قواعد البيانات Databases

جامعـــــــــة ديالـــــــــى
كليـــة التربيـــة للعـــلوم الصرفـــة
قســـم عـــلوم الحاسبـــات
المرحلـــــة الثانيـــــة

Asst.Prof.Dr. Ali Mohammed Saleh Ahmed
University of Diyala.. College of Education for Pure Sciences.. 2nd Class.. Computer Dept.
dr.alimahmed@uodiyala.edu.iq

قواعد البيانات ونظم إدارة قواعد البيانات

Database & Database Management System DBMS

تُعتبر قواعد البيانات من أهم أسباب تطور و تقدم نظم المعلومات، إذ تُستقبل بيانات العالم في قواعد البيانات، كما تُغذي قواعد البيانات أنظمةً و تقاريرَ أخرى لأهداف مختلفة، إختلاف البرامج التي تُغذي قواعد البيانات و إختلاف الأهداف من وجودها يحتم عليها أن تتخذ أشكالاً و نماذج عدة.

١- البيانات DATA:-

هي الأرقام أو الحروف أو الرموز أو الكلمات القابلة للمعالجة بواسطة الحاسب مثل: الرقم(٦٥).

٢- المعلومات Information:-

هي بيانات تم تنظيمها أو معالجتها لتحقيق أقصى استفادة منها.

مثال: الرقم (٦) والرقم (٥) عبارة عن بيانات إذا استخدما في عملية الضرب ٥×٦ أصبحا معلومة مفيدة.

٣- قواعد البيانات Data base:-

هي تجميع لكمية كبيرة من المعلومات أو البيانات وعرضها بطريقة أو بأكثر من طريقة تسهل الاستفادة منها.

تعريف اخر: هي مجموعة من البيانات و المعلومات المخزنة بطريقة نموذجية ودون تكرار و المتصلة مع بعضها وفق علاقات و يدير قواعد البيانات نظام يسمى نظام إدارة قواعد البيانات. يتم استرجاع البيانات باستخدام أوامر من لغة الاستعلام.

مثال : دليل الهاتف الذي يشتمل على أسماء وعناوين وأرقام هواتف لمدينة معينة يمكن أن نعتبره قاعدة بيانات وبتحقيق الاستفادة من قاعدة البيانات هذه بإدخال رقم المشترك والحصول على أسمه وعنوانه أو إدخال أسم المشترك والحصول على رقم هاتفه وعنوانه وهكذاً.

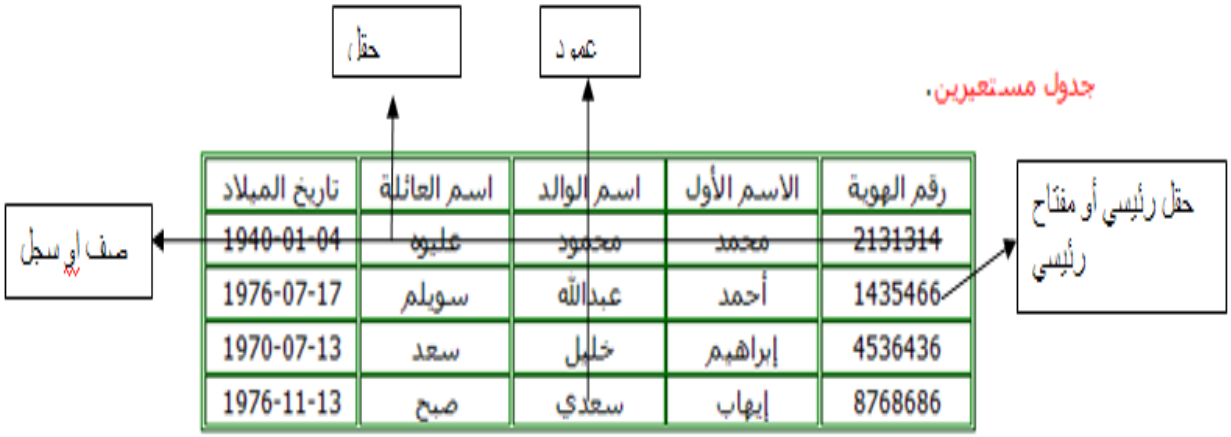
تتكون قاعدة البيانات من :

١- الجداول ٢- السجلات ٣-الحقول

جداول البيانات ← (Tables) :هي مجموعة من السجلات (كل سجل يعتبر سطر داخل الجدول) ، وتحتوي هذه السجلات على معلومات من نفس النوع. ويمكن اعتبار دليل التليفونات جدول واحد.

السجلات ← (Records) السجل عبارة عن سطر واحد في الجدول ويحتوي على مجموعة من الحقول. ويمكن اعتبار أي سطر من أي صفحة من دليل التليفونات سجلاً.

