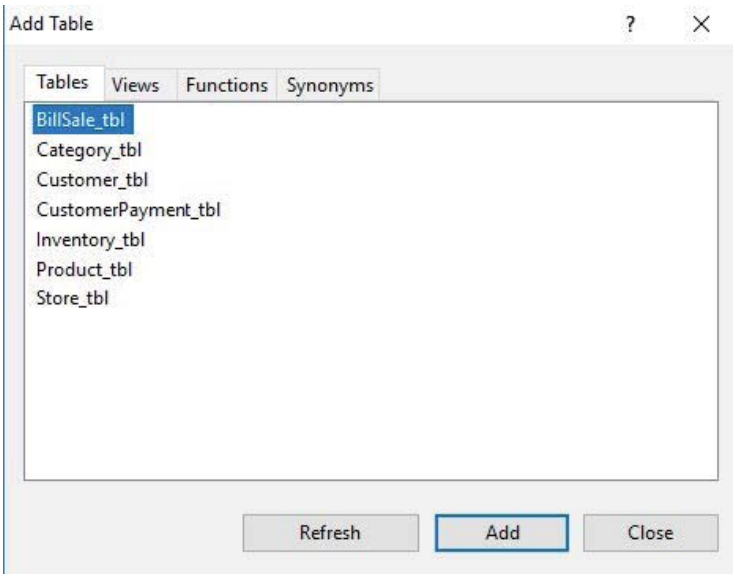
• إنشاء (View)

لإنشاء (View) هناك طريقتان:

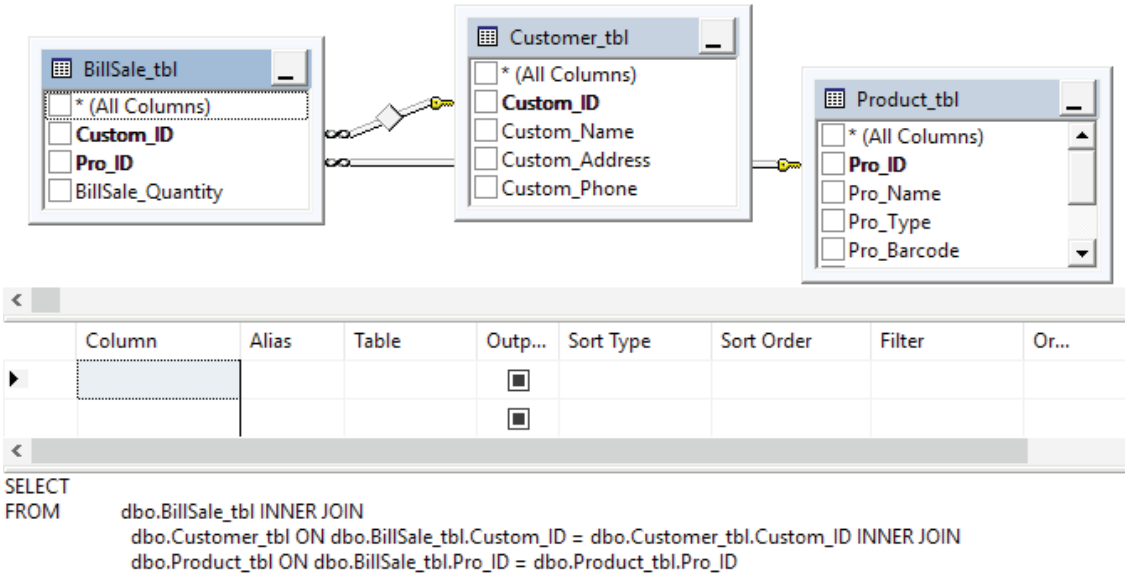
الطريقة الأولى: من خلال متصفّح الكائنات (Object Browsers)؛ نقوم بتوسيع مجلّد قاعدة البيانات، وبالضغط بزر الماوس الأيمن على مجلد (View) ونختار (New View) كما في الشكل الآتي:



فيظهر الشّكل الآتي والذي يعرض جميع الكيانات التي تمّ إنشاؤها.



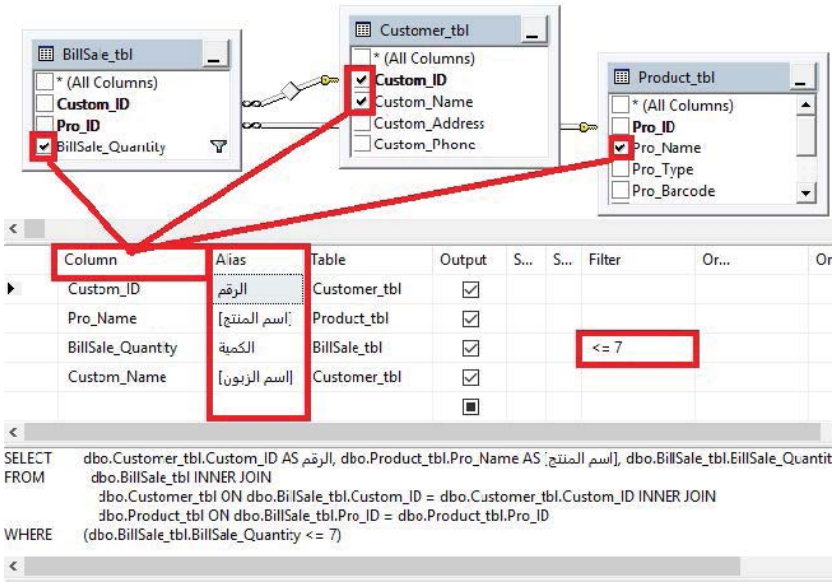
نقوم باختيار الكيانات من خلال الضغط على الكيان فيظهر الشكل التالي:



من خلال الشكل السابق، يمكن اختيار الصّفات التي سوف تظهر في (View) من خلال وضع إشارة (☑) في المربع الجانبي للصفة المطلوبة، ثمّ الضّغط على زر (Save) في شريط الأدوات لحفظ الإستعلام.

يمكن وضع شرطٍ لعرض السّجلات داخل (View) من خلال وضع معيار أو أكثر من معيار داخل مربع (Filter) كما هو موضّح في الشّكل المجاور.

يمكن استعراض (View) من خلال الضغط على زر الماوس الأيمن واختيار ((Edit top 200 rows فتظهر نتيجة الاستعلام.



الطريقة الثانية: من خلال كتابة جملة SQL والصيغة العامة كالتالي:

وضع اسم ل (View) جملة select	**CREATE VIEW** view_name **AS** **SELECT** column1, column2,... **FROM** table_name **WHERE** condition;

نشاط (١٢) إنشاء (View)

أنشيء (View) السّابقة عن طريق كتابة جملة (SQL)، واعرض من الصفات ما تراه مناسباً، مع الأخذ بعين الاعتبار أنّ الشرط هو: (BillSale_Quantity >=15).

1- وضّح المقصود بالمصطلحات الآتية: النموذج، التقارير، العرض (View)، الاستعلامات.

2- قارن بين استخدام المعامل (WHERE) مع استعلام الحذف (DELETE) من عدمه داخل لغة (SQL) من حيث: عدد السجلات المحذوفة.

3- ضع دائرة حول رمز الإجابة الصحيحة:

- استخدام النجمة (*) مع أمر (SELECT) في لغة (SQL) تعني:

أ- اختيار عدة صفوف من الكيان

ب- اختيار عدّة صفات من الكيان

ج- اختيار جميع السجلات من الكيان

د- اختيار جميع الصفات من الكيان

- اللغة التي تتحكم بالبيانات من حيث منح الصلاحيات للمستخدمين هي:

أ- DML ب- DSL ج- DDL د- UML

4- لديك الكيان التالي، ادرسه جيّداً، ثم أجب عن الأسئلة التالية:

STUDENT_ID	STUDENT_NAME	STUDENT_COST
1	محمد	$20
2	سليمان	$30
3	جميل	$44
4	حسين	$55

أ- اكتب جملة SQL تقوم بإظهار مجموع التكلفة (STUDENT_COST).

ب- اكتب جملة SQL تقوم بطباعة أكثر تكلفة دفعت.

ج- اكتب جملة SQL تقوم بطباعة الكيان السابق مرتّباً تصاعديّا حسب صفة اسم الطالب(STUDENT_NAME).

د- أُضِيف كيان الصف (Class) للنظام تحوي الصفات:(Class_No، Class_Name)، ويحوي 5 سجلات داخله، أجب عن الأسئلة الآتية:

1- أنشئ علاقة مناسبة بين الكيانين.

2- أنشئ مفتاحاً أجنبيّا مناسباً.

3- اكتب جملة (SQL) لإظهار اسم الطالب واسم الصف بطريقة (CROSS JOIN).

4- ما عدد السجلات الناتجة؟

5- أنشئ عرض (View) مناسباً، بحيث أنّ: ((STUDENT_COST)>=40).

أسئلة الوحدة

س١ ضع دائرة حول رمز الاجابة الصحيحة.

1- ما هي قاعدة البيانات؟

أ) مجموعة من البرامج تعمل على حفظ البيانات.

ب) مجموعة من أوامر SQL لمعالجة البيانات.

ج) شاشة المستخدم.

د) مخزن لحفظ البيانات.

2- ما الرابطة التي ترتبط فيها الكيانات في قاعدة البيانات العلائقية؟

أ) منطقية.

ب) فيزيائية.

ج) هيكلية.

د) أ، ب معاً.

3- ما الصفة التي يمكن إسناد قيم عديدة إليها؟

أ) المفتاح الاولي المكون من عدة صفات (حقول).

ب) المفتاح الأجنبي المكون من عدة صفات (حقول).

ج) الصفة التي يمكن تجزئتها لعدة صفات.

د) غير ذلك.

4- ما النّموذج في قواعد البيانات والذي يتمّ تخزين البيانات فيه على شكل سلاسل مترابطة؟

أ) النموذج الهرمي.

ب) النموذج الشبكي.

ج) النموذج العلائقي.

د) لا شيء مما ذكر.

ادرس الكيان (PERSON) التالي واجب عن الأسئلة التي تليه:

NUMBER	FIRSTNAME	LASTNAME	CITY	AGE
1	محمد	الرائد	الخليل	44
2	محمود	الرائد	رام الله	55
3	عبدالله	السلمان	الخليل	25
4	رامي	العايد	رام الله	36
5	سلمى	الفارس	الخليل	65

5- جملة SQL التي تعرض جميع الأشخاص الذين أعمارهم فوق 40

أ) SELECT * FROM PERSON;

ب) SELCT * FROM PERSON WHERE AGE GREATER THAN 40;

ج) SELECT * FROM PERSON WHERE AGE>40;

د) SELECT FROM PERSON WHERE AGE>40

6- جملة SQL والتي تحذف الأشخاص الذين أعمارهم اقل من 50

أ) DELETE PERSON;

ب) DELETE FROM PERSON WHERE AGE<50;

ج) DELETE FROM PERSON WHERE AGE=50;

د) DELETE FROM PERSON;

7- الجملة الاتية تعمل على:

INSERT INTO PERSON VALUES(6,'RAWEYA','AHMAD','Salfit' 22);

أ) إضافة شخص جديد الى كيان (PERSON).

ب) حذف شخص من كيان (PERSON).

ج) لا تتم عملية الإضافة لسبب ما.

د) اختيار الطلبة جميعا.

س2 ما المقصود بالمصطلحات الآتية: العرض، التقرير، قاعدة البيانات، أداة الربط (Join)، استعلام (DDL).

س3 عدّد أنواع الاستعلامات مع التوضيح لكلّ منها.

س4 اذكر حالات الربط (JOIN) مع الرسم.

س5 لديك الكيان التالي، من قاعدة بيانات قم بتحويله على درجة التسوية الثالثة:

رقم المشروع	اسم الشروع	رقم الموظف	اسم الموظف	الفئة الوظيفية	التكلفة بالساعة	عدد ساعات العمل
1	صيانة الطريق	11	عبد الله أحمد	مهندس مدني	200 دينار	30
		12	خالد ناصر	مراقب	140 دينار	70
		13	محمد وليد	مساعد مراقب	100 دينار	70
2	بناء طريق	11	عبد الله أحمد	مهندس مدني	100 دينار	45
		14	فهد الفهد	مراقب	133 دينار	50

س6 ما المفتاح الاساسي للكيان الآتي:

A	B	C	D
a1	**b1**	**c1**	**d1**
a1	**b2**	**c2**	**d1**
a2	**b2**	**c3**	**d2**

مشروع الوحدة:

ادرس الحالة الآتية ثم اجب عن الأسئلة التي تليها:

في مدينة رام الله يراد عمل قاعدة بيانات لشركة النقل العام حيث لدى الشركة العديد من الباصات ذات المواصفات المختلفة من حيث سنة الصنع، والسعة، والحالة الفنية للباص، وتقوم الشركة بتسيير العديد من الرحلات على اكثر من خط، ولكلّ خط مخطة انطلاق ومحطة وصول وعدد من المحطات (المواقف)، ولدى الشركة أنواع مختلفة من الموظفين؛ فهناك السائق، ومراقب الخط، والإداري، يسير الباص الواحد ورديتان الأوّلى صباحية بسائق مختلف عن الوردية المسائية، وهناك سائق احتياط يحل مكان السائق الأساسي في حال مرضه أو غيابه لظرف ما.

قم بإنشاء المشروع عمليّا على إحدى برامج قواعد البيانات، مراعياً ما يأتي:

أ- حدّد الكيانات والصفات المناسبة لكلّ منها.

ب- حدّد العلاقات بين الكيانات.

ج- قم برسم مخطط (ERD) للحالة.

د- أنشئ عرض (View) مناسباً.

الوحدة الثانية

البرمجة المتقدّمة

(Advanced Programming)

أتأمل ثم أُناقش: البرمجة طريقك للحياة

مقدمة الوحدة

يُبنى علم البرمجة على أسس وقواعد لا بدّ من تعلّمها لكلّ مبرمج طموح، إلّا أنّ الطّريق لتحقيق النّجاح والتّميّز مرهون بما يملكه المبرمج من إبداع خاص نظراً لإمكانيّة تنفيذ الفكرة بأكثر من طريقة، وتقديمها بمستويات مختلفة من البساطة والسرعة وقابليّة التّحديث وتصحيح الأخطاء والكفاءة؛ فالنّاظر إلى مجموعة البرامج والتطبيقات يلتمس الاختلاف الأدائي بينها من حيث ميّزاتها ومرونتها وكمّ الوظائف التي تؤدّيها وشاشاتها على الرّغم من التّشابه الوظيفيّ بينها، وهذا راجع إلى حنكة المبرمج وعقليّته الخلّاقة وقدرته البرمجيّة المتجّددة دوماً.

في هذه الوحدة، سنتناول المفاهيم المتعلّقة بتراكيب البيانات (**Data Structure**) وبعض تطبيقاتها، وطرق التّعامل معها برمجيّا، وصولاً لنمط البرمجة الهدفيّة «Object Oriented Programming» ومميّزاتها من خلال أمثلة وتطبيقات برمجيّة عمليّة بلغة «**VB.NET**».

أهداف الوحدة:

يَتوقع منك بعد دراستك هذه الوحدة أن تكون قادراً على تطوير المهارات البرمجية وصقلها من خلال تحقيق الأهداف الآتية:

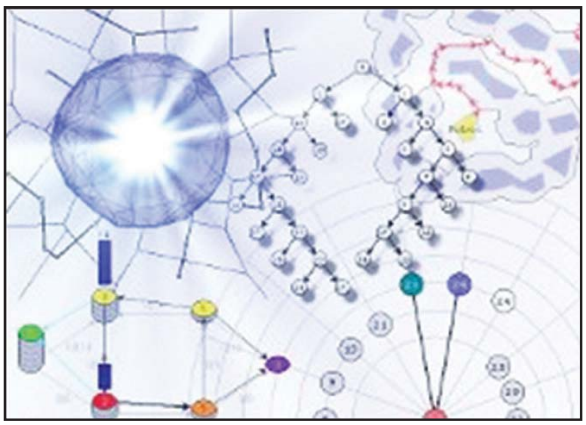
١. حل مشكلات حياتيّة باستخدام مفاهيم تراكيب البيانات الخطيّة.

٢. حلّ مشكلات حياتيّة برمجيّا ومحاكاتها بمفاهيم البرمجة الهدفيّة.

الدرس (١)

تركيب البيانات

(Data Structure)



ينتظم الأفراد في جماعات (جمعيات، أندية، نقابات...) لأداء مَهمّة معيّنة، وكذلك في الحاسوب، تخزّن البيانات في تراكيب وهياكل محدّدة لتجسيد أو محاكاة ظاهرة حياتية معينة (طابور، ملفات، خرائط، أشجار..)، فعلى سبيل المثال؛ يمكن وضع بيانات طلبة لمدرسةٍ ما في ملف واحد بمعايير معيّنة لإجراء عمليات محدّدة عليها؛ كتعديل لمعلومة أو حذفها أو البحث عن معلومة تعود لطالبٍ ما في تلك المدرسة.

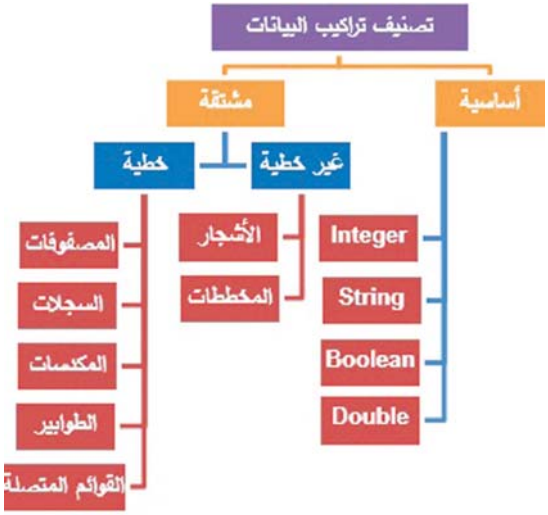
تعلّمتَ سابقاً طرق تخزين البيانات بتراكيب ذات أحجام مختلفة، مثل: المتغيّرات، والمصفوفات الأحادية ومتعدّدة الأبعاد بأنواع مختلفة، مثل: Double، Integer، String. كما تعلّمت أنّ المتغيّر والمصفوفة يتشابهان في تخزين البيانات ذات النوع الواحد فقط، ويختلفان في حجم البيانات التي يمكن لكلّ منهما تخزينها.

نشاط (١) مقارنة بين المتغيّر والمصفوفة.

- أعلن عن متغيّرين (Num1، Num2) من نوع عدد صحيح، ومتغيّرين (Str1، Str2) من نوع نصّي.
- أعلن عن مصفوفة أحادية باسم Std عدد عناصرها 2 من نوع عدد صحيح.
- أعلن عن مصفوفة أحادية اسمها Strr عدد عناصرها 2 من نوع نصي.
- أسند قيماً مناسبة للمتغيّرات والمصفوفتين أعلاه.
- مثّل المتغيّرات والمصفوفتين بالرّسم.
- عدّل قيم المتغيّرات والمصفوفتين بإضافة العدد 3 للأنواع الصحيحة منها، وحرف 'W' للأنواع النصّية منها.
- اطبع قيم المتغيّرات والمصفوفات وعدد عناصرها على أدوات مناسبة.

بشكل عام، تُعبِّر تراكيب البيانات عن الطّريقة التي يتمّ من خلالها تخزين وتنظيم البيانات في ذاكرة الحاسوب منطقيّا وبرمجيّا، بحيث يمكن استخدامها بكفاءة.

تصنّف تراكيب البيانات إلى نوعين رئيسين:



أ. تراكيب البيانات الأساسية (Primitive):

ويقصد بها أنواع البيانات الأوّلية والمعرّفة مسبقاً داخل لغة البرمجة، مثل: Integer، String، Boolean، Double....

ب. تراكيب البيانات المشتقة (Non-Primitive):

ويقصد بها أنواع البيانات غير المعرّفة مسبقاً داخل لغة البرمجة، ويتمّ إنشاؤها من المبرمج اعتماداً على تراكيب البيانات الأساسية، لاحظ الشّكل المجاور. وسنعرض هنا بعضاً من تراكيب البيانات الخطية.

أولاً: المكدّسة (Stack)

في علوم الحاسوب، تمثّل المكدّسة أحد أنواع تراكيب البيانات الخطيّة، والتي تتشابه في التّمثيل والبرمجة مع المصفوفات - التي درستها سابقاً - بتعاملها مع البيانات كمجموعة من العناصر ذات النوع الواحد فقط؛ إلّا أنّ عمليّات الحذف والإضافة بالمكدسة هي ما يميّزها عن المصفوفة؛ حيث يشترط تنفيذ العمليّات المختلفة من طرف واحد فقط، لذلك يطلق عليها الوصف (LIFO) اختصاراً للتّعبير (Last In First Out)، أيّ أنّ آخر العناصر المضافة إلى المكدّسة ستكون أوّل العناصر الخارجة منها، لاحظ الشّكل أعلاه.

نشاط (٢) أمثلة على المكدّسة.

أيّ المواقف الآتية يمثّل فكرة المكدّسة؟

١- ترتيب أوراق اللّعب فوق بعضها البعض.

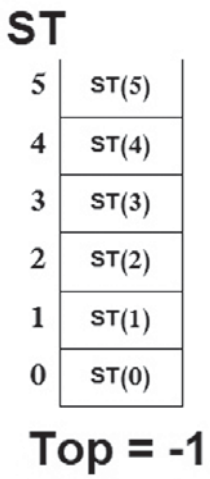
٢- ترتيب الأطباق في خزانة المطبخ.

٣- طابور العملاء في البنك.

٤- اصطفاف الطلبة بطابور الصباح.

٥- اصطفاف السيارات أمام إشارة المرور.

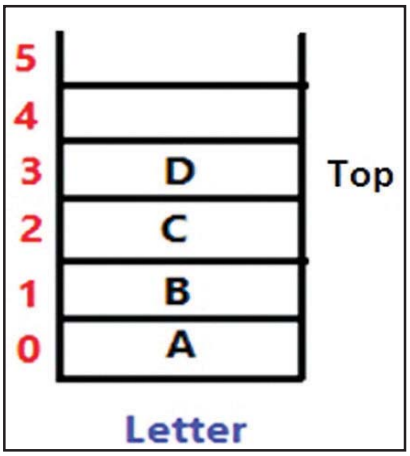
تُمثَّل المكدّسة كمصفوفة أحاديّة البعد، فمثلاً؛ إذا كان لديك مكدّسة فارغة اسمها (ST)، وعدد مواقعها ستة، فإنّه يتمّ تمثيلها كما في الشّكل المجاور.



يُشار إلى موقع آخر عنصر في المكدّسة (الفهرس - index) بمتغيّر يُعلن عنه مسبقاً (وليكن مثلاً Top)، تُسند له القيمة (-1) ابتدائيّاً بحيث تتغيّر قيمته باستمرار أثناء عمليّتَي الحذف والإضافة للعناصر (البيانات) في المكدّسة.

مثال

تمعّن في الشّكل المجاور، ثمّ أجب عن الأسئلة الآتية:



أ. ما اسم المكدّسة؟ وما سعتها؟

ب. ماذا تمثّل الحروف (A،B،C،D) في الشّكل؟

جـ. ما قيمة المتغيّر Top؟

الحل:

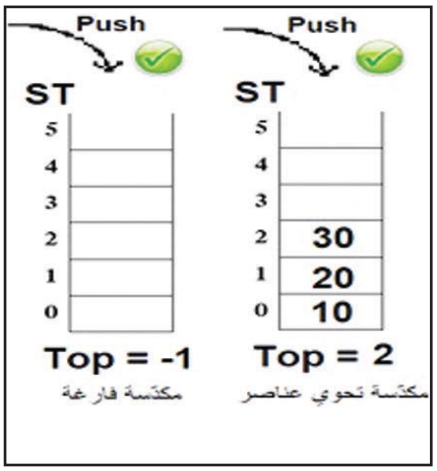
أ. اسم المكدّسة Letter، وسعتها 6.

ب. تمثّل الحروف عناصر (بيانات) مخزّنة داخل المكدّسة من نوع نصّي.

ج. Top = 3.

عمليّات على المكدّسة.

عمليّة الإضافة.



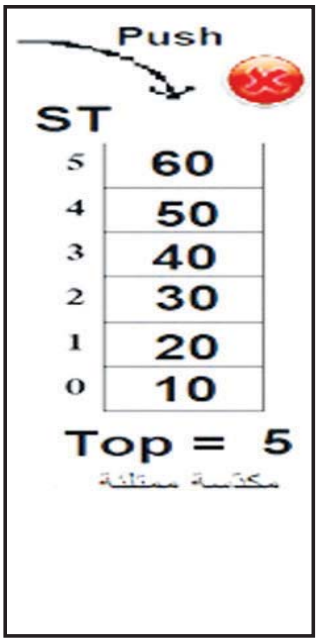
يُطلق على عمليّة إضافة عنصر إلى مكدّسة كما هو متعارف عليه لدى المبرمجين اللّفظ (Push)، ويمكن تلخيص حالات الإضافة إلى مكدّسة كما يأتي:

الحالة الأولى:

أن تكون المكدّسة فارغة (Empty) أو تحوي بعض المواقع الفارغة من العناصر؛ بمعنى أنّ عدد العناصر في المكدّسة أقلّ من عدد المواقع في المكدّسة، وهنا تتمّ عمليّة الإضافة بنجاح دون مشاكل متوقّعة.

الحالة الثانية:

المكدّسة ممتلئة (Full)؛ بمعنى أنّ عدد العناصر في المكدّسة يكافئ سعتها، وهنا لا يمكن إتمام عمليّة الإضافة.



ويمكن ترجمة ما سبق من خلال الخوارزميّة والمخطّط الانسيابي كما يلي:

الخوارزميّـــــة

1. **بداية.**
2. أدخل قيمة العنصر المراد إضافته **(وليكن num).**
3. هل **Top = N–1؟**

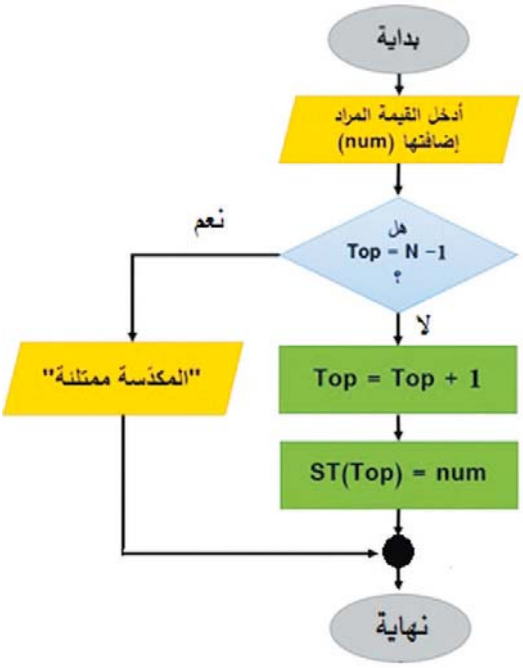
أ. **نعم:** – اطبع **"مكدّسة ممتلئة"**

– اذهب الى الخطوة **6**

ب. **لا:** – اذهب إلى الخطوة **4**

4. **Top = Top + 1**
5. **ST(Top) = num**
6. **نهاية.**

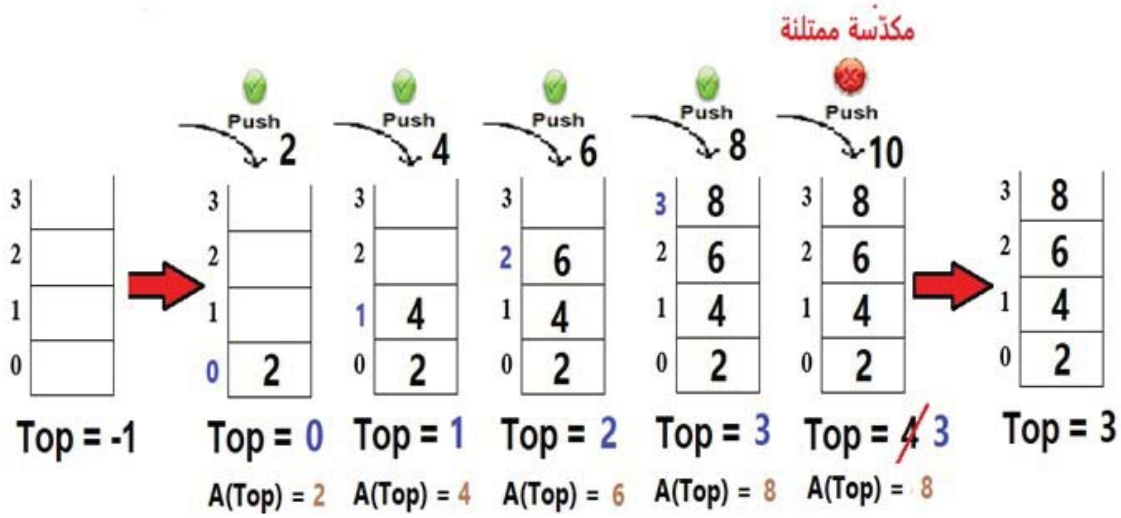
المخطّط الانسيابي



مثال

لديك مكدّسة اسمها (A) وحجمها 4، مستعيناً بخوارزميّة الإضافة للمكدّسة أعلاه، مثّل عمليّة الإضافة المتتالية للعناصر: 2، 4، 6، 8، 10بالرّسم، ومحدّداً قيمة A(Top)، Top في كل مرّة.

الحل:



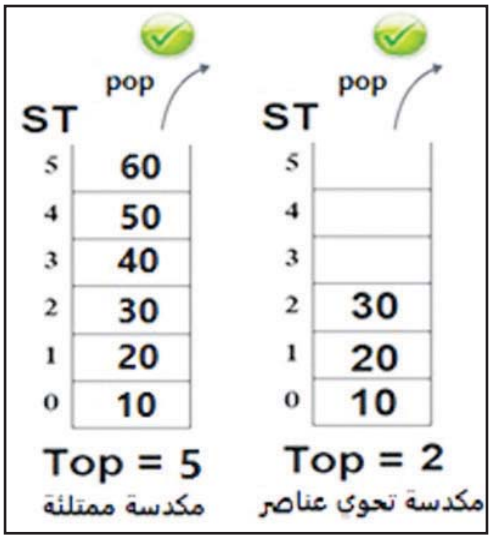
نشاط (٣) تتبّع خوارزمية الإضافة إلى مكدّسة.

لديك المكدّسة الفارغة Employee (5)، من خلال الرّسم تتبّع قيمة Top، Employee(Top) لكلّ جملة من الجمل الآتية:

١. أضف للمكدّسة العناصر 5، 12، 23، 78، 81 بشكل متتالٍ.

٢. أضف العنصر 88، هل يمكن ذلك؟ لماذا؟

عمليّة الحذف (الإزالة).



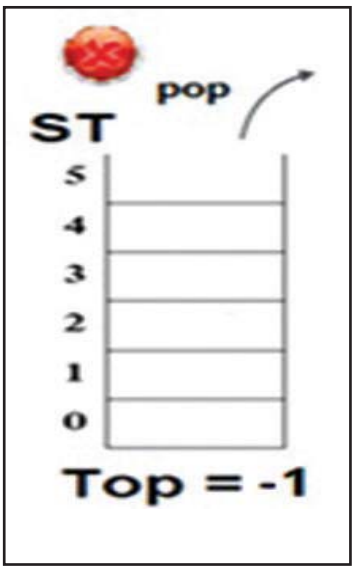
يُطلق على عمليّة حذف عنصر من مكدّسة كما هو متعارف عليه لدى المبرمجين اللّفظ (POP)، ويمكن تلخيص حالات الحذف من مكدّسة كما يأتي:

الحالة الأولى:

أن تكون المكدّسة ممتلئة (Full) أو تحوي بعض العناصر، وهنا تتمّ عمليّة الحذف بنجاح دون مشاكل متوقّعة. انظر إلى الشّكل المجاور.

الحالة الثانية:

المكدّسة فارغة (Empty)، ولا يمكن إتمام عمليّة الحذف كما في الشّكل الآتي.



ويمكن ترجمة ما سبق بالخوارزميّة والمخطّط الانسيابي أدناه:

الخوارزميّـــــة

1. **بداية.**
2. **هل Top = −1؟**

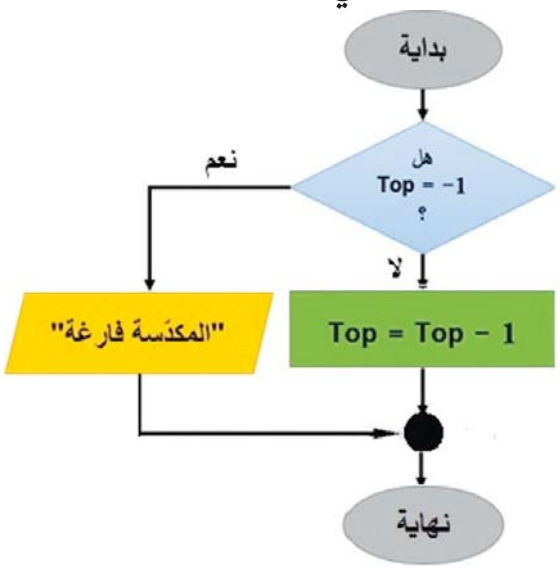
أ. **نعم:** اطبع "**مكدّسة فارغة**"

اذهب إلى الخطوة **4**

ب. **لا:** اذهب إلى الخطوة **3**

3. **Top = Top − 1**
4. **نهاية.**

المخطّط الانسيابي



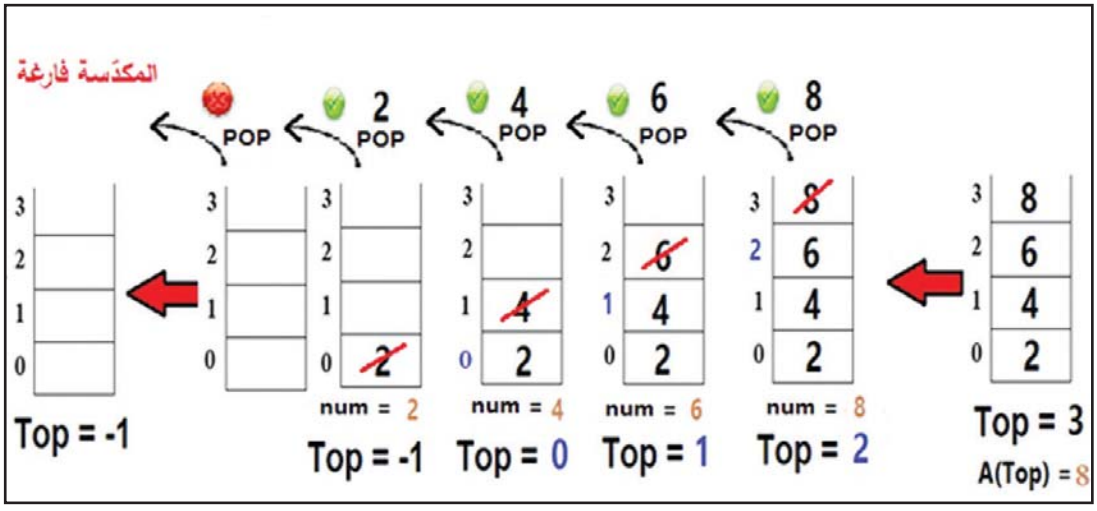
فكّر:

قم بإجراء التّعديلات اللازمة أعلاه لطباعة العنصر المراد حذفه قبل عمليّة الحذف.

مثال

مستعيناً بخوارزميّة الحذف من المكدّسة أعلاه، مثّل بالرسم عمليّة الحذف لـ5 عناصر من المكدسة (A) في المثال السابق، ومحدّداً قيمة العنصر المحذوف(num) وTop في كلّ مرّة.

الحل:



ويمكن تصميم وبرمجة عمليّتي الإضافة والحذف للمكدّسة بلغة VB.NET كما في المثال الآتي:

مثال صمم الواجهة أدناه واضبط خصائصها، ثم اكتب الجمل البرمجية المناسبة لإضافة وحذف مجموعة بيانات من المكدسة بشكلٍ متتالٍ.

الحل:

١. تصميم الواجهة وضبط خصائص الأدوات والأزرار الظاهرة كما يأتي:

أدوات واجهة التصميم	الخصائص	الخصائص الفرعية	القيم
Form1		Icon	(صورة مناسبة من نوع Ico)
		Language	Arabic (Palestinian Authorirty)
		RightToLeft	No
	Size	Width	800 px
		Height	470 px
	Font	Name	Traditional Arabic
		Size	8
		Bold	True
GroupBox1	Font	Name	Traditional Arabic
		Size	20
		Bold	True
	Size	Width	350
		Height	200
		RightToLeft	Yes
		Text	الاضافة الى المكدّسة

العمليّـــات على المكدّســات	Text		Label1
328	Width	Size	
61	Height		
Traditional Arabic	Name	Font	
30	Size		
True	Bold		
" "	Text		TextBox1
Left	TextAlign		
525	Width	Size	
39	Height		
Info	Backcolor		

نشاط (٤) ضبط خصائص الأدوات.

قـم بضبـط خصائـص الأدوات للعناصـر المتبقيـة فـي واجهـة التّصميـم السـابقة محـدّدا القيـم المناسـبة للخصائـص المسـتخدمة لـكلّ منهـا.

٢. فتح نافذة البرمجة للنّموذج (Form1) وكتابة الجمل البرمجيّة الآتية:

```
Private Sub Form1_Load (sender As Object, e As EventArgs) Handles MyBase.Load
    TextBox1.Enabled = False
    TextBox2.Enabled = False
    TextBox3.Enabled = False
    TextBox4.Enabled = False
    TextBox5.Enabled = False
    TextBox6.Enabled = False
    TextBox7.Enabled = False
    TextBox1.Text = ""
    TextBox2.Text = ""
```

```
TextBox3.Text = “”
TextBox4.Text = “”
TextBox5.Text = “”
TextBox6.Text = “”
TextBox7.Text = “”
        End Sub
```

٣. إضافة الجمل البرمجيّة الآتية إلى منطقة التّعريفات العامّة في نافذة البرمجة:

```
#Region “Declare Stack”
  Dim ST(3) As Integer      ' الإعلان عن مكدّسة حجمها 4
  Dim Top As Integer = -1   ' الإعلان عن المؤشر
#End Region
```

٤. فتح نافذة البرمجة لزر الأمر (PUSH) وكتابة الجمل البرمجيّة الآتية:

```
  Private Sub Button1_Click (sender As Object، e As EventArgs) Handles Button1.Click
    GroupBox1.Visible = True
    GroupBox2.Visible = False
    PUSH ()                                   ' استدعاء دالة الإضافة إلى المكدّسة
  End Sub
Public Sub PUSH ()
    Dim num As Integer
  If Top = 3 Then
    MessageBox.Show ("المكدسة ممتلئة!!! لا يمكن الاضافة")   ' في حالة المكدسة ممتلئة
 Else                                         ' في حالة الإضافة ممكنة للمكدسة
    TextBox2.Text = Top
    Top += 1
```

```
    TextBox4.Text = Top
    num = InputBox("أدخل القيمة الصحيحة المراد إضافتها")          ' إدخال قيمة من خلال صندوق الإدخال
    TextBox3.Text = num                                            ' طباعة القيمة المضافة
    ST(Top) = num                                                  ' إضافة قيمة الى المكدسة
    TextBox1.Text = TextBox1.Text & " " & ST(Top)
  End If
End Sub
```

٥. فتح نافذة البرمجة لزر الأمر (POP) وكتابة الجمل البرمجيّة الآتية:

```
Private Sub Button2_Click (sender As Object، e As EventArgs) Handles Button2.Click
    GroupBox1.Visible = False
    GroupBox2.Visible = True
   POP ()                                                          ' استدعاء دالة الحذف
  TextBox1.Text = ""
   For i = 0 To Top
   TextBox1.Text = TextBox1.Text & " " & ST(i)                     ' طباعة محتويات المكدسة بعد عملية الحذف
    Next
End Sub
  Public Sub POP ()
    Dim num As Integer                                             ' المتغير المراد حفظ العنصر المحذوف داخله
 If Top = -1 Then
   MessageBox.Show("المكدسة فارغة!!! لا يمكن الحذف")
  Else
    num = ST(Top)                                                  ' قيمة العنصر المراد حذفه
    TextBox6.Text = num                                            ' طباعة قيمة العنصر المراد حذفه
```

```
        TextBox5.Text = Top     ' طباعة قيمة المؤشر Top قبل عملية الحذف
        Top -= 1                ' تمّ الحذف
        TextBox7.Text = Top     ' طباعة قيمة المؤشر Top بعد عملية الحذف
    End If
End Sub
```

٦. فتح النافذة البرمجية لزر (أمر الخروج) وكتابة الجمل البرمجيّة الآتية:

```
Private Sub Button3_Click (sender As Object، e As EventArgs) Handles Button3.Click
    End
    End Sub
```

قم بتنفيذ البرنامج، ماذا تلاحظ؟

تطبيقات على المكدّسات

أ. فحص مطابقة الأقواس

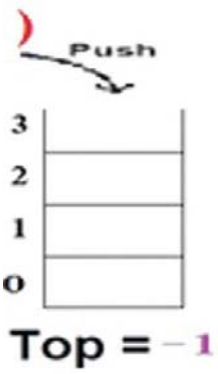
ملحوظة:
) قوس أيمن
(قوس أيسر

لدينا تعبير رياضي، نريد أن نعرف هل عدد الأقواس اليمنى تساوي عدد الأقواس اليسرى للحكم على صحّة التّعبير الرّياضي من حيث مطابقة عدد الأقواس اليمنى مع اليسرى.

نفترض أنّ التّعبير مكتوب باللّغة الإنجليزية من اليسار إلى اليمين، فتكون خوارزميّة العمل كالآتي:

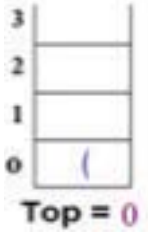
أ. إذا صادفنا قوسا أيسر نضيفه إلى المكدّسة.

ب. إذا صادفنا قوسا أيمن نحذف قوسا أيسر من المكدّسة.



يكون التّعبير خاطئا في إحدى الحالات الآتية:

١. إذا واجهنا قوساً أيمن بالتعبير والمكدّسة فارغة، ففي هذه الحالة يكون الحكم بعدم تطابق الأقواس، بسبب ان عدد الاقواس اليمنى أكبر من عدد أقواس اليسرى، مثال: (4/ (2 + 3).



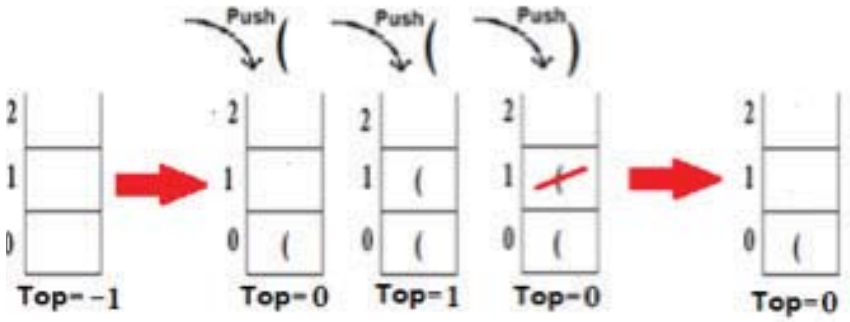
٢. إذا انتهينا من الإدخال والإخراج وكانت المكدّسة ليست فارغة، ففي هذه الحالة يكون الحكم بعدم تطابق الاقواس، والسبب ان عدد الأقواس اليمنى أقلّ من عدد أقواس اليسرى، مثال: ((4/ 2)).

٣. إذا انتهينا من الإدخال والإخراج، وكانت المكدّسة فارغة، فيكون الحكم ان الأقواس متطابقة بمعنى ان عدد الأقواس اليمنى مساوٍ لعدد الاقواس اليسرى مثال: (9+(4/2)-5).