الحقول ← (Fields) هو عنصر محدد داخل السجل قد يكون (في دليل الهاتف مثلا) اسم المواطن أو العنوان أو رقم التليفون ويتم تحديده من حيث النوع والحجم بواسطة مصمم البرنامج.



رقم الهوية	الاسم الأول	اسم الوالد	اسم العائلة	تاريخ الميلاد
2131314	محمد	محمود	عليوه	1940-01-04
1435466	أحمد	عبدالله	سويلم	1976-07-17
4536436	إبراهيم	خليل	سعد	1970-07-13
8768686	إيهاب	سعدي	صبح	1976-11-13

طرق ربط العلاقات في قواعد البيانات:

كل جدول نقوم بإنشائه يجب ان نجعل له ما يسمى المفاتح الرئيسي (PRIMARY KEY) وهذا المفتاح لا يتم تكراره في الجدول , حيث ان كل مستخدم لديه رقم رئيسي نعرفه به , ونحدد به اسمه ومعلوماته الاخرى ولكي نقوم بربطه في علاقة , في الجدول المربوط به نجعل هذه القيمة على شكل مفتاح اجنبي (FOREIGN KEY):

CUSTOMERS

CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Yarraville

ORDERS

OrderID	CustomerID	Amount	Date
1	3	27.50	02-Apr-2000
2	1	12.99	15-Apr-2000
3	2	74.00	19-Apr-2000
4	4	6.99	01-May-2000

لاحظ ان العمود CustomerID في جدول CUSTOMERS هو مفتاح رئيسي ولا يمكن تكراره , ولكنه في جدول ORDERS على شكل (FOREIGN KEY) ويمكن تكراره , لان عدة طلبات يمكن عملها عن طريق مستخدم واحد , ولكن كل طلب مربوط بمستخدم واحد فقط.

أهمية ومزايا قواعد البيانات:-

- تخزين جميع البيانات بكافة الأنشطة لجهة ما بطرق متكاملة ودقيقه وتصنيف وتنظيم هذه البيانات بحيث يسهل استرجاعها في المستقبل.
- متابعة التغيرات التي تحدث في البيانات المخزنة وإدخال التعديلات اللازمة عليها، حتى تكون دائماً في الصورة الملائمة لاستخدامها فور طلبها.
- تخزين كم هائل من البيانات التي تتجاوز الإمكانيات البشرية في تذكر تفاصيلها ومن ثم إجراء بعض العمليات والمعالجات التي يستحيل تنفيذها يدوياً.
- تساعد على تخزين البيانات بطريقه متكاملة، بمعنى الربط بين النوعيات المختلفة للبيانات المعبرة عن كافة الأنشطة.
- تساعد على تحقيق السرية الكاملة للبيانات المخزنة بها بحيث لا تتاح أية معلومات لأي شخص ليس له الحق في الاطلاع عليها.
- امكانية التقليل من التكرار غير المبرر للبيانات.
- تحقيق مبدأ المشاركة في البيانات و المحافظة على تكامل البيانات.

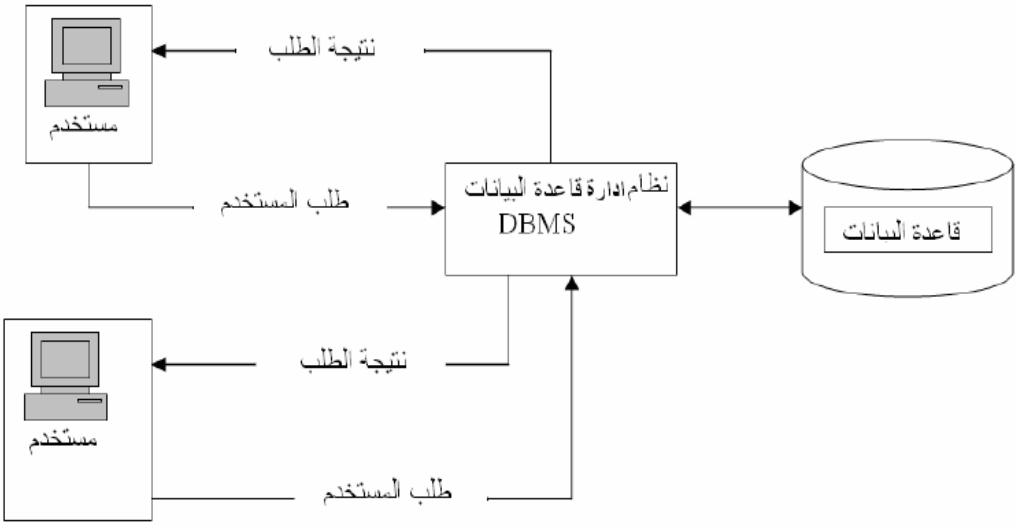
- وظائف قواعد البيانات:-

- إضافة معلومة أو بيان جديد إلى الملف.
- حذف البيانات القديمة والتي لم تعد هناك حاجة إليها.
- تغيير بيانات موجودة تبعاً لمعلومات استحدثت .
- البحث والاستعلام عن معلومة أو معلومات محددة .
- ترتيب وتنظيم البيانات داخل الملفات .
- عرض البيانات في شكل تقارير أو نماذج منظمه .
- حساب المجموع النهائي أو المجموع الفرعي أو المتوسط الحسابي لبيانات مطلوبة .