مثال

من خلال الرسم، أَصدِر حُكماً على صحّة تطابق الأقواس في التّعبير ، معلّلا السبب (1-8-4)*8)-9

الحل: الأقواس غير متطابقة؛ بسبب انتهاء الإدخال والإخراج للمكدسّة مع بقاء قوسٍ واحدٍ أيسر في المكدّسة؛ أي أنّ عدد الأقواس اليمنى أقل من عدد الأقواس اليسرى في التّعبير.



سؤال

من خلال الرسم، أَصدِر حُكماً على صحّة تطابق الأقواس في التّعبير أدناه، معلّلا السبب.

أ. ((4 - 66)) ب. ((5 * 33)) ج. ((3 / 9))

ب. تقييم التّعبير (Expression Evaluation).

يقصد به مجموعة العمليّات (Operations/Operators) والمعاملات (Operands) التي تُمثّل قيمة محدّدة، مثال ذلك: (2+3 = 5 أو 6 < 3)، وهي أحد تطبيقات تركيب المكدّسة المهمّة إلى جانب حساب أقصر طريق وإدارة الذاكرة وعمليّات البحث.

تُقسم التّعابير الحسابية حسب موقع المعاملات الحسابيّة في التّعبير إلى ثلاثة أنواعٍ رئيسة:

١. تعبير (infix): التّعبير الأكثر شُهرة والمُتعارف عليه في تمثيل العمليات الحسابية في الرياضيات؛ حيث يقع رمز العمليّة بين المُعامِلَيْن، والصّورة العامة للتّعبير كما في الشّكل المجاور.

٢. تعبير (Postfix): طريقة لتمثيل العمليات الحسابيّة؛ حيث يقع رمز العمليّة بعد المُعامِلَيْن، والصّورة العامّة كما في الشّكل المجاور.

٣. تعبير (Prefix): طريقة لتمثيل العمليات الحسابيّة؛ حيث يقع رمز العمليّة قبل المُعامِلَيْن، والصّورة العامّة كما في الشّكل المجاور.

وهنا، سنتطرّق لطريقة تقييم/حساب التّعبير من نوع (Postfix) دون احتماليّة وجود أخطاءٍ في التّعبير، من خلال تتبّع الخوارزميّة الآتية:

1. البداية.

2. الإعلان عن مكدّسة فارغة.

3. لكلّ جزء (Token) من التّعبير (ابتداءً من اليسار إلى اليمين):

أ. هل الجزء يعبّر عن معامل(في العادة رقم)؟

- نعم: 1. ضع المعامل في المكدّسة

2. انتقل للفرع 3 لقراءة جزء جديد.

- لا: انتقل إلى الفرع (3-ب)

ب. هل الجزء عمليّة (Operator)؟

- نعم: 1. أخرج قيمتين من المكدّسة (تنفيذ POP مرتين).

2. احسب الناتج بناءً على العملية والقيمتين الخارجتين من المكدّسة، ثمّ ضع الناتج في المكدّسة.

4. هل انتهت جميع أجزاء التعبير؟

- لا: انتقل إلى الفرع 3

- نعم: 1. اطبع الناتج النهائي في المكدّسة.

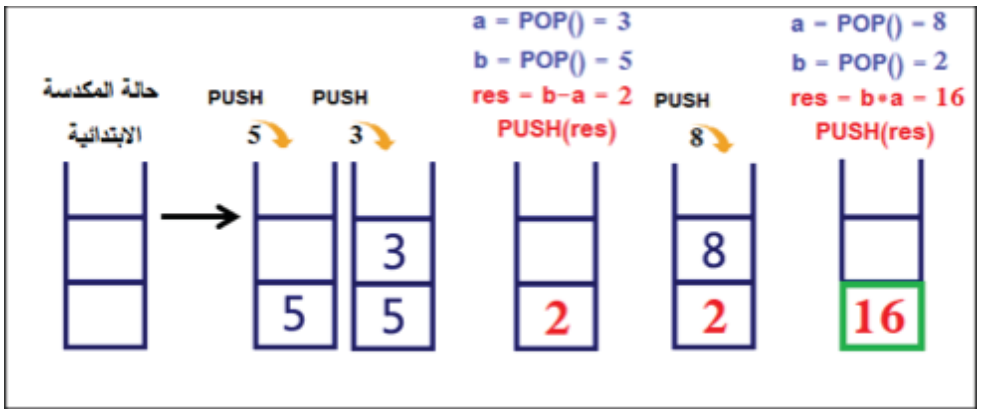
2. انتقل إلى فرع 5.

5. النهاية.

مثال

من خلال الرسم، احسب التّعبير الآتي: 5 3 - 8 *

الحل:



مثال

احسب التعبير الآتي: 6 2 + 5 * 8 4 / -

الحل:

6 2 + 5 * 8 4 / -

8 5 * 8 4 / -

40 **8 4 /** -

40 2 -

38

سؤال

احسب قيمة التعبير الآتي:

10 2 8 * + 3 -

1. ما المقصود بالمصطلحات الآتية: تركيب البيانات، المكدّسة.
2. عدّد أنواع التراكيب البيانية مع مثال على كل منها.
3. أعلن عن مكدّسة (A) سعتها 5 عناصر من نوع نصّي، وبيّن حالة ((Top، A(Top) بعد تنفيذ كلّ عمليّة من العمليّات الآتية بالرّسم على التوالي:

أ. إضافة العنصر S	و. إضافة عنصرين R، A
ب. إضافة العنصر A	ز. إضافة القيم L،L،Y
ج. ازالة عنصر واحد.	ح. إضافة العنصر P
د. إزالة عنصر واحد.	ط. إزالة قيمتين.
هـ. ازالة عنصر واحد.	ي. إزالة ثلاثة عناصر.

4. بالرّسم، قم بفحص اقواس التّعبير الآتي:(8 - (9 + (2 / 8)- 6 * 5) + 3
5. اكتب ناتج طباعة العناصر الخارجة من المكدسّة إذا علمت أنّ العناصر داخل المكدّسة هي كالآتي: 1،2،3،4. ما العلاقة بين ترتيب العناصر الداخلة للمكدّسة والخارجة منها؟
6. أعط بعض الأمثلة أو تطبيقات حياتية تمثّل مفهوم المكدّسة غير الواردة في الدرس.
7. احسب قيمة التعبير الآتي: **+ 3 ^ 2 * + / 2 8 3 - + 3 2 6**

ثانياً: الطّابـور (Queue)

في علوم الحاسوب، يُمثِّل الطّابور أحد أنواع تراكيب البيانات الخطيّة، والتي تتشابه في التّمثيل والبرمجة مع المصفوفات - التي درستها سابقا - بتعاملها مع البيانات كمجموعة من العناصر ذات نوع واحد فقط، إلّا أنّ عمليّات الحذف والإضافة بالطابور هي ما يميّزها عن المصفوفة؛ حيث يشترط أن تجرى عمليّة إضافة العناصر إلى الطابور من نهايته، بينما عمليّة الحذف تكون من بدايته؛ لذلك يطلق عليها الوصف (FIFO) اختصارا للتّعبير (First In، First Out)، أيّ أنّ أوّل العناصر المضافة إلى الطابور ستكون أوّل العناصر الخارجة منها لاحظ الشّكل أعلاه.

نشاط (٥) أمثلة على الطابور

أيّ المواقف الآتية يمثّل فكرة الطابور؟

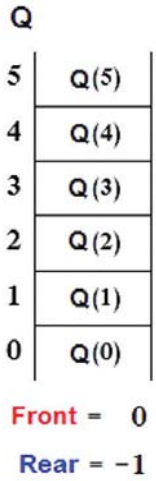
١) اصطفاف الطلبة بطابور الصباح.

٢) ترتيب الأطباق في خزانة المطبخ.

٣) طابور العملاء في البنك.

٤) وصول رسالة جديدة إلى هاتفك، وحذفها بناءً على قِدمها بصندوق الرسائل.

٥) اصطفاف السّيارات أمام إشارة المرور.



يُمثَّل الطابور كمصفوفة أحاديّة البعد، فمثلاً؛ إذا كان لديك طابوراً فارغاً اسمه (Q)، وعدد مواقعه ستة، فإنّه يتمّ تمثيلها كما في الشّكل المجاور.

يُشار إلى موقع آخر عنصر في الطابور (index) بمتغيّر يُعلن عنه مسبقاً (وليكن مثلاً Rear)، وتسند له القيمة (-1) ابتدائيّا، بينما يُشار إلى أوّل موقع في الطّابور بمتغيّر يُعلن عنه مسبقاً (وليكن مثلاً Front)، تُسند له القيمة (0) ابتدائيّا

بحيث تتغيّر القيمتان باستمرار خلال عمليّتَي الحذف والإضافة للعناصر (البيانات) في الطّابور.

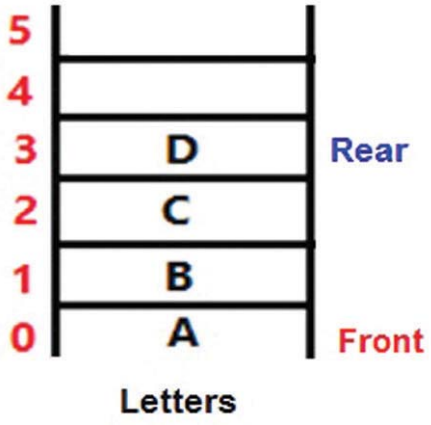
مثال

تمعّن الشّكل المجاور، ثمّ أجب عن الأسئلة الآتية:

أ. ما اسم الطابور؟ وما سعته؟

ب. ما نوع البيانات في الشكل؟

ج. ما قيمة المتغيّران Front، Rear في الشّكل؟



الحل:

أ. اسم الطابور: Letters، وسعته 6.

ب. بيانات نصّية.

ج. Front = 0، Rear = 3.

عمليّات على الطابور

عمليّة الإضافة.

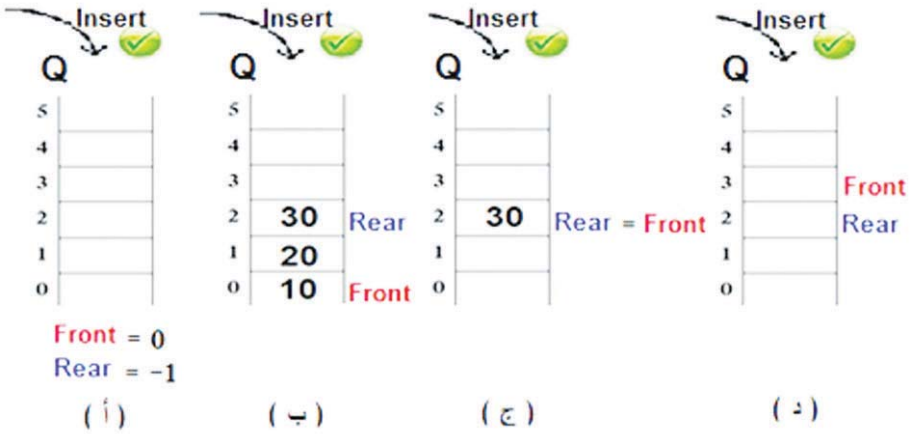
يطلق على عمليّة إضافة عنصر إلى طابور كما هو متعارف عليه لدى المبرمجين اللّفظ (Insert) أو اللّفظ (Enqueue)، ويمكن تلخيص حالات الإضافة إلى الطابور كما يأتي:

- الحالة الأولى:

أ. أن يكون الطّابور فارغا تماما (Empty) كما بالحالتين (أ) و (د) في الشّكل اللاحق، وهذا يعني أن يكون:

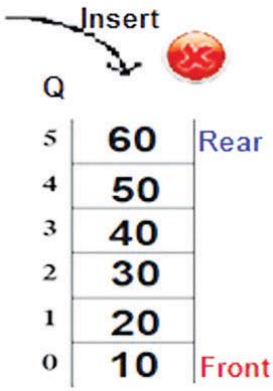
- Front = 0 و Rear = - 1.
- Front = Rear + 1.

ب. أن يحوي الطّابور على عنصر واحد فقط (**Front = Rear**) كما بالحالة (ج) في الشكل التالي.



ج. أن يحوي الطابور على بعض المواقع الفارغة Rear = N-1 و Front ≠ 0، كما هي بالحالة (ب) في الشكل أعلاه.

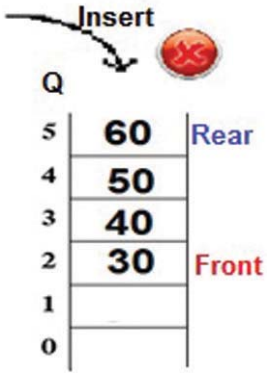
- الحالة الثانية:



الطابور ممتلئ (Full)، بمعنى أنّ عدد العناصر في الطابور يكافئ سعته، وهنا لا يمكن إتمام عمليّة الإضافة.

- الحالة الثالثة (الطابور الدائري)

عندما يكون Rear = N-1 و Front ≠ 0، بمعنى أنّ Rear يشير إلى آخر موقع بالطّابور وFront ليس بالموقع الأوّل، يمكننا الإضافة بجعل Rear ينتقل إلى بداية الطابور (Rear= (Rear+1) **Mod** N) ومن ثمّ إضافة العنصر المراد، إلّا أنّنا لن نتطرّق لهذه الحالة برمجيّا.



ويمكننا حساب عدد العناصر في الطابور(Q_N) من خلال العلاقة الآتية:

$$Q_N = Rear - Front + 1$$

مثال

لديك طابور سعته 10، احسب عدد عناصر الطابور لكلّ حالة من الحالات الآتية:

أ) Front = 2، Rear = 8 ب) Front = Rear = 5

ج) Front=3، Rear=2

الحل: أ) $Q_N = 8 - 2 + 1 = 6 + 1 = 7$ عناصر.

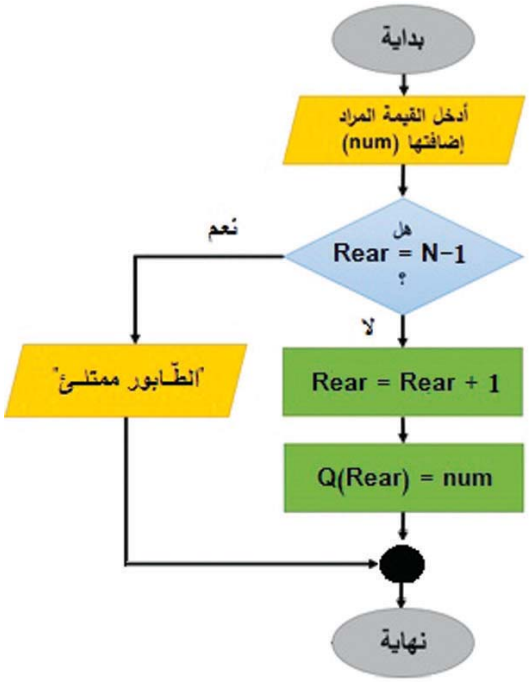
ب) $Q_N = 5 - 5 + 1 = 0 + 1 = 1$ عنصر.

ج) $Q_N = 2 - 3 + 1 = -1 + 1 = 0$ عنصر (الطابور فارغ).

يمكننا ترجمة ما سبق من خلال الخوارزميّة والمخطّط الانسيابي كما يأتي:

الخوارزميّـــــة **المخطّط الانسيابي**

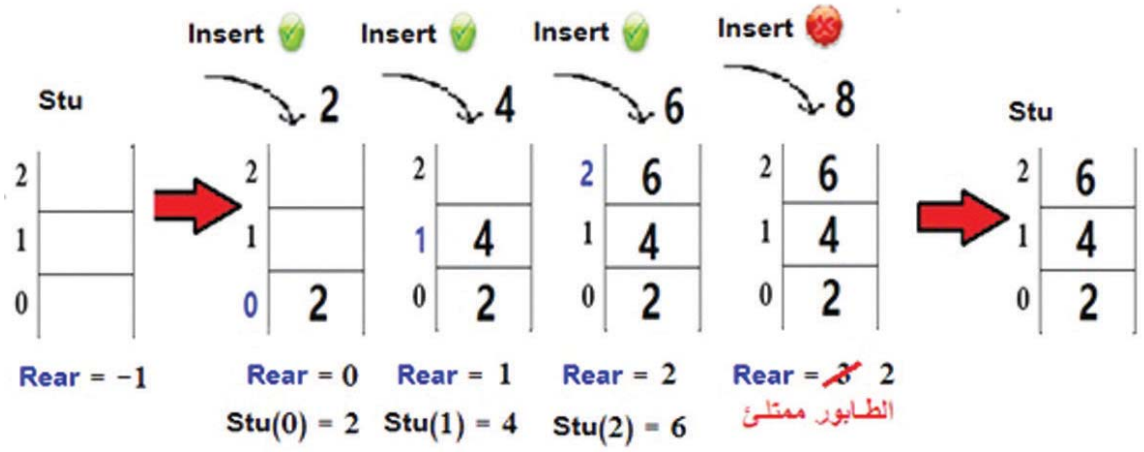
1. بداية.
2. أدخل قيمة العنصر المراد اضافته (وليكن num).
3. هل Rear = N−1؟
   - أ. نعم: – اطبع "الطابور ممتلئ"
     – اذهب الى الخطوة 6
   - ب. لا: – اذهب الى الخطوة 4
4. Rear = Rear + 1
5. Q(Rear) = num
6. نهاية.



مثال

لديك طابور اسمه (Stu) وسعته 3، مستعيناً بخوارزميّة الإضافة للطابور أعلاه، مثّل عمليّة الإضافة المتتالية للعناصر: 2، 4، 6، 8 موضّحا ذلك بالرّسم، ومحدّدا قيمة Rear، Stu(Rear) في كل مرّة.

الحل:



نشاط (٦) تتبّع خوارزميّة الإضافة إلى الطابور.

لديك طابور فارغ Employee (5)، من خلال الرّسم تتبّع قيمة Rear، Employee(Rear) لكلّ جملة من الجمل الآتية:

١. أضف للطابور العناصر 98، 1، 34، 56، 22 بشكلٍ متتالٍ.

٢. أضف العنصر 88، هل يمكن ذلك؟ لماذا؟

عمليّة الحذف (الإزالة)

يطلق على عمليّة حذف عنصر من طابور كما هو متعارف عليه لدى المبرمجين اللّفظ (Delete) أو (Remove) أو (Dequeue)، ويمكن تلخيص حالات الحذف من الطابور كما يأتي:

الحالة الأولى:

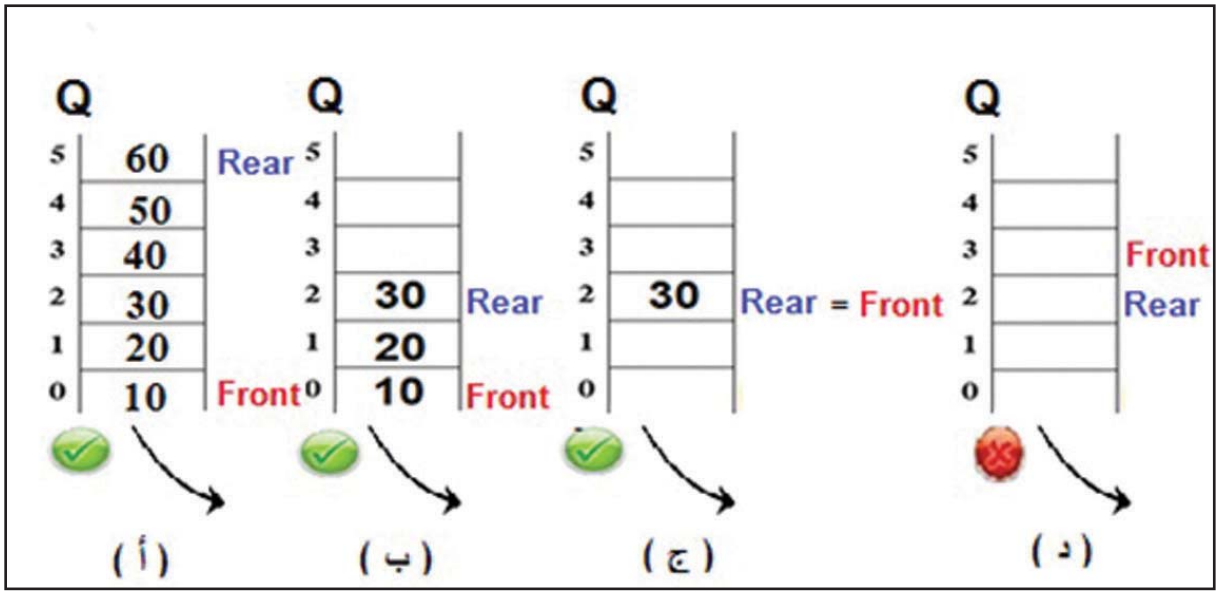
أ. أن يكون الطابور ممتلئاً (**Full**)، بمعنى أنّ (Front = 0، Rear = N-1)، كما بالحالة (أ) في الشّكل أدناه.

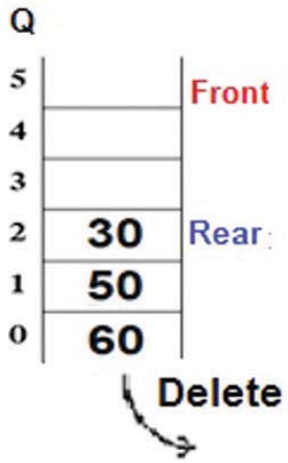
ب. أن يحوي الطابور بعض العناصر داخله (Front <= Rear) و (Rear < N-1) كما بالحالة (ب) في الشّكل أدناه.

ج. أن يكون بالطابور عنصرا واحدا فقط (Front = Rear)، وهنا يمكننا الحذف مرة واحدة فقط كما بالحالة (ج) في الشّكل أدناه.

الحالة الثانية:

الطابور فارغ (Empty)، بمعنى أنّ (Front > Rear) أو (Front=0، Rear=-1)، ولا يمكن إتمام عمليّة الحذف كما بالحالة (د) في الشّكل أدناه.