- **نظم إدارة قواعد البيانات Database Management Systems (DBMS):-**

هي مجموعة من البرامج التي تدير و تتحكم بعملية تخزين و استرجاع البيانات وكذلك توفر امكانية عدد كبير من المستخدمين من الوصول الى قاعدة البيانات و التعامل معها و ينظر اليها انها حلقة وصل بين المستخدمين و قاعدة البيانات حيث تقوم باستقبال طلبات المستخدمين و من ثم نقلها الى قاعدة البيانات و تنفيذ البرامج اللازمة لتنفيذ هذه المتطلبات و من ثم تزويد المستخدم بالنتائج المطلوبة مثل : Oracle,

SQL SERVER ,MS Access, FoxPro…



الفرق بين قاعدة البيانات ونظام ادارة قواعد البيانات:

قاعده البيانات: هي عباره عن وعاء الكتروني يتم حفظ جميع البيانات عليها ويتم استرجاع جميع هذه البيانات بشكل بسيط وسهل

نظام اداره قواعده البيانات : تنقسم الى لغة تعريف البيانات ولغة معالجه البيانات وهي التي تمتلك الحذف والإضافة وانشاء الجداول والحقول وعمل العلاقات بين الجداول.

وظائف نظم إدارة قواعد البيانات DBMS:

- إنشاء قواعد البيانات
- تحديث قواعد البيانات
- تشغيل قواعد البيانات
- صيانة قواعد البيانات
- حماية وتأمين سلامة البيانات
- إضافة معلومة أو بيان جديد إلى الملف
- حذف البيانات القديمة والتي لم تعد هناك حاجة إليها
- تغيير بيانات موجودة تبعاً لمعلومات استحدثت
- البحث والاستعلام عن معلومة أو معلومات محددة
- ترتيب وتنظيم البيانات داخل الملفات
- عرض البيانات في شكل تقارير أو نماذج منظمه
- حساب المجموع النهائي أو المجموع الفرعي أو المتوسط الحسابي لبيانات مطلوبة

مكونات نظام قاعدة البيانات

يقسم نظام قاعدة البيانات إلى خمسة أقسام :

١ - **المكونات المادية (Hardware) :**

وتشمل جميع الأجهزة المادية في النظام مثل الحاسبات، الأجهزة الطرفية، الطابعات وكذلك أجهزة الاتصال في بيئة قاعد البيانات الموسعة... الخ.

٢ - **البرمجيات (Software):**

وهي مجموعة البرامج المستخدمة في قاعدة البيانات، وتقسم إلى ثلاثة أقسام:

أ - **أنظمة التشغيل**: وهي البرامج التي تقوم بإدارة الأجهزة وتهيئتها للعمل وتمكين بقية البرامج من العمل مثل Linux, Unix, Windows....

ب - **برنامج قاعدة البيانات**: وهو البرنامج الذي يتولى إدارة قاعدة البيانات مثل Oracle, Sybase, DB2 ... .

ج - **البرامج التطبيقية والبرامج المساعدة**: وهي البرامج التي تقوم بعمليات الاسترجاع والتخزين وكذلك استخراج التقارير... .

٣ - **المستخدمون** : وهم عبارة عن الأشخاص الذين يقومون بالعمل في بيئة قاعدة البيانات وهم :

أ - **مدير النظام**: وهو الشخص المسؤول عن إدارة عمل البيئة العامة التي يعمل بها نظام قاعدة البيانات ويقوم بما يلي:

١. بإدارة المستخدمين ومنح الصلاحيات لاستخدام النظام.
٢. إدارة أجهزة التخزين والأجهزة الأخرى.
٣. متابعة عمل النظام.

ب -**مدير قاعدة البيانات**: وهو المسؤول عن إدارة قاعدة البيانات وتشمل واجباته:

١. تحديد متطلبات قاعدة البيانات من برامج وتجهيزات.
٢. متابعة نظام قاعدة البيانات وتنسيق عملية استخدامه.
٣. توفير الأمن والحماية للنظام.
٤. تصميم آليات المحافظة على قاعدة البيانات وتحديد الإجراءات اللازمة لتوفير الخدمات للمستخدمين الآخرين.