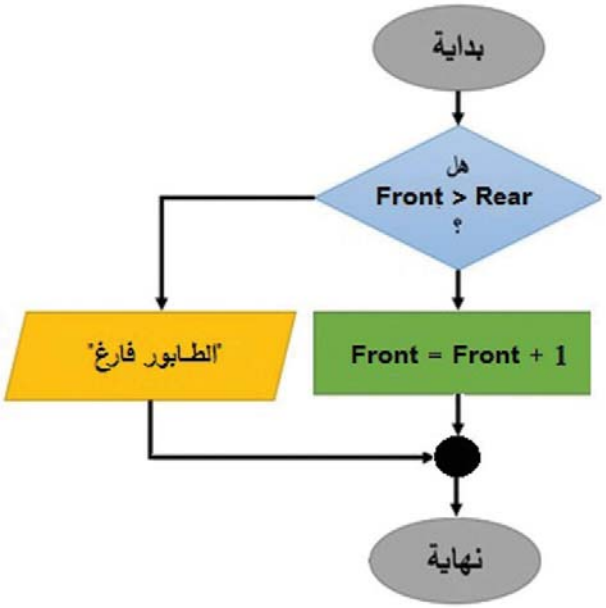
الحالة الثالثة (الطابور الدائري)

عندما يكون (Front > Rear) وهناك عناصر بالطابور، كما في الشّكل المجاور، ولن نتطرّق لمثل هذه الحالة برمجيّا.

ويمكن ترجمة ما سبق من خلال الخوارزميّة والمخطّط الانسيابي أدناه:

الخوارزميّـــــة **المخطّط الانسيابي**

1. بداية.
2. هل Front > Rear؟
   - نعم: اطبع "الطابور فارغ".
     اذهب إلى الخطوة 4
   - لا: اذهب إلى الخطوة 3
3. Front = Front + 1
4. نهاية.



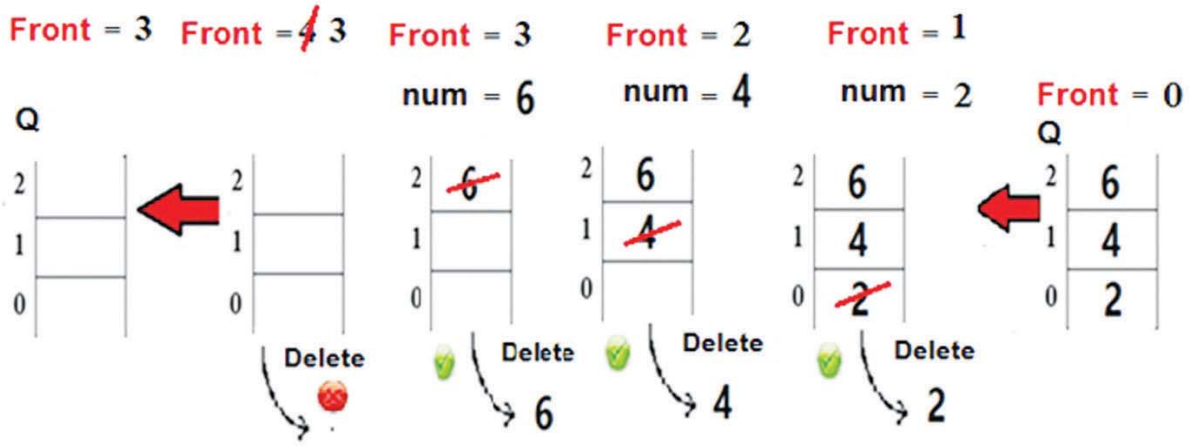
فكّر:

قم بإجراء التّعديلات اللّازمة بالمخطط الانسيابي أعلاه لطباعة العنصر المراد حذفه (وليكن num) قبل عمليّة الحذف.

مثال

لديك الطابور (Q) حجمه 4، ويحوي القيم (2،4،6). مستعينا بخوارزميّة الحذف من الطابور، مثّل عمليّة الحذف بالرّسم في حال قمنا بـ 4 مرات حذف، محدّدا قيمة كلّ من العنصر المحذوف، وقيمة Front في كل مرّة.

الحل:



ويمكن برمجة عمليّتي الإضافة والحذف للطابور بلغة VB.NET كما في المثال الآتي:

مثال

قم بتصميم الواجهة أدناه وضبط خصائصها، ثمّ كتابة الجمل البرمجيّة المناسبة لإضافة وحذف مجموعة عناصر(بيانات) من طابور(Queue) بشكل متتالٍ.

الحل:

١. إضافة نموذج ثان (Form2) إلى مشروع المكدسة السّابق.

٢. تعديل واجهة المشروع السابق باضافة زر أمر للانتقال الى Form2 الذي تمّت اضافته من خلال كتابة الجمل البرمجيّة الآتية في نافذة الأوامر:

```
#Region "To Form1 Button"
Private Sub Button4_Click(sender As Object، e As EventArgs) Handles Button4.Click
    Me.Hide()
    Form2.Show()
  End Sub
#End Region
```

٣. تصميم الواجهة وضبط خصائص الأدوات والأزرار الظاهرة في المثال الحالي كما يأتي:

أدوات واجهة التصميم	الخصائص	الخصائص الفرعية	القيم
Form1	**Icon**		(صورة مناسبة من نوع Ico)
	Language		Arabic (Palestinian Authority)
	RightToLeft		No
	Size	Width	800 px
		Height	470 px
	Font	Name	Traditional Arabic
		Size	8
		Bold	True
GroupBox1	**Font**	Name	Traditional Arabic
		Size	20
		Bold	True
	Size	Width	350
		Height	200
	RightToLeft		Yes
	Text		الاضافة إلى الطابور
Label1	Text		العمليات على الطابور
	Size	Width	328
		Height	61
	Font	Name	Traditional Arabic
		Size	30
		Bold	True
TextBox1	Text		" "
	TextAlign		Left
	Size	Width	525
		Height	39
	Backcolor		Info

نشاط (٧) ضبط خصائص الأدوات.

أكمل ضبط خصائص الأدوات للعناصر المتبقية لواجهة التّصميم السابقة محدّدا القيم المناسبة للخصائص المستخدمة لكل منها.

٤. فتح نافذة البرمجة للنّموذج Form 2 وكتابة الجمل البرمجيّة الآتية:

```
Private Sub Form2_Load (sender As Object، e As EventArgs) Handles MyBase.Load
    TextBox1.Enabled = False
    TextBox2.Enabled = False
    TextBox3.Enabled = False
    TextBox4.Enabled = False
    TextBox5.Enabled = False
    TextBox6.Enabled = False
    TextBox7.Enabled = False
    TextBox1.Text = “”
    TextBox2.Text = “”
    TextBox3.Text = “”
    TextBox4.Text = “”
    TextBox5.Text = “”
    TextBox6.Text = “”
    TextBox7.Text = “”
      End Sub
```

٥. إضافة الجمل البرمجيّة الآتية إلى منطقة التّعريفات العامّة في نافذة البرمجة للنّموذج Form2:

```
#Region “Declare Queue”
Dim QU (3) As Integer          ' الاعلان عن طابور حجمه 4
  Public Front As Integer = 0      ' الاعلان عن مؤشر الحذف
        Public Rear As Integer = -1    ' الاعلان عن مؤشر الاضافة
      #End Region
```

٦. فتح نافذة البرمجة لزر الأمر (Insert) وكتابة الجمل البرمجيّة الآتية:

```
#Region "Insert Button"
Private Sub Button1_Click (sender As Object، e As EventArgs) Handles Button1.Click
    GroupBox1.Visible = True
    GroupBox2.Visible = False
    Insert ()
 End Sub
 Public Sub Insert ()
    Dim num As Integer
    If Rear = 3 Then
    MessageBox.Show ("الطابور ممتلئ!!! لا يمكن الإضافة") ' في حالة الطابور ممتلئ
    Else                                                  ' في حالة الاضافة ممكنة للطابور
    TextBox2.Text = Rear
    Rear += 1
    TextBox4.Text = Rear
    num = InputBox("أدخل القيمة الصحيحة المراد اضافتها")   ادخال قيمة من خلال صندوق الادخال
    TextBox3.Text = num
    QU(Rear) = num

    TextBox1.Text = TextBox1.Text & " " & QU(Rear)
    End If
End Sub
#End Region
```

٧. فتح نافذة البرمجة لزر الأمر (Delete) وكتابة الجمل البرمجيّة الآتية:

```
# Region "Delete Button"
Private Sub Button2_Click(sender As Object، e As EventArgs) Handles Button2.Click
    GroupBox1.Visible = False
    GroupBox2.Visible = True
    Delete()                    ' استدعاء دالة الحذف
    TextBox1.Text = ""
    For i = Front To Rear
    TextBox1.Text = TextBox1.Text & " " & QU(i)                '
 Next                طباعة محتويات الطابور بعد عملية الحذف
End Sub
 Public Sub Delete()
   Dim num As Integer ' المتغير المراد حفظ العنصر المحذوف داخله
If Front > Rear Then
   If Rear = 3 Then      ' لحل مشكلة الطابور الدائري بالرجوع الى القيم الابتدائية
     Rear = -1
     Front = 0
   End If
   MessageBox.Show  ("الطابور فارغ!!! لا يمكن الحذف")
Else
   num = QU(Front)           ' قيمة العنصر المراد حذفه
```

```
    TextBox6.Text = num          ' طباعة قيمة العنصر المراد حذفه
    TextBox5.Text = Front        'طباعة قيمة المؤشر Front قبل عملية الحذف
    Front += 1                   ' تمّ الحذف
    TextBox7.Text = Front        ' طباعة قيمة المؤشر Front بعد عملية الحذف
End If
End Sub
#End Region
```

٨. فتح النافذة البرمجية لزر أمر "الخروج" وكتابة الجمل البرمجيّة الآتية:

```
#Region "Exit Button"
Private Sub Button3_Click (sender As Object، e As EventArgs) Handles Button3.Click
    End
    End Sub
#End Region
```

٩. فتح نافذة البرمجة لزر أمر الانتقال إلى نموذج1 (Form1)، وكتابة الجمل البرمجيّة الآتية:

```
#Region "To Form1 Button"
 Private Sub Button4_Click (sender As Object، e As EventArgs) Handles Button4.Click
   Me. Hide ()
   Form1.Show()
 End Sub
#End Region
```

قم بتنفيذ البرنامج، ماذا تلاحظ؟

أسئلة الدرس

1. ما المقصود بالطابور؟
2. عدّد حالات الاضافة الى الطابور.
3. اذكر حالات الحذف من الطابور.
4. عدّد ثلاثة من تطبيقات الطوابير في الحياة العمليّة.
5. في دائرة حكومية نريد أن ننظم المراجعين بحيث تقدّم الخدمة لهم حسب وقت قدومهم، وارتأت الدائرة أن تعطي أرقاما للمراجعين مبتدئة بالرقم 200.
   أ. بماذا نمثّل طريقة الخدمة للمراجعين.
   ب. مثّل الوضع بالرسم في حال قدوم 3 أشخاص.
   ج. مثّل الوضع بالرسم عند اتخاذ خدمة اثنين منهم.
6. اكتب الخوارزمية ثمّ ارسم المخطط الانسيابي لكلّ من:
   أ. حالة الإضافة الى طابور.
   ب. حالة الحذف من طابور.

الدرس (٢)

البرمجة الهدفيّة
(Object Oriented Programming-OOP)

تعلّمت سابقاً بيئة البرمجة المرئيّة، وتمّ تناول لغة "VB.NET" كأحد لغات البرمجة المستخدمة لأسلوب البرمجة الإجرائية المعتمدة على تسلسل الجمل البرمجية والإجراءات أثناء كتابة البرامج والتي تركز على العمليّات والبيانات بالتزامن مع التّصميم الجيد لواجهات المستخدم (UI) وما يسبقها من خوارزميّات ومخطّطات ضمن دورة حياة تطوير البرمجيّات (SDLC).

كما قمتَ في صفّ سابقٍ بكتابة الكثير من البرامج بالنّمط الإجرائي، ويمكن تلخيص خصائصها كما يأتي:

الخصائص العامّة	البرمجة الاجرائية (Procedural Programming)
شكل البرنامج العام	ينقسم البرنامج إلى أجزاء صغيرة تسمى وظائف (Procedure).
منهجيّة البرمجة	من أعلى إلى أسفل (Top – Down).
انتقال البيانات وحركتها	يمكن للبيانات التّنقل بحرية بين أجزاء النّظام المختلفة.
الوصول إلى البيانات (Data Access)	الوصول للبيانات وانتقالها بحريّة من إجراء إلى آخر في النظام.
إخفاء البيانات	لا تدعم إخفاء البيانات (أقل أمانا).

البرمجة الهدفية

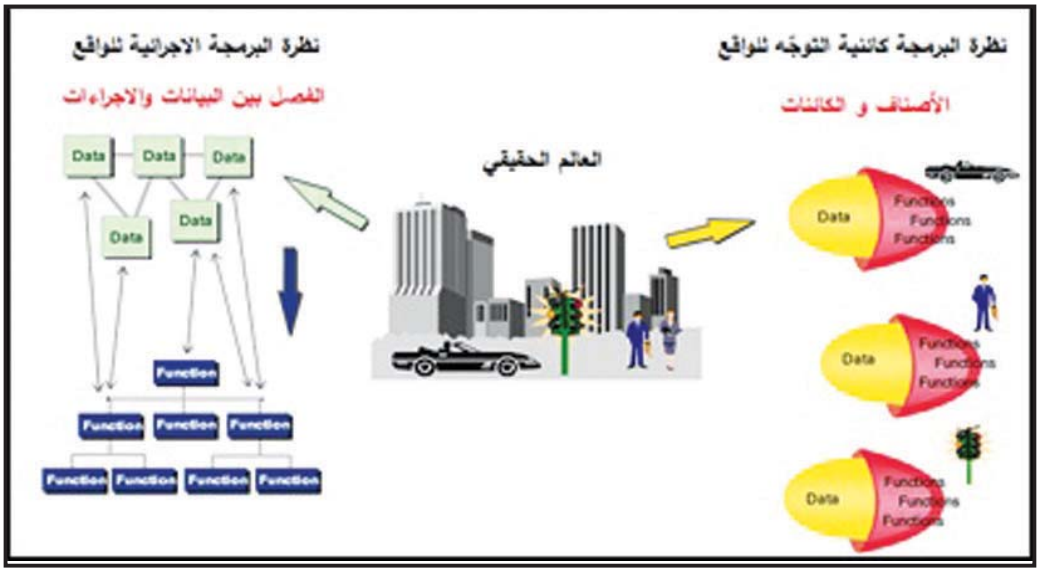
قد لا يقدّم النّمط البرمجي الواحد أفضل الحُلول للمشكلات والمسائل لدى المبرمج من أجل الوصول إلى مستوىً مُرضٍ من حيث: التّصميم، والبناء، وأداء المهامّ المَنُوطة بالبرنامج، ونتيجةً لذلك؛ ظَهرت الحاجة إلى لغات متعدّدة الأنماط البرمجيّة كما هو الحال بلغة (VB.NET) والتي تدعم نمطاً برمجيّاً آخر يُطلق عليه "البرمجة الهدفيّة" لتصبح البرمجة من خلالها أكثر تخصّصاً ودقّة في محاكاة الواقع الحقيقيّ.
تُعرّف البرمجة الهدفيّة على أنّها أسلوب (نمط) برمجي يعتبر "الأشياء" مكوّنة من مجموعة صفات وسلوكيات تحدّد هويّتها المميّزة، تتأثّر بالمحيط حولها، وتتعامل معه ضمن إمكاناتها المصمّمة لتنفيذها.

ملاحظة:

للبرمجة الهدفية مسمّيات أخرى، مثل: البرمجة الغرضيّة، البرمجة الشيئيّة، برمجة الكائنات، البرمجة كائنيّة التوجّه.

قضية مناقشة:

اعتماداً على الشّكل أدناه، وبمساعدة معلمك، ناقش مع زملائك منظور النّمط الإجرائي والنّمط الهدفيّ للبيانات والعمليّات في العالم الواقعي.



قضية بحث:

ابحث في شبكة الانترنت عن أنماط أخرى في لغات البرمجة، مبيّنا الفروق الأساسية بينها، ثم اكتب تقريراً حولها، وسلّمه لمعلمك.

إيجابيّات استخدام البرمجة الهدفيّة

من إيجابيّات استخدام البرمجة الهدفيّة في البرمجة، نذكر ما يأتي:

1. **وصف المسألة الواقعية ببساطة ووضوح:** استخدام النّماذج والكائنات والأصناف يقلّل من درجة تعقيد المسألة، ويصبح التّركيب البرمجيّ أكثر وضوحاً، ممّا يقلّل عدد الجمل البرمجيّة والزّمن اللّازم لإتمام عمليّة البرمجة.

2. **تحسين بنيويّة البرنامج (Modularity):** تقسيم المسألة الكبيرة إلى مجموعة وحداتٍ برمجيّة صغيرة تقوم كلّ منها بوظيفة محدّدة وضرورية، ممّا تؤدي إلى تبسيط المشكلة لتصبح عمليّة تتبّع الأخطاء وتصحيحها وتطوير الحلول مستقبلاً أكثر سهولة.

3. إعادة استخدام الجمل البرمجيّة (Reusability): يوفّر على المبرمج عناء إعادة كتابة الجمل البرمجيّة مرّة تلو الأخرى؛ من خلال السماح باستخدامها مباشرة أو التّعديل في وظيفتها أو إضافتها دون إعادة كتابتها بما يتناسب مع طبيعة حلّ المشكلة من خلال مفاهيم الوراثة وتعدّد الأشكال التي سنتعلّمها لاحقاً.

4. قابليّة الصيانة للبرامج (Maintainability): تقسيم البرنامج الى أجزاء صغيرة (كائنات) يجعل من عمليّة تحديد الأخطاء في البرنامج المصدري وصيانته أكثر سرعة وسهولة.

سؤال

عدّد إيجابيّات أخرى لاستخدام البرمجة الهدفية في البرامج.

مبادئ البرمجة الهدفيّة.

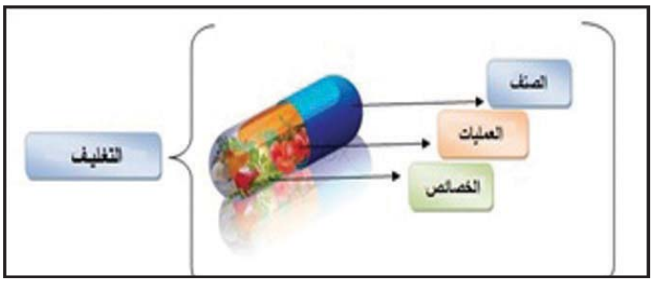
تعتمد البرمجة الهدفيّة في بنائها وتصميمها على أربع قواعد أساسية، لا بدّ من مراعاتها أثناء القيام بالبرمجة وحلّ المشكلات، وهي: التجريد، والتغليف، والوراثة، وتعدّد الأشكال.

1. التّجريد (Abstraction): عمليّة إخفاء تفاصيل وتعقيدات البرمجة غير المهمّة عن المستخدم وإظهار المهمّة منها. فمثلا؛ الرّياضيّ يمارس نشاط الرّكض فتظهر حركة الأيدي والأرجل دون التمكّن من مشاهدة تفاصيل الجهاز العضليّ وتفاعله مع وغيره من أعضاء الجسم الدّاخلية، حيث تقتصر المشاهدة على النتائج المهمّة (الرّكض) دون الخوض في تفسير تفاصيل العمل.

سؤال

وضّح مفهوم التّجريد في آلة ATM، التّلفاز، السيّارة؟

٢- التغليف (Encapsulation):

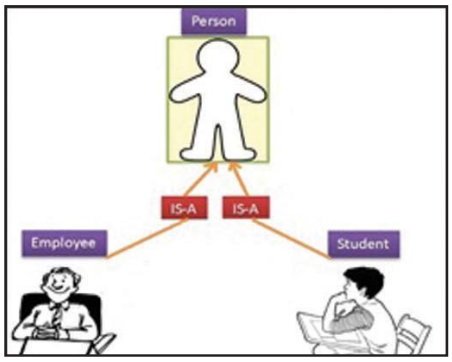


تهدف عمليّة التّغليف إلى إخفاء البيانات والدّوال، وآليّة عملها عن المستخدم بتحديد مستوى صلاحيّات معيّنة؛ بهدف الحماية وتحقيق الأمان، ومثال ذلك: الصّنف في البرمجة الهدفيّة، وعمليّات التّشفير للبيانات.

والجدول أدناه يوضّح الفروق بين مفهوميّ التّجريد والتّغليف، على النحو الآتي:

	التجريد	التغليف
1.	إخفاء كيفية العمل.	إخفاء المكوّنات (البيانات والدّوال) في وحدة واحدة.
2.	يكون من خلال واجهات المستخدم.	يكون من خلال محددات الوصول (Private, Public..).
3.	حلّ المشكلات على مستوى التصميم.	حل المشكلات على مستوى التّنفيذ.

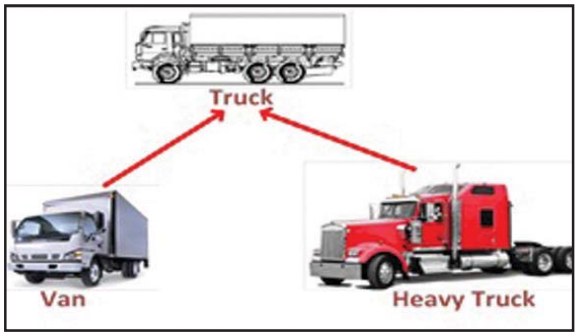
٣- الوراثة (Inheritance):



يتمّ السّماح للأبناء بوراثة خصائص الأب وسلوكه. حيث يمتلك الابن خصائص وسلوكيّات خاصّة به، إضافة إلى مجموعة الصفات والسلوكيّات الموروثة والمشتركة مع الأب، وهي تدعم مفهوم إعادة الاستخدام. وتكون العلاقة بينهما هي **(is -a)**. فمثلا؛ نقول:
(Employee **is a** Person)، (Student **is a** Person).

نشاط (١) مفهوم الوراثة.

بالاعتماد على الشّكل المجاور:



أ. حدّد: أيّ الجمل الآتية صحيحة وأيّها خاطئة؟ مع التّعليل.