ج - **مصمم قاعدة البيانات** وهو الشخص(الأشخاص) الذي يقوم بعملية تصميم قاعدة البيانات وتشمل واجباته:

١. تحديد البيانات الواجب تخزينها في قاعدة البيانات
٢. تصميم أفضل التراكيب لحفظ البيانات .
٣. تصميم قاعدة بيانات خالية من التكرار .
٤. تحديد طرق الوصول والمعالجة والاسترجاع للبيانات من خلال تصميم الشاشات والتقارير الواجب استخدامها .
٥. توثيق عملية التصميم وطرق الوصول للبيانات .

د - **المبرمجون ومحللو النظم** : وهم الأشخاص الذين يقومون بعملية تصميم البرامج وتنفيذها وتشمل واجباتهم :

١. تصميم التطبيقات وتحويلها إلى برامج بلغة(لغات) برمجة حسب السياسات المقررة في عملية التصميم .
٢. تنفيذ وتطبيق تلك البرامج والتأكد من سلامتها .
٣. عمل الصيانة اللازمة لتلك البرامج .

هـ - **المستخدم النهائي:** وهو الشخص أو مجموعة الأشخاص الذين يقومون بالعمل اليومي على النظام وتطبيق البرامج في مجال محدد مثل الاسترجاع، التعديل، الحذف، تنفيذ التقارير... الخ .

٤ - **الإجراءات والعمليات:** وهي عبارة عن القوانين والتعليمات التي تحكم عمل قاعدة البيانات بشكل صحيح وتكون على شكل تعليمات موثقة بشكل واضح ومحدد.

٥ - **البيانات**: وهي أهم مكونات النظام حيث تشمل مجموعة الحقائق المخزنة في قاعدة البيانات. وكون البيانات تكون على شكل بدائي إذ لابد من تحديد مكان وكيفية التخزين لهذه البيانات حتى تسهل عملية معالجتها والاستفادة منها وهذا عمل المصمم .

أهمية تصميم قواعد البيانات:

إن عملية بناء قاعدة بيانات جيدة لا يأتي بتلك السهولة، إذ لابد من بذل جهد كبير للحصول على قاعدة بيانات جيدة. والتصميم الجيد لقاعدة البيانات يسهل عملية استخدام وإدارة هذه القاعدة أما التصميم السيئ فسيؤدي إلى تكرار البيانات(ويعني وجود نفس البيانات في أكثر من مكان) وبالتالي تصعب عملية الحفاظ على توافقية البيانات وعادة ما يؤدي تكرار البيانات إلى نتائج غير صحيحة عند طلب تلك البيانات من تلك القاعدة وهذا بدوره يؤدي إلى أن أي قرارات إدارية وكذلك أي تخطيط مستقبلي سيكون خاطئا لاعتماده على معلومات غير صحيحة.

دورة الحياة لنظام قاعدة البيانات:

١ - الدراسة المبدئية للنظام القائم وتشمل ما يلي:

أ - تحليل الوضع الحالي للمؤسسة ومعرفة طبيعة الإجراءات المستخدمة والتعليمات وقواعد العمل .

ب -تحديد المشاكل التي تواجه النظام المستخدم وكذلك القيود المادية مثل الطاقة البشرية والتمويل المتوفر لتطوير أو استبدال النظام الحالي .

ج - تحديد الأهداف الواجب تحقيقها والمزايا المطلوبة في النظام الجديد.

٢ - **تصميم قاعدة البيانات**: وتعتبر هذه المرحلة من أهم المراحل في دورة حياة النظام إذ لابد من بذل جهد كبير لتصميم النظام للوصول إلى نظام جيد وتؤدي الأهداف المرجوة من عمل النظام وتشمل عملية التصميم ما يلي:

أ - بناء نموذج المفاهيم وتشمل هذه العملية عدة خطوات(سنتطرق إلى هذه العملية بالتفصيل في الفصول اللاحقة):

١. تحليل البيانات ومتطلبات المستخدمين والإجراءات المطلوبة
٢. تعريف وتحديد الكيانات وخصائصها وعلاقتها مع بعضها وكذلك وضعها في الصيغة المعيارية.
٣. رسم مخطط المفاهيم وهو عبارة عن نموذج رسومي يوصف كيانات النظام وعلاقتها مع بعضها.
٤. تعديل النموذج بحيث يشمل الإجراءات الرئيسة، وقواعد عمليات الإضافة والتعديل والحذف على البيانات والتقارير، والشاشات، ومقدار التشاركية و توافقية البيانات....

.

ب - اختيار نظام إدارة قاعدة البيانات (DBMS).

ج - تحويل نموذج المفاهيم إلى نموذج داخلي بالاعتماد على نظام إدارة قاعدة البيانات(DBMS).