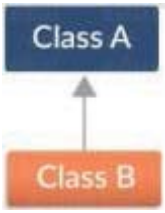
1. Van **is a** Truck.

2. Truck **is a** Heavy Truck.

ب. أيّهما يمثّل: صنف الابن، صنف الأب؟

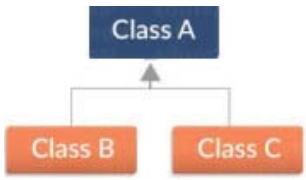
أنماط الوراثة

من أنواع (أنماط) الوراثة المستخدمة في لغات البرمجة:

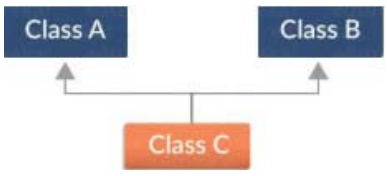


أ- **الأحادية:** يرث صنف الابن صفات وعمليات من صنف أب واحد فقط.

ب- **المتتالية:** يرث صنف الابن من عدّة مستويات عليا، وبشكل عمودي مباشر وغير مباشر، كما هو الحال في لغة(C++).



جـ- **الهرميّة:** وراثة عدّة أصناف من صنف واحد، بمعنى؛ وراثة عدّة أصناف للأبناء من صنف أب واحد.

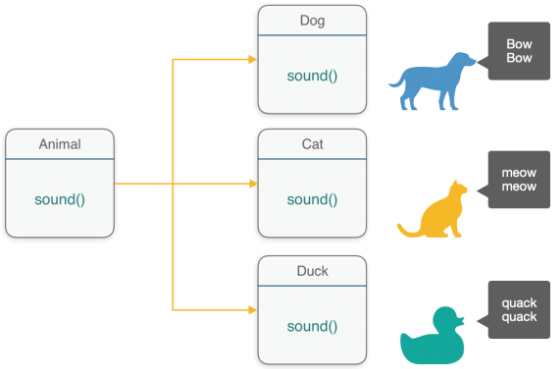


د- **المتعدّدة:** لا تدعم (VB.NET) هذا النّوع من الوراثة الذي يرث فيه صنف الابن من عدّة أصناف أعلى منه. لاحظ الشّكل المجاور.

ملاحظة:

يطلق على صنف الأب مفهوم (Super Class) أو (Parent Class)، ويطلق على صنف الابن مفهوم (Sub Class) أو (Child Class) بالترتيب.

٤- تعدّد الأشكال (Polymorphism):



مجموعة الخصائص والسلوكيات مختلفة الوظيفة ومتشابهة الاسم المستخدَم. فمثلاً؛ إذا طُلب إلى مجموعة طلبة حساب مِساحة الشكل الهندسي، فستختلف طريقة التنفيذ بناء على الشّكل الهندسي الذي قام كلُّ طالب بحسابه (مربع، مستطيل، دائرة...) حيث أنّ كلًّا منها يمثّل (شكلاً هندسيّاً)، لكنّها مختلفة في طريقة تنفيذ الحسابات والنّاتج.

قضية مناقشة:

على ضوْء مفهوم تعدّد الأشكال، ناقش وزملاءك الشّكل أعلاه.

ومن أبرز آليّات التّعامل مع مفهوم تعدّد الأشكال نذكر ما يأتي:

أ - آليّة الحمل الزائد (Overloading): ويقصد بها القدرة على تعريف أكثر من دالّة داخل الصّنف الواحد بالاسم ذاته، شريطة اختلاف عدد المعاملات (Parameters)، أو نوعها داخله، ومثال ذلك:

```
Public Function GetWord () As String
Public Function GetWord (ByVal Position As Integer) As String
Public Function GetWord (ByVal Search As String) As String
```

يهدف استخدام هذه الطريقة إلى استدعاء الدّالة بغضّ النّظر عن نوع البيانات، بدلاً من الحاجة إلى كتابة دوالّ جديدة، أو عمل تسميات مختلفة لطريقة العمل والتّنفيذ ذاتها. ومن تطبيقاتها: البنّاءات (Constructors) التي سيتمّ التطرّق إليها لاحقاً.

ب. آليّة الهيمنة (Overriding): يكمن الهدف الأساسي منها الحاجة لإضافة دوال جديدة إلى صنف الأب أو إلغائه، ووضع دوالّ جديدة في صنف الابن؛ ولكي نحقّق ذلك برمجيّاً لا بدّ من استخدام الكلمة المحجوزة **(Overrides)**، واستخدام دالّة أو أكثر في صنف الابن، لها الاسم نفسه في صنف الأب، مع ضرورة اختلاف التّطبيق البرمجي في صنف الابن.

ولكن، كيف يمكن تحقيق مجموعة مبادئ البرمجة الهدفيّة السّابقة؟

بالتأكيد يكون من خلال عمليّة البرمجة؛ فالبرنامج المكتوب بصورة البرمجة الهدفيّة يتكوّن من ركيزتين أساسيّتين تحقّقان مبادئه، وهما: الصنف، والكائن، وهذا ما سيتمّ تناوله.

الصّنف/الفئة (Class):

يمثّل الصّنف العمود الفقري لنمط البرمجة الهدفية، حيث يعرّف على أنّه وصف (مخطّط) عام للمسألة (الواقع) المراد محاكاتها من خلال ما يحويه من صفات (Attributes)، وسلوكيات (Methods). فمثلاً؛ للطلبة مجموعة صفات مثل: الاسم، وتاريخ الميلاد، والصف الدراسي، والعلامات، وبعض السلوكيات، مثل: حساب المعدل، حساب العمر، طباعة بطاقة تعريفية لكلّ منهم وهكذا.

ملاحظة:

- يمثّل كلّ صنف جديد يتمّ إنشاؤه نوعاً جديداً من أنواع البيانات (Data Types).
- تقابل (الصفات/الخصائص) مفهوم المتغيّرات، و(السلوكيات/العمليات) مفهوم الإجراءات والدّوال في البرمجة الإجرائية التي درستها في صفّ سابق.

مثال أذكر صفات(خصائص) وسلوكيّات(عمليّات) صنف مستطيل.(Rectangle)

الحل:

الصفات: طول القاعدة (Base)، الارتفاع (Height)، اللّون(color).

السّلوك: حساب مساحة المثلثArea()، حساب محيط المستطيل().Perimeter

قضية مناقشة:

على ضوْء مفهوم الصّنف الذي تعلّمته سابقاً، ناقش وزملاءك الشّكل أعلاه.

مثال

اقترح مجموعة من الصفات (الخصائص) والسلوكيات (العمليات) لصنف (انسان).

الحل:

الصفات (Attributes)	السلوكيات (Methods)
الاسم	المشي ()
العمر	الأكل ()
الطول	الجلوس ()
لون البشرة	الجري ()
الوزن	الوقوف ()

نشاط (٢) تحديد صفات وسلوكيات أصناف مختلفة.

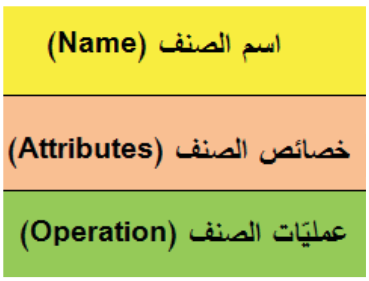
اقترح عدداً من الصفات والسلوكيّات المناسبة لكلّ صنف من الأصناف أدناه.

	الصنف	الصفات	السلوكيات
٠١	حيوان (Animal)		
٠٢	شكل هندسي (Shape)		
٠٣	حاسوب (computer)		

مخطّط الصنف (Class Diagram - UML)

درست في صف سابق مفهوم لغة النمذجة الموحدة (UML) كلغةٍ رسوميّة تجريديّة قياسيّة عامّة الاستخدام تصف المكوّنات الرئيسة للنظام، كما تعرّفت على مخطط الصنف(Class Diagram) أحد أشهر المخططات الهيكليّة والذي يعرض الأصناف بنظام معيّن والعلاقة فيما بينها.

يتألف مخطّط الصّنف من ثلاثة أقسام رئيسة كما في الشّكل المجاور.



١- اسم الصنف: أمثلة ذلك: (سيارة، حيوان, clsShape, clsStudent).

٢- صفة الصنف: مجموعة الخصائص المقترحة لذلك الصنف، مع تحديد النوع (Type)، ومحدّد الوصول لكلّ منها، أمثلة ذلك:

(# رقم الطالب: عدد صحيح) ، (- Height: Double).

٣- سلوك الصنف: الإجراءات والدوال التي تحدّد الوظائف والعمليّات التي يقوم بها الصنف، مع ضرورة تحديد نوعه، ومحدّد وصوله، واسم ونوع المعاملات ونوع القيمة الراجعة –إن وُجد-، مثال ذلك:

(+ حساب_حجم_الاسطوانة (عدد صحيح، عدد صحيح)، (- ClculateAge(int,int,int)).

ملحوظة:

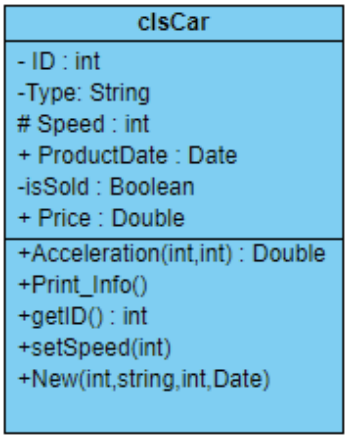
تشير الرّموز في رسم UML إلى نوع محدّد الوصول، كما يأتي:

الرمز	نوع محدّد الوصول
+	Public – عام
–	Private – خاص
#	Protected – محمي

مثال

ارسم UML - Class Diagram لصنف سيارة (clsCar) مع اقتراح صفات وعمليّات مناسبة.

الحل:



نشاط (٣) صنف حساب بنكي (BankAccount)

ارسم مخطّط (Class Diagram - UML) لصنف حساب بنكي (BankAccount) ذات الصفات والعمليّات الموضّحة في الجدول أدناه.

BankAccount	
الخصائص	العمليّات
– رقم العميل (Customer_ID)	– السحب (Withdraw)
– اسم العميل (Customer_Name)	– الايداع (Deposite)
– رصيد الحساب (Customer_Balance)	– طباعة معلومات (PrintInfo)

- اقترح أنواع ومحدّدات وصول ومعاملات مناسبة لكلّ منها.
- اقترح صفات وعمليات أخرى غير التي ذكرت بالجدول أعلاه.

إنشاء صنف برمجيّاً

يتكوّن الصنف بشكلٍ عام برمجيّا من المكوّنات الرئيسة الآتية:

أ. اسم الصنف: والذي يحقّق مبدأ التغليف.

ب. منطقة الصفات.

ج. منطقة الدوال المتعلّقة بالصفات (Property).

د. منطقة دوال العمليات/السلوكيّات.

هـ. منطقة البنّاءات (Constructors).

وفي حال تعدّدت الأصناف، فإنّه لا بدّ من تحقيق: الوراثة، وتعدّد الأشكال، ويمكن الوصول إلى مكوّنات الأصناف المختلفة في المشروع عمليّا من خلال الكائنات (Objects)، وواجهات المستخدم، وتعتمد درجة وصول المستخدم على محدّدات الوصول التي فرضها المبرمج ببرنامجه.

مقترح مشروع فرع تكنولوجي (Technology Branch)

- وصف المشروع.

سنبدأ بتصميم وبرمجة برنامج يقوم بعرض بيانات طالب في الفرع التكنولوجي من خلال صنف الأب المسمّى (clsStudent)، بحيث يحوي على بعض صفاتٍ للطالب، مثل: رقم الطالب، واسمه، وتاريخ ميلاده؛ وسيتمّ إدخال علامات الطالب الحاصل عليها في المدرسة من خلال زرّ (صندوق الإدخال)، مع تخزين أسماء المباحث في مصفوفة أحاديّة خاصّة تحوي (8) مباحث محدّدة مسبقاً، مع العلم أنّ العلامة القصوى لكلّ مبحث (100 علامة) باستثناء مبحثٍ واحدٍ تكون علامته القصوى (200 علامة).

يتمّ احتساب المعدّل والنتيجة من خلال دوال مناسبة؛ ثمّ يتمّ عرض جميع ما سبق من خلال أداة عارض البيانات (DataGridView)، وتكون النتيجة (ناجح أو راسب).

كذلك، يُراد تصميم وبرمجة صنفٍ آخر يسمّى (clsTechnology) لإدخال علامات مباحث طلبة الفرع التكنولوجي من خلال صناديق نص (Textbox)، ويتمّ ضمن الشّروط الآتية:

أ- احتساب معدّل علامات المباحث الإجبارية (البرمجة والأتمتة/200، اللغة العربية/100، اللغة الانجليزية/100، الرياضيات/100).

ب. احتساب معدّل أعلى علامة في 3 مباحث من أصل (4) مباحث أساسية (مبحث الاتّصالات/100، التربية الدينية/100، الثقافة العلمية/100، الإدارة والريادة/100)، بحيث يكون المعدل للمباحث الأساسية يساوي علامة مبحث الاتصالات وأعلى علامة مبحث من المباحث المتبقّية.

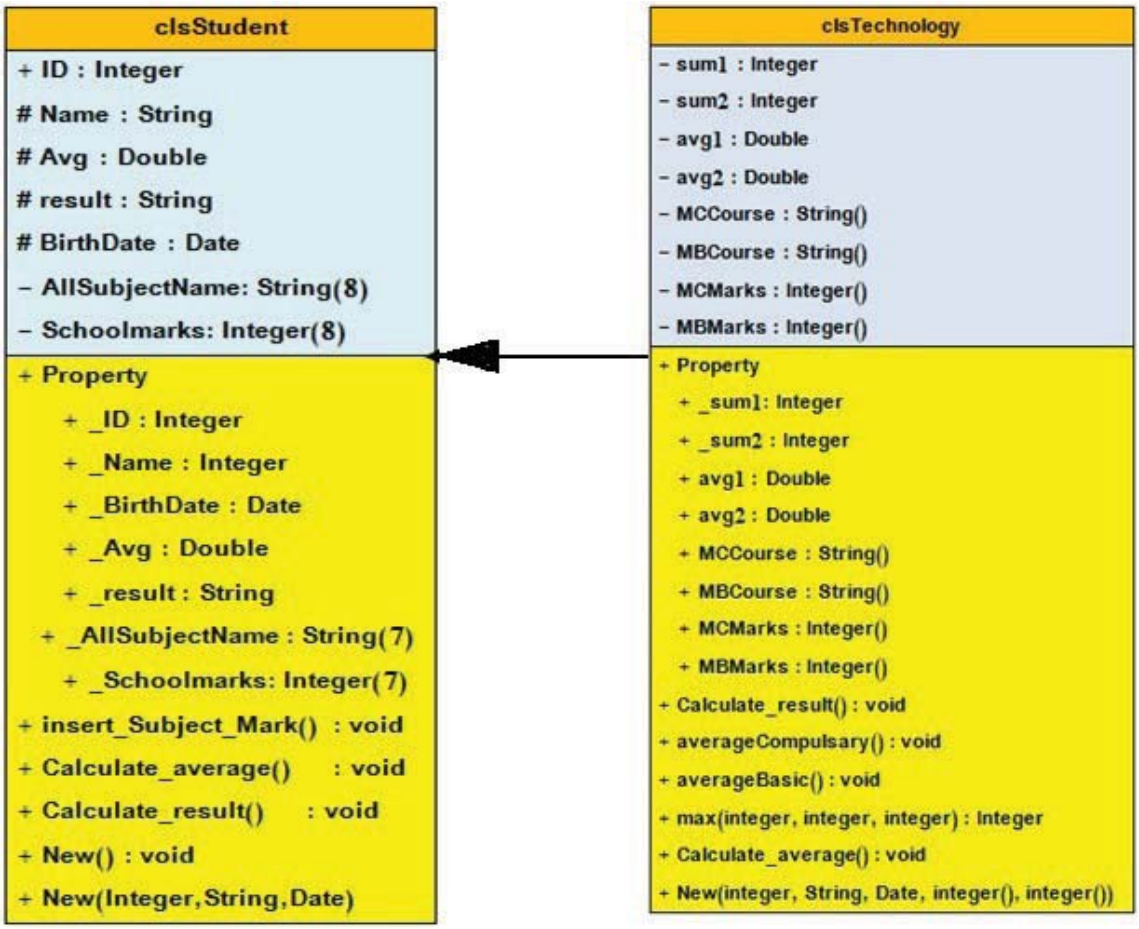
ج. احتساب المعدّل العام للمباحث الستة، والنتيجة السنوية.

د. تكون قيم النتيجة السنوية كالآتي: (ناجح) في حال نجح الطالب بجميع المواد، و(راسب) في حال رسوبه بإحدى المباحث أو جميعها.

هـ. عرض النتائج على أداة عارض البيانات (DataGridView)

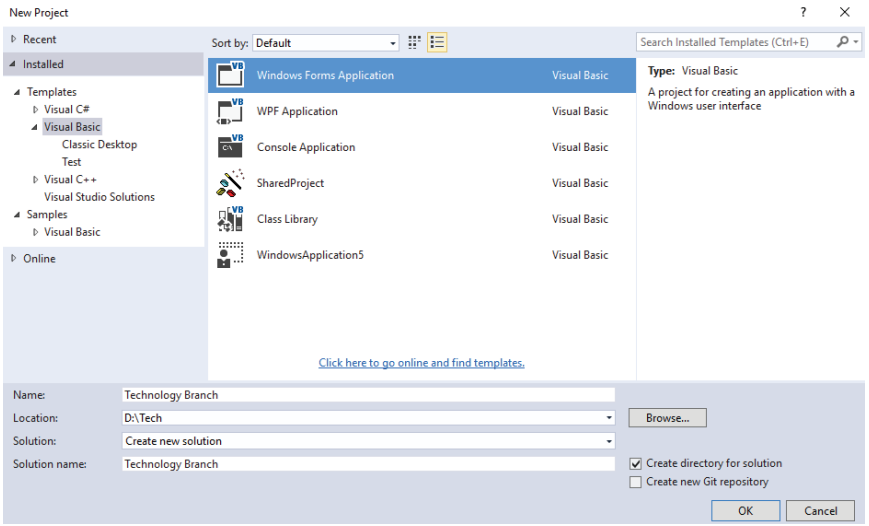
أوّلاً: رسم مخطّط (Class Diagram - UML)

قبل البدء بتنفيذ المشروع، لا بدّ من القيام برسم مخطّط ClassDiagram – UML الخاصّ به، وهو على الشّكل الآتي:



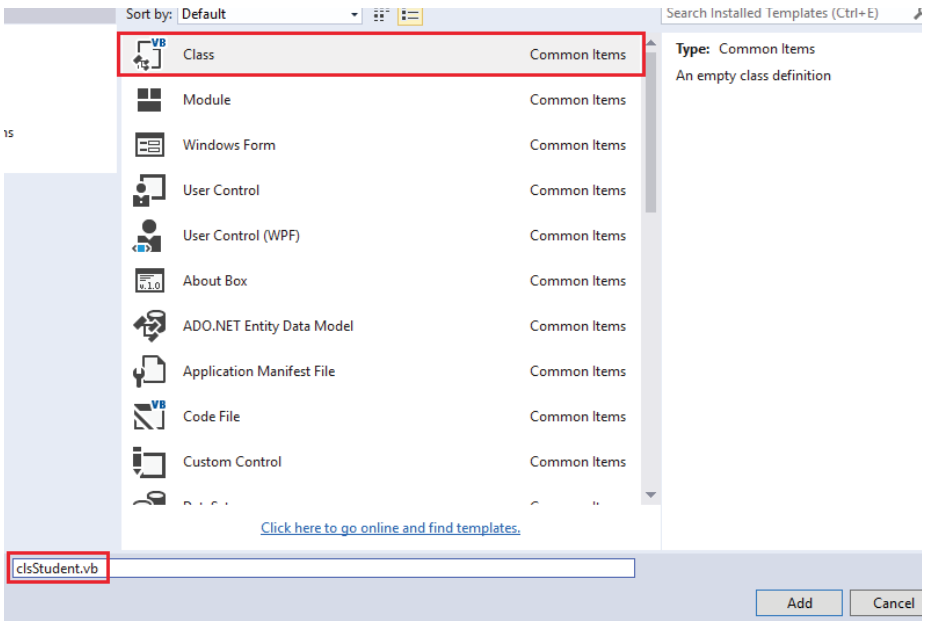
ثانياً: لتنفيذ مشروع برمجيّا، نتّبع الخطوات الآتية:

1. إنشاء مشروع جديد (New Project...) كما تعلّمت سابقا في بيئة Visual Studio.NET بلغة VB.NET، وتسمية المشروع (Technology Branch) وتحديد مكان حفظه، كما في الشّكل أدناه.



2. تحليل النّظام ومتطلّباته الواقعيّة ثمّ البدء بعمليّة التّصميم المبدئي للمشروع كما في الشّكل أدناه.

3. من قائمة (Project) نختار (Add New Item...) ثمّ اختيار صنف جديد (Class) وتسميته (clsStudent)، ثمّ الضّغط على زرّ (Add) كما في الشّكل أدناه.

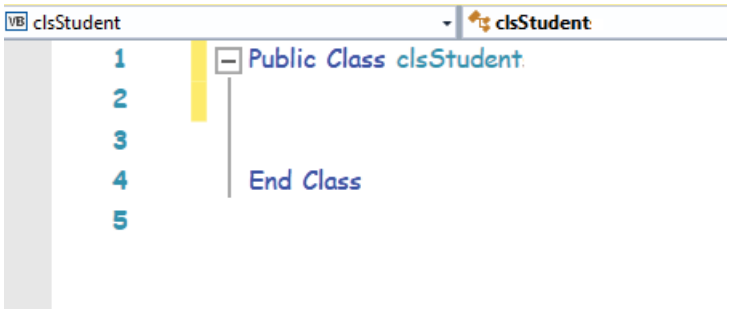


ويمكن إضافة صنف جديد من خلال شريط القوائم أو Solution Explorer.

نشاط (٤) إنشاء صنف جديد برمجيّا.

أَضِف صنفاً جديداً باسم "clsTechnology" إلى مشروعك.

4. بعد الانتهاء من الخطوات السّابقة، تظهر الشاشة أدناه للبدء بكتابة الجمل البرمجيّة للصّنف (clsStudent).



بشكلٍ عام، يكون الصنف على الشّكل الآتي:

```
<Access Modifier> Class <Class_Name>
    <Attributes>
    < Methods >
    <Constructor>
End Class
```

ملحوظة:

تسمية اسم الصّنف واسم الدّالة تخضع لشروط تسمية المتغيّرات عينها.

محدّدات الوصول (Access Modifiers)

يقصد بها الكلمات المحجوزة (Keywords) والتي تحدّد إمكانيّة الوصول للوحدات داخل الصّنف، نذكر منها:

Keywords	المفهوم	أمثلة
Private	هي تلك العناصر (**Members**) التي يمكن الوصول لها من داخل الصنف فقط.	**Private** DOB as Date **Private** function sum () as Double
Public	هي تلك العناصر (**Members**) التي يمكن الوصول لها من **أيّ مكان** بالمشروع.	**Public** avg as Double **Public** Sub binary_tree()
Protected	هي تلك العناصر (**Members**) التي يمكن الوصول لها من الأصناف الوارثة (**Inherits**) فقط.	**Protected** Password as String **Protected** Sub Print_Info()

خصائص الصنف

بعد القيام بإنشاء صنف جديد (clsStudent)، لا بدّ من إضافة الخصائص على مستوى الصّنف كما في الشّكل الآتي:

```
Public Class clsStudent                    'اسم صنف الأب
#Region "Attributes"                               'منطقة الصفات
    Public ID As Integer                           'رقم الطالب
    Protected Name As String                       'اسم الطالب
    Protected Avg As Double                        'للمعدل العام
    Protected result As String                     'للتقدير
    Protected BirthDate As Date                    'تاريخ ميلاد الطالب
    Private AllSubjectName() As String = {
"البرمجة و الأتمتة", "الرياضيات", "اللغة العربية", "اللغة الانجليزية",
"الاتصالات و الالكترونيات", "التربية الدينية", "الثقافة العلمية",
"الادارة و الريادة"}
    Private Schoolmarks(7) As Integer
#End Region
```

ملحوظة:

#Region: هي أحد أنواع الموجّهات (Directive) التي تجعل منطقة محدّدة من البرمجة قابلة للطيّ، وتظهر أهميتها في تقسيم برنامجك إلى مناطق من أجل عملية التنظيم فقط.

نشاط (٥) إضافة خصائص إلى الصنف clsTechnology.

قم بإضافة الخصائص أدناه إلى صنف الابن (clsTechnology)

```
Public Class clsTechnology                 ' اسم صنف الابن
    Inherits clsStudent                    'clsStudent تحقيق الوراثة من صنف الأب
#Region "Attributes"                               ' منطقة الاعلان عن الصفات
    Private sum1 As Integer = 0                    ' مجموع علامات المباحث الاجبارية
    Private sum2 As Integer = 0                    ' مجموع علامات المباحث الأساسية
    Private avg1 As Double = 0.0                   ' معدل علامات المباحث الاجبارية
    Private avg2 As Double = 0.0                   ' معدل علامات المباحث الأساسية
    Private MCCourse() As String = {"البرمجة و الأتمتة", "الرياضيات",
                                    "اللغة العربية", "اللغة الانجليزية"}
    ' MCCourse: MinistryCompulsaryCourse
    Private MBCourse() As String = {"الاتصالات و الالكترونيات", "التربية الدينية",
                                    "الثقافة العلمية", "الادارة و الريادة"}
    'MBCourse: Ministry Basic Course
    Private MCMarks() As Integer           'MCMarks:  Ministry Compulsary Marks
    Private MBMarks() As Integer           'MBMarks: Ministry Basic Marks
#End Region
```

الدّالة (Property).

تعرّف الدّالة "Property" بمحدّد وصول عام (Public)، وهي إحدى الآليّات غير المباشرة والمستخدمة للوصول إلى خصائص الأصناف ذات محدّد وصول "Private" أو "Protected" لقراءتها أو إسناد قيمة لها، وتحوي على نوعين من الدّوال يطلق عليهما اسم (Accessors) هما:

- Get: لإسترجاع قيمة البيانات وقراءتها، وهي بذلك تكون غير قادرة على تغيير القيم.

- Set: لإسناد قيم محدّدة إلى البيانات أو تغييرها حسب الحاجة، وهي بذلك تكون غير قادرة على إظهار البيانات.

ملاحظة:

- يستخدم المعرّف Readonly للدّالة Get فقط داخل Property.
- يستخدم المعرّف Writeonly للدّالة Set فقط داخل Property.

لإضافة Property إلى الصنف (clsStudent) في مشروعك، اكتب الجمل البرمجيّة الآتية:

```
#Region "Property"
    Public Property _ID As Integer
        Get
            Return ID
        End Get
        Set(value As Integer)
            ID = value
        End Set
    End Property

    Public Property _Name As Strin
        Get
            Return Name
        End Get
        Set(value As String)
            Name = value
        End Set
    End Property
```

```
Public Property _Schoolmarks As Integer()
    Get
        Return Schoolmarks
    End Get
    Set(value As Integer())
        Schoolmarks = value
    End Set
End Property
Public ReadOnly Property _Avg As Double
    Get
        Return Avg
    End Get
End Property
Public ReadOnly Property _result As String
    Get
        Return result
    End Get
End Property
```

```
Public Property _BirthDate As Date
    Get
        Return BirthDate
    End Get
    Set(value As Date)
        BirthDate = value
    End Set
End Property

Public Property _AllSubjectName As String()
    Get
        Return AllSubjectName
    End Get
    Set(value As String())
        AllSubjectName = value
    End Set
End Property
```

نشاط (٦) إضافة خصائص إلى الصنف clsTechnology.

- أضف "Property" مناسبة لخصائص الصّنف (clsTechnology).

الـدّوال (Methoeds)

تعلّمت في صفّ سابقٍ كيفيّة كتابة الدّوال والإجراءات المختلفة برمجيّا، والتي يُطلق عليها في البرمجة الهدفية مسمّى (Methoeds)؛ فهي تُعدّ وحداتٍ برمجيّة ذات مهمّة محدّدة.
في مشروعنا، فإنّ الدّوال اللّازمة لبنائها في صنف (clsStudent) كالآتي:

1. دالّة إدخال العلامات من خلال صندوق الإدخال (InputBox) إلى المصفوفة (Schoolmarks()) من نوع أحاديّة تتسّع لـ (8) علامات والتي تساوي عدد المباحث، انظر الشكل أدناه.

```
#Region "Methods"                          'منطقة الدوال
    Public Sub insert_Subject_Mark()  'دالة ادخال علامات لطالب من خلال صندوق الادخال
        For i = 0 To (AllSubjectName.Length - 1)
            Schoolmarks(i) = InputBox("Enter the Mark of " + " " + AllSubjectName(i))
        Next
    End Sub
```

2. دالّة حساب المعدّل Calculate_average(): حيث سيتمّ حساب المعدّل للعلامات المُدخلة في المصفوفة (SCHOOLMARKS) بجمع العلامات وقسمتها على 9، وسيظهر المعدل في حال كان الطالب قد نجح في جميع المباحث، أمّا إذا لم ينجح في إحدى المباحث فإنّ المجموع سيظهر بالقيمة (99-).

```
Public Overridable Sub Calculate_average()   'دالة حساب المعدل العام لطالب
    Dim sum As Integer                       'لحساب مجموع العلامات
    Dim count As Integer = 0                 'لعدّ المواد التي نجح بها الطالب
    If Schoolmarks(0) >= 100 Then  'فحص نجاح الطالب بمادة البرمجة في المدرسة
        count += 1
    End If
    For i = 1 To AllSubjectName.Length - 1 'فحص نجاح الطالب ببقية المواد بالمدرسة
        If Schoolmarks(i) >= 50 Then
            count += 1
        End If
    Next
    If count = AllSubjectName.Length Then
        For i = 0 To (AllSubjectName.Length - 1)
            sum += Schoolmarks(i)
        Next
        Avg = sum / (AllSubjectName.Length + 1.0) ' معدل الطالب في حال نجاحه
    Else                                          'بكل المواد
        Avg = -99.0  ' قيمة افتراضية تعبّر عن حالة رسوب الطالب ببعض أو كل المواد
    End If
End Sub
```

3. دالّة حساب النّتيجة السّنوية، حيث يكون النّاتج (ناجح) في حال نجح الطالب في جميع المباحث، أو (راسب) في حال أخفق في إحدى المباحث أو جميعها. انظر الشكل أدناه.

```
Public Overridable Sub Calculate_result()        'دالة حساب تقدير الطالب النهائية
    If Avg <> -99.0 Then
        result = "ناجح"
    Else
        result = "راسب"
    End If
End Sub
#End Region
```

نشاط (٧) الدوال في صنف (clsTechnology)

أكتب دوالّ الصنف (clsTechnology) في مكانها المخصّص لها، كما يأتي:

- دالّة حساب مجموع ومعدّل علامات المباحث "الإجباريّة" للفرع كما يأتي:

```
Public Sub averageCompulsary()    ' دالة حساب مجموع ومعدل المباحث الاجبارية
    Dim count1 As Integer = 0     ' متغير عداد
    If MCMarks(0) >= 100 Then     ' فحص حالة نجاح الطالب بمادة البرمجة/٢٠٠ علامة
        count1 += 1
        sum1 += MCMarks(0)        'sum1 اضافة العلامة إلى متغير المجموع
    End If
    For i = 1 To (MCMarks.Length - 1)  ' فحص نجاح الطالب ببقية المواد الاجبارية
        If MCMarks(i) >= 50 Then
            count1 += 1
            sum1 += MCMarks(i)        'sum1 اضافة العلامة إلى متغير المجموع
        End If
    Next
    If count1 = 4 Then                ' حساب معدل علامات الطالب بالمواد الاجبارية
        avg1 = sum1 / 5.0
    Else
        sum1 = -99                    ' يعطى المجموع والمعدل القيمة -٩٩ في حال رسوبه بإحدى المواد
        avg1 = -99
    End If
End Sub
```

- دالّة حساب مجموع ومعدّل علامات المباحث "الأساسيّة" مع تحقيق الشّروط السابقة في الوصف للفرع المقترح كما يأتي:

```
Public Sub averageBasic()
    Dim count2 As Integer = 0
    Dim HighestMark As Integer = 0

    For i = 0 To (MCMarks.Length - 1)    'فحص نجاح الطالب بالمواد الاساسية
        If MBMarks(i) >= 50 Then
            count2 += 1
        End If
    Next
    If count2 = 4 Then
        HighestMark = max(MBMarks(1), MBMarks(2), MBMarks(3))  'دالة لمعرفة أعلى قيمة بين قيم ٣ مواد
        sum2 = MBMarks(0) + HighestMark  'مجموع علامة مادة الاتصالات وأعلى مادة بين المواد الثلاث المتبقية
        avg2 = sum2 / 2.0                'حساب معدل الطالب في مادتين اساسيتين
    Else
        sum2 = -99
        avg2 = -99
    End If
End Sub
```

- اكتب دالّة فحص أكبر قيمة بين ثلاث قيم وإرجاع القيمة الأكبر كما يأتي:

Public Function max(Integer, Integer, Integer) As Integer

- اكتب دالّة حساب المعدّل العام Overrides Sub Calculate_average() بحيث إذا كانت معدّل علامات المباحث "الإجبارية" ومعدّل علامات المباحث "الأساسيّة" لا تساوي (99-) فإنّه يحسب قيمة المعدّل العام (Avg).

- اكتب دالّة حساب النتيجة السنوية Overrides Sub Calculate_result() بناءً على قيمة المعدّل العام (Avg).

دالّة البَنّاء (Constructor)

يُمثَّل البَنّاء برمجيّا دالّة تحتوي على الكلمتين المحجوزتين (Sub New) داخل الصنف، وما يميّزه هو عدم وجود اسم له في جملة الإعلان عن الدّالة، ويهدف إلى اسناد قيم أوليّة لصفات الكائن.

تسمح لغة (VB.NET) بخاصّية (Overloading) للبنّاء دون الحاجة لاستخدام كلمة (Overloads)، حيث تختلف البنّاءات داخل الصّنف الواحد باختلاف عدد المعاملات التي تمرّر لها من الكائنات.

ملاحظة:

في حال لم يتمّ إنشاء بنّاء داخل الصنف، فإنّ المترجم سيقوم بإنشاء بنّاءٍ افتراضي في زمن التنفيذ للبرنامج.

مثال

إضافة بنّاءات الى صنف (clsStudent).

الحل: تمثّل الجمل البرمجيّة أدناه مثالاً على دوال البنّاءات، أَضِفها إلى مشروعك.

```
#Region "Constructors"                    'منطقة البناءات الخاصة بالصنف
    Public Sub New()
        ID = ""
        Name = ""
        BirthDate = ""
    End Sub
    Public Sub New(_ID As Integer, _Name As String, _BirthDate As Date)
        ID = _ID
        Name = _Name
        BirthDate = _BirthDate
    End Sub
#End Region
```

نشاط (٨) دالة البنّاء للصنف (clsTechnology)

تمعّن البنّاء أدناه للصنف (clsTechnology)، ثمّ أجب عن الأسئلة التي تليه:

```
#Region "Constructor"                     'منطقة البناءات الخاصة بالصنف
    Public Sub New(_ID As Integer, _Name As String, _BirthDate As Date, tempArray1() As Integer,
                   tempArray2() As Integer)
        MyBase.New(_ID, _Name, _BirthDate)
        MCMarks = tempArray1              ' اسناد قيم مصفوفة لمصفوفة أخرى
        MBMarks = tempArray2              ' اسناد قيم مصفوفة لمصفوفة أخرى
    End Sub
#End Region
```

- ما الوظيفة التي تحقّقت من الجملة البرمجيّة: (MyBase.New(_ID, _Name, _BirthDate))؟
- ماذا تعني الجملة البرمجيّة: MCMarks = tempArray1؟

الوراثة

يمكن توضيح ذلك من خلال إيجاد العلاقة بين الصّنفَين: (clsStudent) و (clsTechnology)، حيث تكون العلاقة: (clsTechnology is A clsStudent) وهذا واضح من مخطّط UML سابقاً.

ولتحقيق ذلك برمجيّا، نقوم بإضافة كلمة إلى الصّنف (clsTechnology) كما في الشّكل الآتي:

```
Public Class clsTechnology                    ' اسم صنف الابن
    Inherits clsStudent                       'تحقيق الوراثة من صنف الأب clsStudent
```

كما تُحدَّد الصفات والدّوال الموروثة وغير الموروثة برمجيّا من خلال المحدّدات برمجيّة التي درستها سابقاً، والجدول أدناه يبيّن إمكانيّة استخدام تلك المحددّات في الأصناف كالآتي:

المحدّدات (Modifiers)	البيانات والدّوال		
	Private	Public	Protected
صنف الاب (Base Class)	نعم	نعم	نعم
صنف الابن (Derived Class)	لا	نعم	نعم
أصناف غير وارثة (Outside Classes)	لا	نعم	لا

تعدّد الأشكال

يتمّ من خلال آليّات Overloading وOverriding بعد تطبيق مبدأ الوراثة التي تجعل منه أمراً ممكناً، ولتحقيق ذلك برمجيّا، أَضِف الجمل البرمجيّة الآتية إلى مشروعك:

* آليّة الهيمنة Overriding:

√ في صنف (clsTechnology):

```
Public Overrides Sub Calculate_average()      'دالة حساب المعدل العام للطالب
    If avg1 <> -99 And avg2 <> -99 Then
        Avg = (sum1 + sum2) / 7.0             ' حساب المعدل العام لطالب التكنولوجي
    Else
        Avg = -99                             ' في حالة رسوب الطالب بإحدى المواد
    End If
End Sub
```

√ في صنف (clsStudent): يتم إضافة الكلمة المحجوزة (Overridable) إلى الدّوال كما يأتي:

```
Public Overridable Sub Calculate_average()
Public Overridable Sub Calculate_result()
```

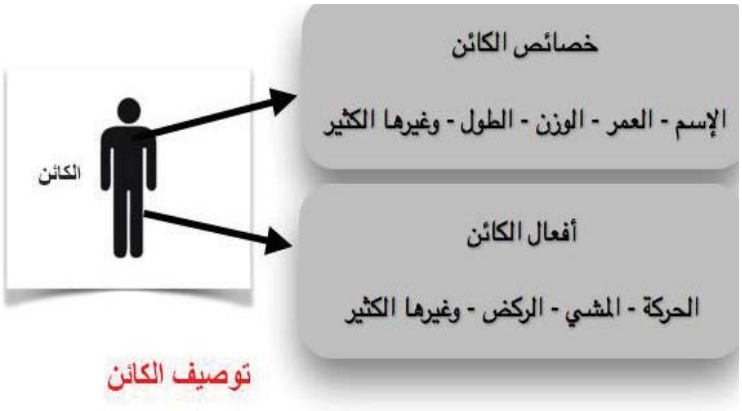
* آليّة Overloading: الظاهرة في مفهوم البنّاءات، انظر لبرمجتها في صنف (clsStudent) السابق، ثمّ لاحظ عدد المعاملات ونوعها.

ولتحقّق الأصناف الفائدة المرجوّة منها، لا بدّ من المرور على مفهوم الكائن (Object).

مفهوم الكائن (Object)

كلّ شيء حولنا عبارة عن كائنات!

وتعتبر لغات البرمجة الهدفية الأشياء من حولنا باعتبارها كائنات؛ فسيّارة المرسيدس كائن من صنف السّيارة، وأنت كائن من صنف انسان، وشجرة الزّيتون كائن من صنف الأشجار... أعط أمثلة أخرى؟

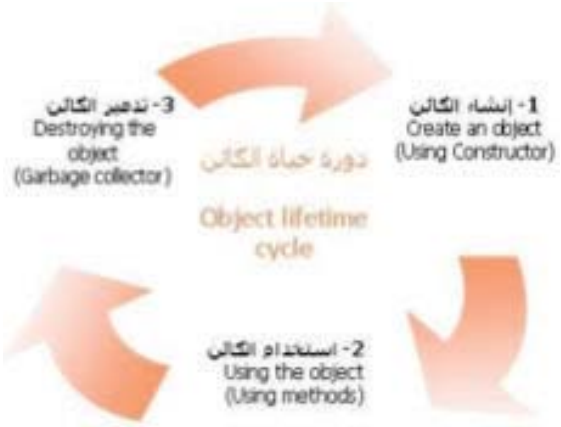


تمثّل الكائنات نسخة (Instance) من الصّنف الممثّل له، فلو أخذنا الإنسان مثالاً؛ لوجدنا أنّ من خصائصه (الاسم، العمر، الوزن، الطول...)، أعط خصائص أخرى؟ ومن سلوكيّاته (الحركة، المشي، الركض...) أعط سلوكيّات أخرى؟ وإذا ما أردنا إضفاء هويّة خاصة بذلك الكائن نسند لخصائصه قيماً محدّدة (مثال: العمر = 45، الاسم = "آدم")، فالكائن يمثّل الجانب التّنفيذي للصّنف.

نشاط (9) الكائنات من حولنا

تقوم كل مجموعة من الطلبة بتحديد كائن ما من بيئة الصف، وتحدّد كل مجموعة خصائص وسلوكيات الكائن وإسناد قيم لخصائصه من خلال جدول، ثمّ يقوم طالب من المجموعة بعرض ما تمّ التّوصل إليه ومناقشته، ويدوّن على السبورة.

الكائن (Objects) برمجيّا



يمرّ الكائن بمراحل دورة حياة هي:

1 **مرحلة إنشاء الكائن (Object).**

أ- الإعلان (Declaration):

لا يتمّ استخدام البنّاءات، كما لا يعطى الكائن هويّة خاصة به، ويتمّ ذلك من خلال الصيغة البرمجيّة الآتية:

```
Public Cow As Animal
```

```
Public BA As BankAccount
```

ب- التهيئة (Initialization): تأتي مباشرة بعد عمليّة الإعلان، حيث يصبح الكائن جاهزاً للاستخدام (تنفيذ الخصائص والسّلوكيّات الخاصّة بالصّنف الّذي يمثّله)، ويعطى من خلال (البنّاء) هويّة افتراضيّة تتغيّر خلال تنفيذ البرنامج كما هو أدناه:

```
BA = New BankAccount (10)
```

```
Cow = New Animal ()
```

ويمكن دمج عمليّة الإعلان والتّهيئة في جملة برمجيّة واحدة مع تمرير المعاملات اللازمة كما يأتي:

```
Public BA as BankAccount = New BankAccount(10)
```

نشاط (٩) إنشاء كائن في مشروعك

من خلال نافذة البرمجة للنموذج في مشروعك، في منطقة التعريفات العامّة، أعلن عن كائن (وليكن Stu) من نوع (clsStudent)، وكائن آخر (وليكن StuTech) من نوع (clsTechnology).