د - التصميم المادي وتتم خلاله عملية وضع مواصفات التخزين والوسائط المستخدمة في عملية التخزين وطرق الوصول للبيانات بالاعتماد على نظام إدارة قاعدة البيانات (DBMS).

٣ - **تنفيذ النظام:** وخلال هذه المرحلة تتم عملية إنشاء الجداول وكتابة جميع البرامج اللازمة لتنفيذ متطلبات النظام من الشاشات المختلفة و التقارير المطلوبة ... .

٤ - **عملية الفحص والتقييم للنظام وتشمل :**

أ - فحص قاعدة البيانات والتأكد من عملها بشكل صحيح.

ب -تقييم عمل البرامج والتطبيقات المستخدمة.

٥ - **تطبيق النظام في مكان العمل:** وتشمل هذه العملية عمليات إنشاء الجداول والمستخدمين والصلاحيات..، وتحميل جميع البرامج والتطبيقات وتنفيذها في البيئة الحقيقة التي يجب أن يعمل بها النظام.

٦ -**متابعة عمل النظام:** وهذه العملية تستمر طيلة فترة حياة النظام للتأكد من عمله بشكل صحيح وكذلك تعديل النظام ليتواءم مع المتطلبات الجديدة لبيئة العمل مثل تغير القوانين والأنظمة وقواعد العمل.

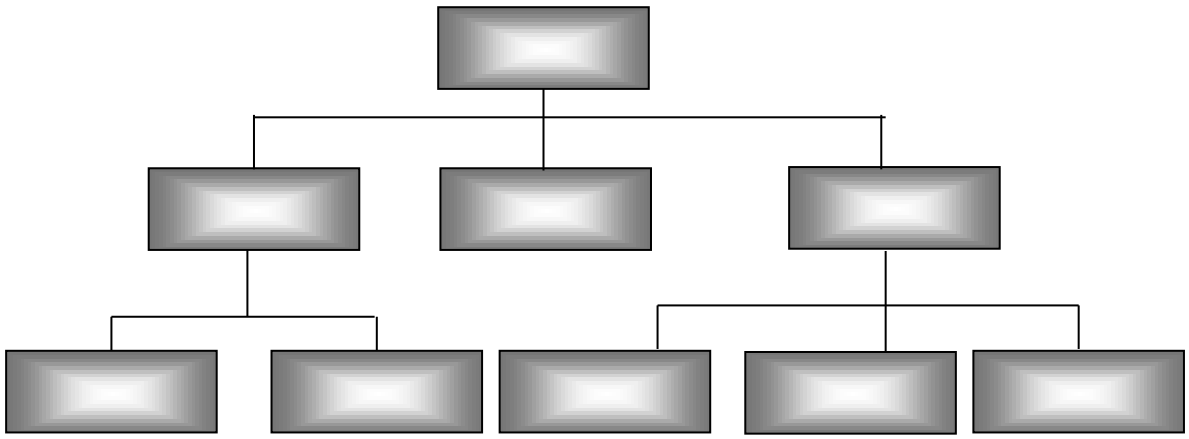
انواع قواعد البيانات:

١. قواعد البيانات اليدوية Manual Database

تلك البيانات المبعثرة في الأرف والأوراق التي تملأ المكتبات والمخازن تُمثل قواعد البيانات اليدوية ، و بسبب هذه البعثرة جاءت تصنيفات مهمة ومنطقية لتصنف البيانات حسب علاقتها ببعضها أو حسب بنيتها الرياضية او المنطقية، و بناء على ما ذكرتُ لك تُعتبر الملفات المخزنة في الأرف و فواتير المبيعات المجدولة إلكترونياً وعناوين الأشخاص في ملف وثائقي أمثلةً لبيانات يدوية.

٢. قواعد البيانات ذات الشكل الهرمي Hierarchy Database

وتعتمد هذه القاعدة على مبدأ التسلسل الهرمي في العمل ، حيث أنها تقوم بعمل تسلسل من الأصل ، أو الجذر ، حيث أن هذا النظام يبدأ في التفرع على شكل أقسام ، ويقوم مبدأ عمله على الوصول إلى البيانات بطريقة متسلسلة ومتفرعة ، وتكون إما من أسفل للأعلى أو من الأعلى للأسفل.



قاعدة بيانات هرمية

عيوب النموذج الهرمي:

١- يفتقد للمرونة و التجاوب الجيد مع المستخدم.

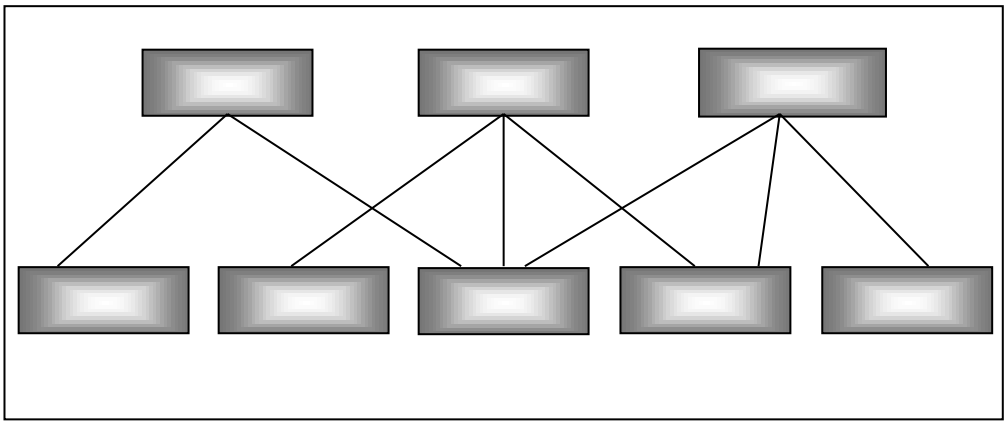
٢- التعقيد في البرمجة .

٣- ان البيانات تخزن في تركيب هرمي و بالتالي من الصعوبة اجراء تغير او تعديل على هذا التركيب.

٤- لا تقدم دعما مناسبة للاستعلامات غير المهيكلة.

٣. قواعد البيانات الشبكية Network Database

يتم تخزين البيانات في الهيكل الشبكي بصورة سلاسل مترابطة من البيانات و بالتالي يمثل هذا الهيكل علاقات منطقية أكثر تعقيدا . و لا تزال تستخدم قواعد البيانات الشبكية مع نظم إدارة قواعد البيانات لنظم الحاسوب الكبيرة ، تمثل هذه القاعدة نمط علاقات الكثير الى –كثير بين السجلات.



نموذج قواعد البيانات الشبكية

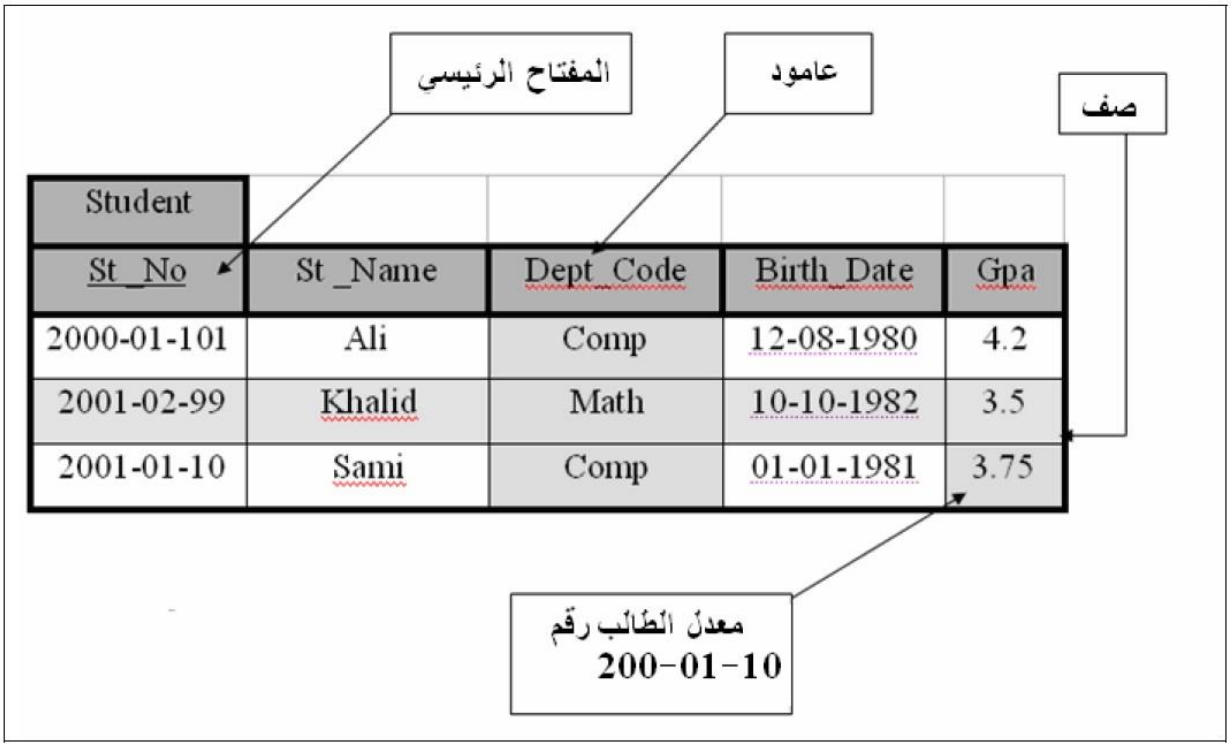
عيوب النموذج الشبكي: من عيوبه انه غير مرن و صعب او معقد من ناحية البرمجة و الصيانة . الا انه يعالج المعلومات بشكل كفوء.