2 **استخدام الكائن:** بعد عمليّة إنشاء الكائن يصبح جاهزاً لتنفيذ متطلّبات البرنامج من خلال الوصول إلى خصائص ودوال الصّنف الذّي يمثّله بناء على محدّدات وصولها، ومثال ذلك

```
stu.insert_Subject_Mark()
horse.legs = 4
```

نشاط (١٠) برمجة أزرار النموذج في مشروعك

أ. برمجة زرّ الأمر "إضافة طالب" في النّموذج، والعرض في أداة (DataGridView) من خلال كتابة الجمل البرمجيّة الآتية:

```
Private Sub btn_insertStudent_Click(sender As Object, e As EventArgs) Handles btn_insertStudent.Click
    stu = New clsStudent(txt_ID.Text, txt_Name.Text, DateTimePicker1.Text)

    stu.insert_Subject_Mark()
    stu.Calculate_average()
    stu.Calculate_result()

    DataGridView1.Rows.Add(txt_ID.Text, txt_Name.Text, DateTimePicker1.Text, " ", " ", stu._Avg, stu._result)
End Sub
```

ب. برمجة زرّ الأمر "إضافة طالب تكنولوجي" في النموذج مع العرض على أداة (DataGridView) من خلال كتابة الجمل البرمجيّة الآتية:

```
Private Sub btn_insertTechStudent_Click(sender As Object, e As EventArgs) Handles btn_insertTechStudent.Click
    tempArray1 = {Val(txt_Mark1.Text), Val(txt_Mark2.Text), Val(txt_Mark3.Text), Val(txt_Mark4.Text)}
    tempArray2 = {Val(txt_Mark5.Text), Val(txt_Mark6.Text), Val(txt_Mark7.Text), Val(txt_Mark8.Text)}

    stuTech = New clsTechnology(txt_ID.Text, txt_Name.Text, DateTimePicker1.Text, tempArray1, tempArray2)

    stuTech.averageCompulsary()
    stuTech.averageBasic()
    stuTech.Calculate_average()
    stuTech.Calculate_result()

    DataGridView1.Rows.Add(txt_ID.Text, txt_Name.Text, DateTimePicker1.Text, stuTech._Avg1, stuTech._Avg2,
                           stuTech._Avg, stuTech._result)
End Sub
```

ج. برمجة زرّ الأمر "الخروج" في النموذج من خلال كتابة الجمل البرمجيّة الآتية:

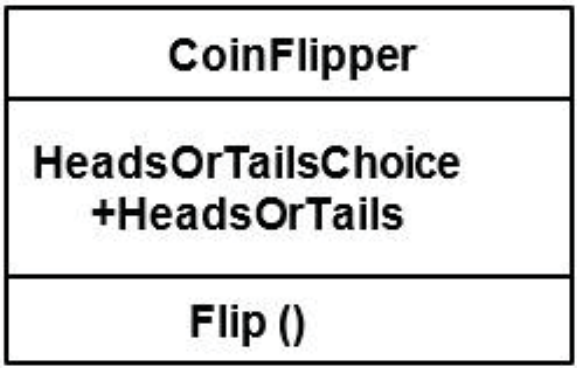
```
Private Sub btn_Exit_Click(sender As Object, e As EventArgs) Handles btn_Exit.Click
    End
End Sub
```

3 **تدمير الكائن (Destruction):** التّخلص من الكائن وتحرير المصادر المرتبطة معه (طابعة، حجز ذاكرة، استخدام كرت الشّاشة..) عن طريق جامع القمامة (Garbage Collector (GC))، من خلال استخدام الجملة البرمجيّة (ObjName = Nothing) لتفعيلها، أو من خلال استخدام الجمل البرمجيّة (Finalize) أو (Dispose).

1. ما المقصود بالمصطلحات الآتية: البرمجة الهدفية، تعدّد الأشكال، الصنف، الكائن، الوراثة.
2. قارن بين خصائص النّمط البرمجي الإجرائي والكائنيّ من حيث: النّهج، مستوى الأمان، مثال.
3. ما أهمية استخدام الخاصّية Property في البرمجة.
4. عدّد اثنين من الفروق بين: (Overloading) و(Overriding) في البرمجة.
5. اكتب جملاً برمجيّة مناسبة لوصف صنف (Time) يحوي الآتي:
- الصفات: Hour، Minute، Second من نوع Private.
- الدوال: Get_Time ()، Set_Time () مع معاملات مناسبة لكلّ منها.
- استخدام آليّة Overloading مناسبة.
- بنّاء (Constructor) مناسب.

ثمّ مثّل الصّنف أعلاه من خلال نموذج UML مناسب.

6. بالاعتماد على مفاهيم البرمجة الهدفية، اكتب برنامجاً لصنف (CoinFlipper) مستعيناً بالرسم أدناه، بحيث يظهر النّاتج ("Tail") أو ("Head") عشوائيّا عند الضّغط على زرّ أمر (Button)، وبتصميم مناسب.



7. قم بإجراء تعديلات مناسبة على مستوى التصميم والبرمجة لمشروع الفرع التكنولوجي المقترح، بحيث يحسب المعدل العام للطالب في الفرع التكنولوجي في المواد الأساسية الأربعة و أعلى علامة بين كل مادتين من المواد الآتية: - (الاتصالات، التربية الدينية). - الثقافة العلمية، الادارة والريادة.

أسئلة الوحدة

س١ ضع دائرة حول رمز الإجابة الصّحيحة لكل جملة من الجمل الآتية:

١. ما المصطلح الذي يطلق على آليّة الإضافة والحذف إلى مكدّسة؟

أ. FIFO ب. LIFO ج. FINO د. LILO

٢. ما المصطلح الذي يطلق على آليّة الإضافة والحذف إلى طابور؟

أ. FIFO ب. LIFO ج. FINO د. SIMO

٣. ماذا يطلق على عملية الحذف من مكدّسة لدى المبرمجين؟

أ. Delete ب. Insert ج. POP د. ADD

٤. ماذا يطلق على عمليّة الإضافة لطابور لدى المبرمجين؟

أ. PUSH ب. Insert ج. POP د. Delete

٥. ما قيمة المؤشر Rear عند الإعلان عنه بالبرنامج؟

أ. -1 ب. 0 ج. N-1 د. N

٦. ما قيمة المؤشّر Front عند إضافة 3 عناصر إلى طابور وحذف عنصر؟

أ. -1 ب. 0 ج. 1 د. 2

٧. ما قيمة المؤشّر Top عند ظهور الرّسالة "لا يمكن الاضافة" في حالة المكدّسة؟

أ. N-1 ب. N ج. -1 د. 1

٨. أيّ الآتية مثالٌ على "غلاف الهاتف النقال" في البرمجة الهدفيّة؟

أ. التّجريد ب. التّغليف ج. التّوارث د. تعدّد الأشكال

٩. أيّ الآتية تطبيق لمفهوم البنّاءات "Constructor" في نفس الصّنف؟

أ. التّوارث ب. Overriding ج. Overloading د. Public

س٢ ضع إشارة (✓) أمام العبارة الصحيحة، أو إشارة (✗) أمام العبارة الخاطئة لكلّ ممّا يأتي:

( ) أ. السّجلات مثال على تراكيب البيانات الخطية.

( ) ب. يمكن الحذف والإضافة إلى مكدّسة من طرفيها.

( ) ج. التّعبير (3*6)-4) خاطئ لأنّ المكدسة فارغة وما زال هناك قوس إخراج.

( ) د. عندما يكون Front = Rear، يكون عدد العناصر في المكدّسة يساوي 1.

( ) هـ. للانتقال من آخر موقع إلى أول موقع بالطابور الدائري نستخدم العبارة (Rear Mod N).

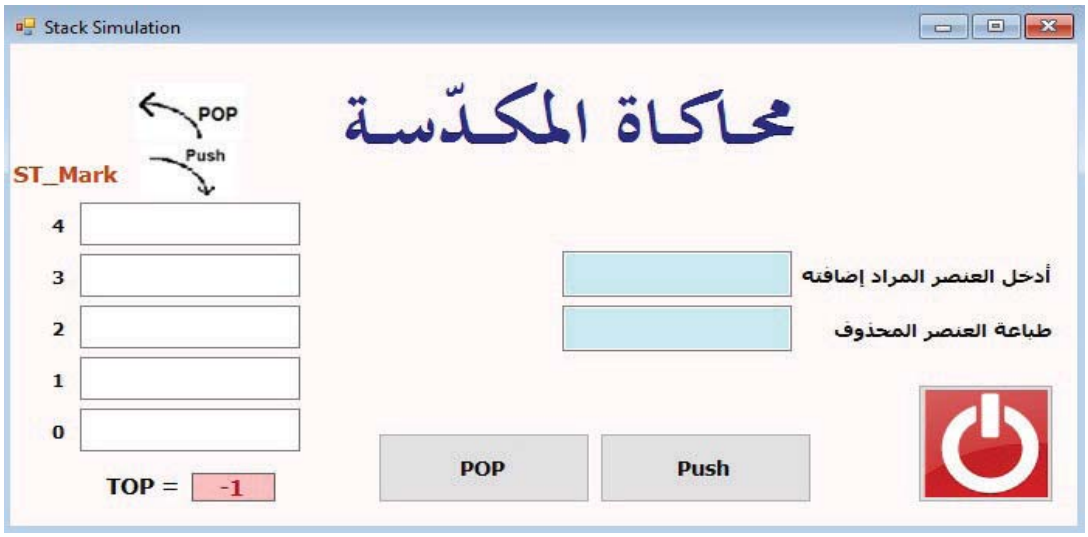
( ) و. يعتبر جهاز الحاسب كائنا مكوّنا من عدّة كائنات.

( ) ز. تشغيل التلفاز يعتبر مثالا على الوراثة.

( ) ح. عندما يرث صنف ما صنفا آخر فإنّه يرث الخصائص فقط.

س٣ قم بإنشاء مشروع جديد في بيئة Visual Studio، ثمّ صمّم الواجهة أدناه، مع كتابة الجمل البرمجيّة المناسبة بلغة VB.NET لمحاكاة آليّة عمل المكدّسة عند إجراء عمليّات الحذف والإضافة لبيانات عدديّة، مراعياً ما يأتي:

- أ. عند كتابة العنصر المراد إضافته في صندوق النص ثمّ الضّغط على زر الأمر "Push"، يقوم بإضافته إلى المكدّسة (ST_MARK) وإظهار صورة "سهم الإضافة" المبيّن أعلاه، يرافقه تغيير قيمة المتغيّر "TOP" وتغيير لون خلفيّة الموقع (Backcolor) المضاف إليه العنصر إلى اللّون الأصفر (YELLOW).
- ب. عند امتلاء المكدّسة والقيام بعمليّة الإضافة يظهر رسالة (MessageBox) تفيد بعدم امكانيّة الإضافة.
- ج. عند الضغط على زر الأمر "POP"، يُطبع العنصر المحذوف بصندوق النصّ، مع تغيير قيمة "TOP"، يرافقه تغيير لون الخلفيّة لموقع العنصر المحذوف إلى اللّون الأبيض (WHITE)، وإسناد القيمة False إلى الخاصيّة Enabled لصندوق النّص المحتوي على العنصر المحذوف دون حذف العنصر داخله.
- د. عند القيام بعمليّة الحذف لمكدّسة فارغة، يظهر رسالة (MessageBox) تفيد بعدم إمكانيّة الحذف.
- هـ. إضافة زر أمر(Exit) للخروج من البرنامج.



- و. قم بإجراء التّعديلات المناسبة على مستوى التّصميم والبرمجة لحساب المجموع والمعدّل للعناصر العدديّة المحذوفة في المكدّسة.

س٤ قم بإضافة نموذجاً جديداً للمشروع وتصميمه كما في الشّكل أدناه، مراعياً ما يأتي:

- أ. إضافة أزرار انتقال بين النموذجين في المشروع على مستوى التصميم والبرمجة.
- ب. عند كتابة العنصر المراد إضافته في صندوق النّص ثمّ الضّغط على زر الأمر "Insert"، يقوم بإضافته إلى الطابور (Q_MARK) المبيّن أعلاه، يرافقه تغيير قيمة المتغيّر "Rear" وتغيير لون خلفيّة

الموقع (Backcolor) المضاف إليه العنصر إلى اللّون الأصفر (YELLOW).

- ب. عند امتلاء الطابور والقيام بعمليّة الإضافة يظهر رسالة (MessageBox) تفيد بعدم إمكانيّة الإضافة.
- ج. عند الضغط على زر الأمر "Delete"، يُطبع العنصر المحذوف بصندوق النصّ، مع تغيير قيمة "Front"، يرافقه تغيير لون الخلفيّة لموقع العنصر المحذوف إلى اللّون الأبيض(WHITE)، وإسناد القيمة False إلى الخاصيّة Enabled لصندوق النّص المحتوي على العنصر المحذوف دون حذف العنصر داخله.
- د. عند القيام بعمليّة الحذف لطابور فارغ، يظهر رسالة (MessageBox) تفيد بعدم إمكانيّة الحذف.
- هـ. إضافة زر أمر(Exit) للخروج من البرنامج.

س٥ باستخدام مفاهيم البرمجة الهدفية، قم بتصميم وكتابة برنامج بلغة (VB.NET) لفحص ما إذا كانت السّلسلة النّصية المدخلة متناظرة (Palindrome) أم لا.

(ملاحظة: كلمة (DAD) متناظرة، بينما كلمة (Sami) غير متناظرة).

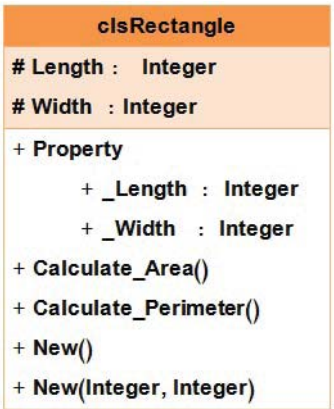
س٦ تمعّن المخطّط المجاور، ثمّ أجب عن الأسئلة الآتية:

أ. اكتب الجمل البرمجيّة المناسبة للصنف.

ب. إذا علمت أنّ صنف المربّع (clsSquare) يرث من صنف المستطيل (clsRectangle)، أنشئ صنف المربع برمجيّا وأضفه إلى البرمجة السابقة محقّقا مبادئ الوراثة وتعدّد الأشكال.(ملاحظة: أضف صفات وعمليّات مناسبة).

ج. صمّم واجهة استخدام (Interface) مناسبة.

د. اعرض مساحة ومحيط المربّع والمستطيل على أدوات مناسبة في التصميم.



الوحدة الثالثة

الذكاء الاصطناعي

أتأمل ثم أناقش: الذكاء الاصطناعي ... هل يصبح بديلاً للإنسان

يتوقع من الطلبة بعد دراسة هذه الوحدة ان يكونوا قادرين على معرفة تكنولوجيا الذكاء الاصطناعي من خلال الآتية:

1- التعرف الى مفهوم الذكاء الاصطناعي ومجالاته.

2- التميز بين البرامج التقليدية وبرمج الذكاء الاصطناعي.

3- التعرف الى واقع ومستقبل الذكاء الاصطناعي.

الدرس (١)

الذكاء الاصطناعي

مفهوم الذكاء الاصطناعي

ظهر علم الذكاء الاصطناعي في خمسينات القرن الماضي، وبناءً على الاكتشافات الحديثة في علم الدماغ والشبكة العصبية، إضافة للتقدم الحاصل في علم البرمجيّات، والتمكّن من زيادة القدرة التحليلية للمعالجات الدقيقة في تحليل معلومات هائلة وتخزينها بشكل متوازٍ، واستخدام نظريات رياضية مبنية على أساس المنطق الرياضي؛ أدى إلى تبني العلماء نهجاً جديداً في إنتاج آلات ذكية تحاكي الذكاء البشري خاصة في قدرتها على اتخاذ القرارات وحلّ المشكلات.

الذكاء الاصطناعي: علم من علوم الحاسوب، يختص بتصميم وتمثيل وبرمجة نماذج حاسوبية في مجالات الحياة المختلفة، تحاكي في عملها طريقة تفكير الانسان وردودHافعاله في مواقف معينة

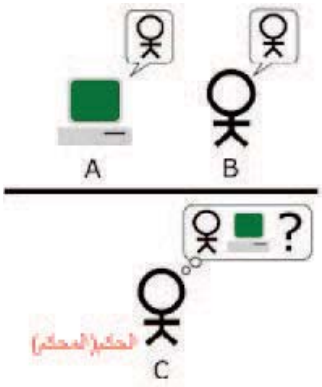
وتتلخّص دوافع هذا العلم في فهم ومحاكاة بعض عناصر الذكاء البشري أو حتى جميعها ضمن المحاور الرئيسية الآتية:

1 **التّفكير:** القدرة على التعلم كقدرة الطفل على تمييز صوت أمه بدون أية قواعد منطقية.

2 **القياس:** القدرة على الاستدلال المنطقي كوجود شكل دون اضلاع فهو دائرة.

3 **التصرّف المنطقي:** فمثلا اذا كان الجو ماطرا فيجب استخدام المظلة.

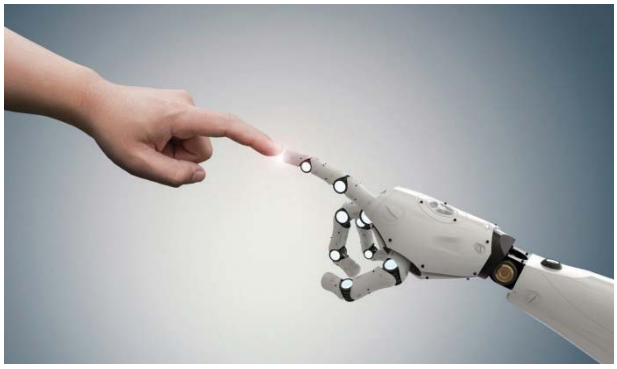
يُعدّ (اختبار تورِنغ) أحد التطبيقات الأولى في مجال الذكاء الاصطناعي، الذي صممه العالم البريطاني أَلَن تورِنغ (Alan Turing) لتعريف الآلة الذكية التي تحدد ما إذا كان البرنامج المحوسب



قادراً على التفكير والتصرف مثل الإنسان أم لا. والاختبار عبارة عن لعبة تتكون من مجموعة اسئلة تُوجه من قبل (الحَكم أو المُحكِم) بصيغة معينة خلال فترة زمنية محددة إلى شخص وحاسب آلي (اللاعبون) متواجدين في غرفة أخرى، وبعد أن يحصل السائل على الإجابات يحاول أن يحدد مصدر كل إجابة من اي لاعب (شخص ام حاسب)؛ فإذا لم يتمكن من تحديد المصدر، يصح حينها وصف هذا الحاسب بالذكاء لمحاكاته طريقة تفكير الإنسان وأعطى نفس إجاباته.

هل يصل الذكاء الاصطناعي إلى مستوى الذكاء البشري؟

نشاط (١) محاكاة ذكاء الآلة للإنسان



حدّد محور الذكاء المناسب لكلّ موقف/عبارة من الآتية:

1- شكل له اربعة اضلاع متساوية وزواياه قوائم.
2- اذا اقتربت من حائط استدر الى جهة اليمين.
3- اذا لمست جسم ساخن ابتعد عنه.
4- التعرف الى لغة المتكلم.

أهداف الذكاء الاصطناعي

1. إنشاء أنظمة خبيرة: الأنظمة التي تظهر سلوكًا ذكيًا ولديها القدرة على التعلم، وتقديم المشورة والمساعدة لمستخدميها.
2. تحقيق الذكاء البشري في الآلات من خلال إنشاء أنظمة تفهم وتفكر وتتعلم وتتصرف كما البشر.
3. القدرة على معالجة المعلومات بشكل متوازٍ باستخدام البرمجة، وهي الطريقة الاقرب لتفكير الانسان في حلّ المشكلات.

البرمجة التقليدية وبرمجة الذكاء الاصطناعي

تعتمد البرمجة التقليدية على خوارزميات ذات خطوات واضحة ومحددة مثل برنامج إيجاد محيط الدائرة، بينما برمجة الذكاء الاصطناعي تعتمد على مجموعة طرق الحل الممكنة بناءً على معرفة مسبقة للوصول الى الهدف المنشود، وذلك من التفكير والتطبيق والاستنتاج، والفعل الذاتي دون إدراك برمجي من المبرمج مثل مشكلة البائع المتجول حيث يريد زيارة عدة مدن واختيار اقصر طريق يجعله يتمم الزيارة بنجاح، يمكن حلها بالطرق التقليدية الا انها تأخذ وقتا طويلا ولن تخرج بالحل الامثل، هناك عدة لغات تهتم بإنتاج الذكاء الاصطناعي منها lisp، matlap، python، prolog وغيرها.

والجدول ادناه يبين الفرق بين البرمجة التقليدية وبرمجة الذكاء الاصطناعي.

البرمجة التقليدية	برمجة الذكاء الاصطناعي
- امكانية البرنامج الحاسوبي الإجابة عن الأسئلة المحددة التي يهدف إلى حلها. - التعديل في البرنامج يؤدي إلى تغيير في هيكله. - التعديل ليس سريعًا وسهلاً. قد يؤدي إلى التأثير سلبا على البرنامج.	- امكانية البرنامج الحاسوبي الإجابة عن الأسئلة العامة التي يُقصد حلها. - التعديل في البرنامج لا يؤدي إلى تغيير في هيكله. - التعديل سريع وسهل.

مراحل تطور الذكاء الاصطناعي

1- المرحلة الأولى (1950 – 1963):

نشأت فور انتهاء الحرب العالمية الثانية، وقد بدأها العالم شانون عام 1950 ببحثه في لعبة الشطرنج وانتهت بالعالم فيجن عام 1963 وتميزت هذه المرحلة بإيجاد حلول للألعاب وفك الألغاز باستخدام الحاسب والتي اعتمدت فكرة تطوير طرق البحث عن الحلول وأدت إلى تطوير واستحداث النماذج الحسابية.

2- المرحلة الثانية (1965 – 1975):

يطلق عليها بـ (المرحلة الشاعرية)، والتي بدأت في منتصف الستينات إلى منتصف السبعينات، حيث قام العالم مِنسكي بعمل الإطارات أو النماذج لتمثيل المعلومات، ووضع العالم (وانجراد) نظام آلي لفهم اللغة الانجليزية في القصص والمحادثات، وقام العالم ونستون في معهد (MIT) للتكنولوجيا بتلخيص كل ما تم تطويره من أبحاث في مجال معالجة اللغات البشرية والمعالجة بالمنطق الرمزي ورؤية الحاسب والروبوتات (الإنسان الآلي).

3- المرحلة الثالثة (1975 –حتى الآن)

ويطلق عليها (المرحلة الحديثة) والتي بدأت منتصف السبعينات، والتي تميزت بظهور التقنيات المختلفة التي تعالج كثير من التطبيقات، التي أدت فعلاً إلى انتقال جزء كبير من الذكاء الإنساني إلى برامج الحاسوب، وتعتبر هذه الفترة هي العصر الذهبي لازدهار هذا العلم والتي أدت إلى ظهور كثير من نظم الذكاء الاصطناعي الحديثة، كوضع حل للمشكلات والانظمة الخبيرة واثبات النظريات.