٤. قاعدة البيانات العلائقية:

بدأ نشوء مفهوم قواعد البيانات العلائقية عام ١٩٧٠ عندما قدم العالم Codd اقتراحا لهذا النموذج والذي تم بناؤه على **نظريات الجبر العلائقي** ومن هنا برزت قوة هذا النموذج وسرعة انتشاره فيما بعد. ففي مطلع الثمانينات بدأت الكثير من الشركات بتبني هذا النموذج وتطبقيه، فنلاحظ الآن أن معظم أنظمة قواعد البيانات الموجودة في الأسواق تتوافق مع هذا النموذج. وتتلخص فكرة النموذج في النظر إلى قاعدة البيانات على أنها مجموعة من الجداول (Tables) أو علاقات تسمى (Relations) ومن هنا جاءت تسمية النموذج وكل جدول يجب أن يكون له اسم (لا يوجد أكثر من جدول يحمل نفس الاسم).والعلاقة هي عبارة عن مصطلح رياضي وتمثل جدولاً ذا بعدين (صفوف وأعمدة)، ولا توجد هناك أهمية لترتيب الصفوف أو الأعمدة.حيث تمثل الصفوف مجموعة سجلات الجدول(Records or Tuple) وتمثل الأعمدة الصفات لهذا الجدول (Attributes) ويجب أن يكون لكل صفة مجال (Domain) من القيم التي يمكن أن يحتويها هذا العامود. وترتبط هذه الجداول مع بعضها بواسطة روابط..ويجب أن يكون لكل جدول مفتاح رئيس (Primary Key) لتمييز الصفوف عن بعضها والنقطة التي تمثل تقاطع الصف مع العمود (الصفة) تمثل قيمة لهذا الصف.و سنقوم في بقية أجزاء هذه الوحدة بتقديم وصفاً لقواعد البيانات العلائقية (Relational Database) من حيث مكوناتها وأهم خصائصها.

الجدول التالي يمثل معلومات الطالب (Student) في قاعدة بيانات إحدى الجامعات

- اسم الجدول Student
- كل صف يمثل معلومات تخص طالباً واحد فقط.
- المفتاح الرئيس للجدول هو St _No كل طالب يجب أن يكون له رقم مختلف عن بقية الطلاب.
- الصفة Dept_Code تمثل القسم الذي ينتمي إليه أي طالب .
- نقطة تقاطع الصفة (Gpa) العمود مع الصف الثالث تمثل المعدل التراكمي للطالب رقم ٢٠٠١ - ٠١ -١٠ .
- مجال القيم:كل صفة يجب أن يكون لها مجال ثابت من القيم فمثلا Gpa يجب أن تحتوي على رقم حقيقي بين ١.. ٥. القسم Dept_Code يجب أن يكون أحد الأقسام الدراسية الموجودة في الجامعة.



- لا توجد هناك أهمية لترتيب الصفوف أو الأعمدة.فمثلا يمكن أن يكون الجدول السابق على الشكل التالي:

Student				
St_No	St_Name	Gpa	Birth_Date	Dept_Code
٢٠٠١-٠١-١٠	Sami	٣,٧٥	١٩٨١-٠١-٠١	Comp
٢٠٠١-٠٢-٩٩	Khalid	٣,٥	١٩٨٢-١٠-١٠	Math
٢٠٠٠-٠١-١٠١	Ali	٤,٢	١٩٨٠-٠٨-١٢	Comp

المفاهيم الخاصة بقواعد البيانات العلائقية:

- الجداول تتكون من سجلات وحقول وترتبط ببعضها البعض بعدة أشكال من العلاقات ولذلك سميت باسم قواعد البيانات العلائقية.
- السجل هو السطر الذي يحوي على كل معلومات العنصر من عناصر قاعدة البيانات مثل جدول الطالب، سيكون لكل طالب سجل يعرف عنه.
- الحقل هو عمود من الجدول ويعبر عن سمة من سمات الطالب كرقمه أو اسمه أو فرعه في الجامعة على سبيل المثال.
- العلاقة هي طريقة لربط الجداول مع بعضها البعض بهدف جمع المعلومات.

خصائص قواعد البيانات العلائقية:

- كل جدول في قاعدة البيانات العلائقية يضم نوع متكرر من السجلات
- لا يوجد ترتيب محدد للحقول
- لا يوجد ترتيب محدد للسجلات
- لكل سجل حقل أو عدة حقول
- إن تغير بيانات في أحد الجداول سيؤدي إلى تغير البيانات في جميع الجداول المرتبطة.