الأنظمة الخبيرة: برنامج حاسوبي ذكي يستخدم مجموعة من المعارف الاولية للخبراء والعلاقات بينها لحل المشكلات في مجال معين كالمجال التعليمي او الطبي وغيره.

خصائص برامج الذكاء الاصطناعي

1 التمثيل الرمزي (Symbolic Representation)

إذا نظرنا إلى الخلايا العصبية في الإنسان -لوجدنا أن الفهم الإنساني يعتمد على الوضع الثنائي للخلايا (شحنات موجبة (+) وأخرى سالبة (-) مما يشير إلى إمكانية التعبير عن الأفكار والتصورات والمفاهيم بالغة التعقيد واتخاذ القرارات بتشكيلات متطوّرة من هذه الأوضاع (الحالات الثنائية). وكذلك الحاسب الذي يتكون من بتات ثنائية binary bit ولا يمكن لهذه البتات أن تتخذ إلا أحد وضعين اتفق على أن يرمز لهما بـ (1 أو صفر). وقد أدى اختيار هذين الرمزين الرقميين إلى

انتشار الفكرة القائلة إن الحاسب لا يستطيع أن يتفهم سوى (نعم أو لا)، وأنه لا يستطيع تمييز المعنى بينهما. ولا شك أن إمكانية التعبير عن المفاهيم المعقدة بواسطة الرموز الثنائية التي يفهمها الحاسب تجعل محاكاة عملية اتخاذ القرارات ممكنة.

إن السمة الأولى لبرامج الذكاء الاصطناعي هي أنها تستخدم أساساً رموزا غير رقمية مما يشكل تناقضاً للفكرة السائدة أن الحاسب لا يستطيع أن يتناول سوى الأرقام، ولا يوجد بالطبع ما يمنع برامج الذكاء الاصطناعي من أداء بعض العمليات الحسابية إذا لزم الأمر، ولكن غالبا ما تستخدم نتائج هذه العمليات على المستوى الإدراكي، بمعنى أن مغزى هذه العمليات الحسابية سيدخل إلى العملية الاستدلالية التي يقوم بها البرنامج، ويوضح ذلك مثال من برامج التشخيص الطبي الذي قد يعطينا معلومة معينة في صورتها الرمزية بالشكل الآتي:

مثال **((المريض يعاني من حمى بسيطة))** وقد توصل إليها الحاسب بقيامه بعملية استدلالية من معلومة رقمية مثل: **((درجة حرارة المريض 38 درجة مئوية))**

كما يمكن لبرامج الذكاء الاصطناعي أن تستخدم معلومة رمزية من علم أمراض النبات مثل:

((العفن نوع من الفطر)) للتوصل إلى معلومة رمزية أخرى كالآتي:

((الأضرار التي يسببها الفطر بشكل عام يمكن أن يسببها العفن بشكل خاص))

وعادة ما يطلق على هذا النوع من العمليات (الاستدلالية أو الاستقرائية) وله أهمية بالغة في علم الذكاء الاصطناعي.

2 الاجتهاد أو الاستدلال: Heuristics

تتحدد السمة الثانية لبرامج الذكاء الاصطناعي بنوعية المسائل التي تتناولها. فهي في العادة ليس لها حل خوارزمي معروف، ونعني بذلك عدم وجود سلسلة من الخطوات المحددة التي يؤدي اتباعها إلى ضمان الوصول إلى حل للمسألة. وطالما لا يوجد حل خوارزمي للمسائل التي يعالجها الذكاء الاصطناعي فلا بد من الالتجاء إلى الاجتهاد، أي إلى الطرق غير المنهجية والتي لا ضمان لنجاحها. ويتمثل «الاجتهاد» في اختيار إحدى طرق الحل التي تبدو ملائمة مع إبقاء الفرصة في نفس الوقت للتغيير إلى طريقة أخرى في حالة عدم توصل الطريقة الأولى إلى الحل المنشود في وقت مناسب.

ولهذا لا تعد البرامج التي تحل المعادلات التربيعية ضمن برامج الذكاء الاصطناعي لأن لها حلا خوارزميا معروفا. أما البرامج التي تحاول حل «التكامل الرمزي» فيتم اعتباره ضمن برامج الذكاء الاصطناعي لاعتمادها طريقة حل أخرى كلما فشلت الطريقة السابقة لتبسيط عملية التكامل. وتشكل برامج لعب الشطرنج مجالا خصبا للذكاء الاصطناعي لأنه لا توجد طريقة معروفة لتحديد أفضل نقلة ممكنة في مرحلة معينة من دور الشطرنج، وذلك لسببين:

- أولهما أن عدد الاحتمالات الممكنة كبير جدا لدرجة يستحيل معها إجراء بحث search كامل عليها.
- السبب الآخر هو أننا لا نعرف سوى القليل عن المنطق الذي يبني عليه اللاعبون المهرة تحركات قطعهم إما لأنهم ليسوا مدركين له بشكل واع، أو لأنهم لا يريدون الإفصاح عنه. وقد اعتاد بعض المستهينين بالذكاء الاصطناعي-ومنهم هربرت دريفس- الادعاء بعدم استطاعة أي برنامج للوصول إلى مستوى اللاعب الإنساني الجيد. وقد ثبت خطأ هذا الادعاء عام 1985. وتستطيع الآن برامج الشطرنج الممتازة هزيمة أغلبية اللاعبين.

3 تمثيل المعرفة: Knowledge Representation

تختلف برامج الذكاء الاصطناعي عن برامج الإحصاء في قدرتها على تمثيل المعرفة. فهي تعبر عن تطابق بين العالم الخارجي والعمليات الاستدلالية الرمزية بالحاسب. ويمكن فهم

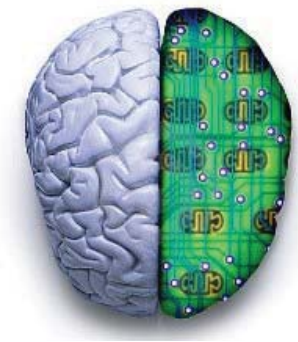
تمثيل المعرفة هذا بيسر لأنه عادة لا يستخدم رموزا رقمية. فقد يستخدم أحد برامج التشخيص العلاجي القاعدة التالية في تشخيص حالة المريض بالأنفلونزا: **((إذا كانت درجة حرارة المريض عالية، ويشعر بآلام عضلية وصداع، فإن هناك احتمالا قويا بأنه يعاني من الأنفلونزا))**. ويكون التعبير عن مثل هذه القاعدة في برامج الذكاء الاصطناعي بوضوح وإيجاز وبلغة أقرب ما تكون إلى لغتنا الطبيعية وليس بلغة الحاسب. والتعبير عن هذه القاعدة في البرامج التقليدية يتطلب إضافة جداول كثيرة ومتعددة للتعبير عن العلاقة بين الأعراض المرضية وتلك الأمراض التي يحتمل أن تسببها. وحتى في هذه الحالة سيكون من الصعب جدا على البرنامج أن يفسر طريقة توصله إلى الحل كما تفعل برامج الذكاء الاصطناعي.

ومن أهم ما يميز طرق بناء برامج الذكاء الاصطناعي الفصل التام بين قاعدة المعرفة ونظم المعالجة التي تستخدم هذه المعرفة. فمواد المعرفة واضحة، ودلالاتها ومعانيها مفهومة، أما ما يُكتب بلغة البرمجة-الذي يصعب فهمه لغير المتخصص-فهو مجموعة نظم المعالجة التي تفسر مواد المعرفة هذه وهي تحدد في أي حالة وفي أي مرحلة من مراحل البرنامج يكون أي من قوانين الاستدلال فعالا.

4 البيانات غير الكاملة

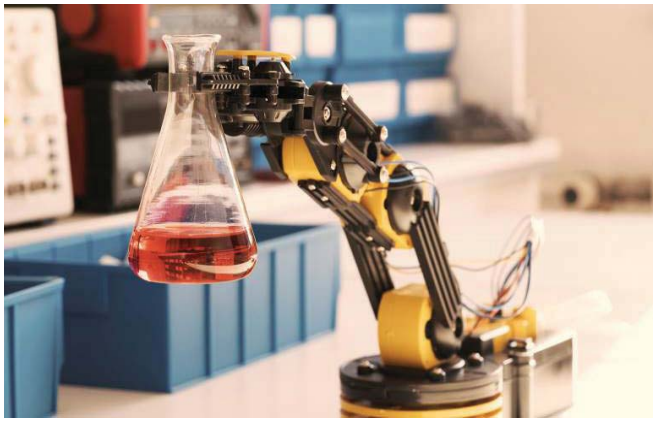
تتمثل السمة الرابعة لبرامج الذكاء الاصطناعي في قدرتها على التوصل لحل المسائل حتى في حالة عدم توفر جميع البيانات اللازمة وقت الحاجة لاتخاذ القرار. ويحدث ذلك كثيرا في الطب حين لا تكون نتائج التحاليل جاهزة وحالة المريض لا تسمح بالانتظار ولا يستطيع الطبيب في هذه الحالة انتظار نتائج التحاليل التي سيستفيد منها بالتأكيد ويضطر إلى اتخاذ قرار سريع. ويترتب على نقص البيانات اللازمة كون النتيجة التي تم التوصل إليها غير مؤكدة، أو كونها أقل صوابا مع احتمال خطئها في بعض الأحيان. وكثيرا ما نتخذ قرارات في حياتنا العملية مع غياب جميع البيانات اللازمة، وبالتالي يظل احتمال خطأ القرار قائما. ويكون غياب بعض البيانات أحيانا نتيجة لطبيعة المسألة نفسها ولا بد من التخمين أو التقدير.

5 البيانات المتضاربة



أما السمة الخامسة لبرامج الذكاء الاصطناعي فهي قدرتها في استخدام استراتيجيات مختلفة للتعامل مع البيانات التي قد يناقض بعضها بعضا، تنشأ الحاجة إلى مثل هذه الاستراتيجية عندما يتم استيفاء شروط قاعدتين أو أكثر بواسطة الحقائق المعروفة حاليًا. وهذا ما نسميه البيانات المتناقضة ونعني بها ببساطة تلك البيانات التي يشوبها بعض الأخطاء.

6 القدرة على التعلم



تمثل القدرة على التعلم، أحد معايير السلوك المتسم بالذكاء وتؤدي إلى تحسين الأداء. وترتبط هذه الملكة بالقدرة على التعلم باستطاعة استشراف التماثل في الأشياء والقضايا والتوصل من الجزئيات إلى العموميات واستبعاد المعلومات غير المناسبة.

ويجد الباحثون في علم الذكاء الاصطناعي صعوبة في تحديد الحالات التي يكون التعميم فيها جائزا وإدراك الأحوال التي لا يصح فيها التعميم. كما يجدون نفس القدر من الصعوبة في تحديد السياق الذي يكون التماثل في إطاره صحيحا. ووجد باحثو الذكاء الاصطناعي في قدرة الإنسان على استبعاد المعلومات غير المناسبة مشكلة دقيقة للغاية، ذلك أن من المميزات الهائلة للحاسب، والتي جعلته مفيدا في أغراض عديدة هي أنه-بخلاف العقل الإنساني-قادر على عدم نسيان الأشياء. والإشكالية المطروحة أمام الذكاء الاصطناعي هي أن قدرة الإنسان على النسيان هي بالتحديد التي تعطيه القدرة الهائلة على التعلم. فالإنسان قادر على نسيان أو تناسي التفاصيل الكثيرة غير الهامة-للتركيز على ما هو أهم طبعا- ويستطيع في نفس الوقت استرجاع هذه التفاصيل عند الحاجة. وتصبح مشكلة إعطاء الحاسب قدرة على التعلم مركزة في جعله قادرا على التمييز بين الحقائق الهامة (التي يجب أن يتذكرها) والحقائق غير الهامة (والتي يمكن أن ينساها).

علل: القدرة على استخلاص مغزى مجموعة من الحقائق بدلا من تخزينها جميعا في الذاكرة هي واحدة من عوامل القوة العظيمة للإنسان.

تطبيقات الذكاء الاصطناعي

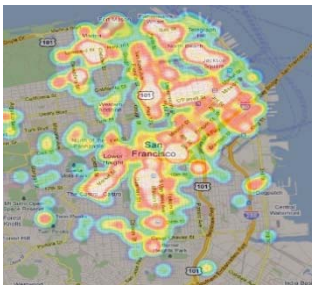
تتعدد تطبيقات الذكاء الاصطناعي في العديد من المجالات مثل:

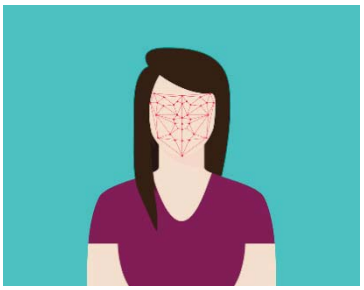
1 **الألعاب:** تلعب الذكاء الاصطناعي دورًا مهمًا في الألعاب الاستراتيجية مثل لعبة الشطرنج، والبوكر، و tic-tac-toe، وما إلى ذلك، حيث يمكن للآلة التفكير في عدد كبير من المواقف المحتملة بناءً على المعرفة الاستكشافية (Heuristic knowledge).

2 **معالجة اللغة الطبيعية** - من الممكن التفاعل مع الكمبيوتر الذي يفهم اللغة الطبيعية التي يتحدث بها البشر.

3 **أنظمة الرؤية:** تقوم هذه الأنظمة بفهم وتفسير وفهم المدخلات المرئية على الكمبيوتر. فمثلا:

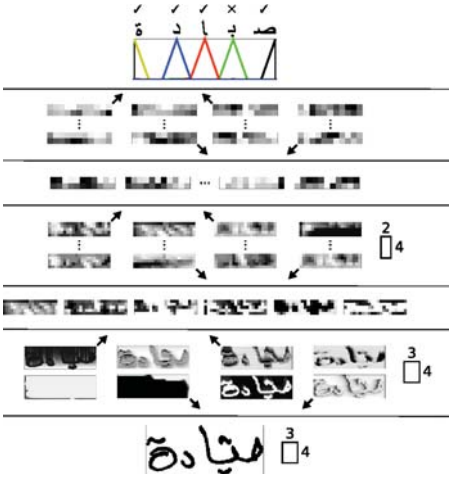
- تلتقط طائرة التجسس صورًا تُستخدم لمعرفة المعلومات المكانية أو خريطة المناطق.
- يستخدم الأطباء نظام خبير سريري لتشخيص المريض.
- تستخدم الشرطة برامج الكمبيوتر التي يمكنها التعرف على وجه المتهم من خلال الصورة التي يجلبها المحقق الجنائي.





4 **التعرف على الكلام** - بعض الأنظمة الذكية قادرة على سماع وفهم اللغة من حيث الجمل ومعانيها بينما يتحدث الإنسان إليها. يمكنه التعامل مع لهجات مختلفة، كلمات عامية، ضجيج في الخلفية، تغيير في صوت الإنسان بسبب المرض، إلخ.

5 **التعرف على خط اليد** - يقوم برنامج التعرف على خط اليد بقراءة النص المكتوب على الورق بواسطة قلم أو على الشاشة بواسطة قلم. يمكنه التعرف على أشكال الحروف وتحويلها إلى نص قابل للتحرير.



6 **الروبوتات الذكية** - الروبوتات يمكن للروبوت تنفيذ التعليمات التي يقدمها الإنسان في تمتلك أجهزة استشعار لاكتشاف البيانات من العالم الحقيقي المحيط بها مثل الضوء ودرجة الحرارة والحركة والصوت والصدمات والضغط. كما تحتوي على معالجات فعالة وأجهزة استشعار متعددة وذاكرة ضخمة. بالإضافة إلى ذلك، لدى الروبوتات القدرة على التعلم من الأخطاء السابقة ويمكنها أيضا التكيف مع متغيرات البيئة.

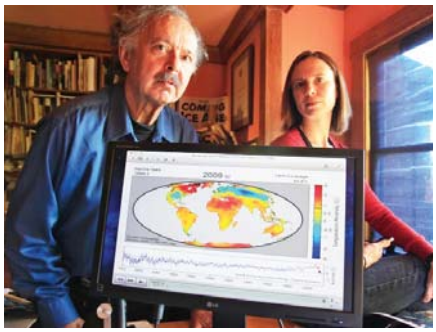
سؤال

اذكر تطبيقات اخرى للذكاء الاصطناعي؟

واقع ومستقبل الذكاء الاصطناعي

لم يحدث أن شهدت البشرية مرحلة أكثر تطوّراً في مجال الذكاء الاصطناعي كما تشهده الآن، فلا يكاد يمر يوم دون حدوث تطور جديد في هذا المجال الذي من المتوقع أن يغيّر حياة البشرية للأبد. ولكن، هل التغيير باتجاه الأفضل أم الأسوأ؟ وهل سيساهم في نقل البشرية للأمام أم سيُعجّل بنهايتها؟ وهل سيجعل حياتنا أسهل أم أكثر تعقيدا؟

1- التصدي للكوارث: من خلال التنبؤ بالكوارث الطبيعية (كالزلازل والبراكين) ومكافحة الشيخوخة، والتّصدي لحوادث المرور وحالات الوفاة، كما سيساعد في منع الكوارث النووية كما حدث في محطة فوكوشيما للطاقة.



2- الإنسان والحاسوب وحدة واحدة: يتوقّع الباحثون أن يكون الإنسان والحاسوب وحدة واحدة غير منفصلة، بحيث سيحسن حياة البشر في كافة مناحي الحياة، مثل: الأطراف الصناعية الذكية والتي يُمكِن للإنسان التحكم بها وتطويعها لرغبته.

3- حلّ مشكلة تغيّر المناخ: يُعتقَد أنّ الأنظمة التي سيوفّرها الذكاء الاصطناعي قادرة على الإجابة عن الكثير من تساؤلات البشر، مثل: علاج الأمراض، والقضاء على الفقر، ومنع الكوارث البيئية، وكذلك معالجة مشكلات التغيّر المناخي والاحتباس الحراري.

4- إمكانيّة إنقاذ العالم: الكثير من أحلام البشرية، مثل: السفر للفضاء، والقضاء على الأمراض والفيروسات القاتلة، قد يكمن إمكانية تحقيقها في علم الذكاء الاصطناعي.

5- التنبؤ بالمستقبل: ليس المقصود هنا التنبؤ بمصائر الناس، فلا يعلم الغيب الا الله؛ بل توقع ما نحن بصدد القيام به، ماذا سنشتري؟ أو أين سنذهب؟ أو الأماكن التي سنزورها، ومستقبل الأعمال كذلك، فالتعلم الآلي سوف يعتمد على معطيات الماضي لمحاولة التنبؤ بالمستقبل، ومعرفة الأشياء حولنا وما سيحدث.

6- مساعدة كبار السن وذوي الاحتياجات الخاصة: الروبوتات المنزلية ستساعد هؤلاء في أنشطة الحياة اليومية، مثل: التأكد من تناولهم الدواء، والمساعدة في إعداد طعامهم، ومساعدتهم على ممارسة الرياضة والحركة وتنفيذ النصائح الطبية والعلاجية.

◀ 7- **تحسين الرعاية الطبية:** الطب أحد أهم تطبيقات الذكاء الاصطناعي، حيث ستساعد في الحصول على تسلسل الجينوم وحفظ التفاصيل الدقيقة للسجلات الطبية، والحصول على كميات هائلة من الدراسات السريرية، وتحسين طرق العلاج، وتطوير الأدوية، والعناية بالمرضى، وغيرها من المجالات التي ستحدث ثورة في مجال العناية الطبية.

◀ 8- **إتاحة المعلومات لمن لا يستطيع الحصول عليها:** من خلال القدرة على مساعدة البشر غير القادرين على القراءة والكتابة، وبالتالي لا يمكنهم الحصول على المعلومات.
يمكن للحاسوب تحدّث اللّغة الطبيعية، فيمكنه بذلك إعطاء الناس المعلومة التي لا يستطيعون الوصول إليها، فتساعد حقيقة في مجالات التعليم ونشر المعرفة.

◀ 9- **خسارة البشرية لمعظم الوظائف:** يتوقع حصول الروبوتات على معظم الأعمال والوظائف التي يقوم بها البشر (خاصة ذات الأعمال الروتينية منها)، وهذا من الجوانب السلبية، فعندما تعمل الروبوتات بدلاً من البشرية سيغير حياة الكثيرين؛ فقد يكون التخلّي عن العمل بداية النهاية للبشرية، و بداية حقبة جديدة من سيطرة الآلات على حياتنا ليعجّل بنهايتنا.

أسئلة الوحدة

س1 ما المقصود بالمصطلحات الاتية: الذكاء الاصطناعي، الانظمة الخبيرة.

س2 اذكر اهداف الذكاء الاصطناعي.

س3 عدد (على شكل نقاط) ميّزات الذكاء الاصطناعي، مع ذكر مثال على اثنتين منها.

س4 ما الفرق بين البرمجة التقليدية وبرمجة الذكاء الاصطناعي.

س5 علل:

1- لا يُعدّ برنامج حلّ المعادلة التربيعية ضمن برامج الذكاء الاصطناعي.

2- شرع الخبراء في دراسة تفكير الانسان ومحاولة محاكاته بالحاسب.

تم بحمد الله

لجنة المناهج الوزارية:

د. صبري صيدم	أ. ثروت زيد	د. شهناز الفار
د. بصري صالح	أ. عزام أبو بكر	د. سمية نخالة
م. فواز مجاهد	أ. علي مناصرة	م. جهاد دريدي

فريق إعداد الخطوط العريضة:

أ. ابراهيم قدح	أ. سهام بدران	أ. مهند أبو الهيجا	أ. جميل الناطور
م. خلدون عوايصه	م. صابرين أبو عياش	م. معاذ أبوسليقة	

المشاركون في ورشة اقرار الكتاب:

ابراهيم قدح	ولاء شتية	ختام عيد	علا ضميدي
سامي غنّام	رولا القاضي	سهام بدران	جهاد خلوف
منذر شواهنة	ربا حمو	رامي نصار	نورالدين جبرين
منار عياد	روند قنيص	وليد بدوي	معاذ أبوسليقة
سمر أبو حجلة	سماهر غياظة	محمد مصري	جميل الناطور