الفرق بين الهرمية والشبكية والعلائقية :

يَستخدم النموذجان الهرمي والشبكي روابط (links) أو مؤشرات (pointers) لوصل السجلات (Records) ببعضها البعض في النظام ، وتدعى هذه الأنظمة بالأنظمة الساكنة (static) أو المتراصة (monolithic) لأن السجلات فيها مربوطة ببعضها بشكل فيزيائي من خلال تعاريفها، وتتميز هذه الأنظمة بأنها معقدة العمل وصعبة التعديل، إلا أن سرعة الوصول فيها تغطي عيوبها .

أما في الأنظمة العلائقية فالربط بين السجلات لا يجري فيزيائياً عن طريق المؤشرات، وإنما عن طريق الأسماء الحقيقية للحقول ، كحقول رقم الموظف أو الاسم أو رقم البطاقة، فالسجلات في هذا النظام قابلة للعنونة بالمحتوى (connect-addressable) بحيث يجري الوصول إليها بمطابقة قيم البيانات المخزنة مع بعضها.

مفاتيح الجداول (العلاقات):

تعتبر المفاتيح من أهم خصائص قواعد البيانات العلائقية حيث إنها تكون المميز لجدول معين من جهة والرابط الذي يربط الجداول المختلفة مع بعضها من جهة أخرى . ويمكن تقسيم المفاتيح في قواعد البيانات العلائقية إلى عدة أقسام :

أ - **المفتاح الأعظم** (Super Key): وهو أقل مجموعة من الصفات التي يمكن أن تميز الصف في الجدول عن بقية الصفوف الأخرى .فمثلا هذه المجموعة من الصفات يمكن أن تكون مفتاحا أعظم.

St_No
St_No, St _Name
St_No ,dept_code

ب - **المفتاح المرشح** (Candidate Key) : وهو الصفة (مجموعة الصفات) التي يمكن اختيارها كمفتاح رئيس للجدول ويجب أن يكون هناك أكثر من صف له نفس القيمة لهذه الصفة أو الصفات وكذلك يجب أن يكون له قيمة (ليس Null) .

ولكن كما لاحظنا فإن St_No, St _Name هي مفتاح أعظم ولكنه ليس مفتاحا مرشحا ليكون مفتاحا رئيسا لأن St_No وحدة يكفي لتمييز أي صف عن بقية الصفوف، ،لذلك فإن St_No يعتبر مفتاحا مرشحا ليكون مفتاحا رئيسيا .

ج - **المفتاح الرئيس** (Primary Key) : وهو المفتاح الذي تم اختاره من مجموعة المفاتيح المرشحة ليكون محددا لكل صف في الجدول .يمكن أن نختار St_No ليكون مفتاحاً رئيسا .

د - **المفتاح الثانوي** : هو عبارة عن صفة أوصفات تستخدم لغايات الاسترجاع ،فمثلا لو كان لدينا جدول يحتوي على قائمة بالعملاء فالمفتاح الرئيس هو رقم العميل Customer _id

ولكن إذا أردنا أن نسترجع رقم هاتف عميل معين (ولكن من سيحفظ أرقام العملاء ؟!) ففي هذه الحالة عادة ما يستخدم الاسم في عملية البحث وليس الرقم ،فيتم اختيار اسم العميل كمفتاح ثانوي .

Customer id	Customer name	tel	Address

هـ - **المفتاح الأجنبي** (Foreign Key) : وهو صفة أو صفات تشير إلى مفتاح رئيس أو قيمة غير مكررة(Unique) في جدول آخر فمثلا تمل الصفة (Dept_Code) في جدول المتدرب (Student) مفتاح أجنبيا(Foreign Key) لجدول الأقسام (Department)

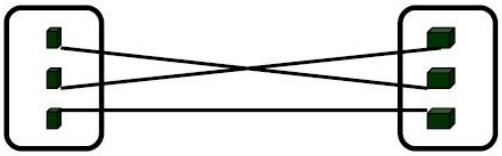
Student				
St _No	St _Name	Gpa	Birth_Date	Dept_Code
2001-01-10	Sami	3.75	01-01-1981	Comp
2001-02-99	Khalid	3.5	10-10-1982	Math
2000-01-101	Ali	4.2	12-08-1980	Comp

Department	
Dept_Code	Dept_name
Comp	Computer
Math	Mathematics

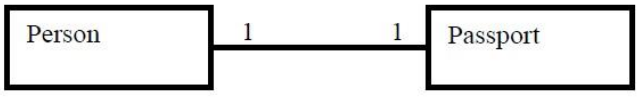
التشاركية بين الجداول (العلاقات):

وتمثل الدرجة التي ترتبط بها الجداول مع بعضها فيجب أن تحدد هذه الروابط بشكل واضح لمعرفة كيفية ارتباط هذه الجداول مع بعضها . هناك ثلاث درجات لارتباط الجداول :

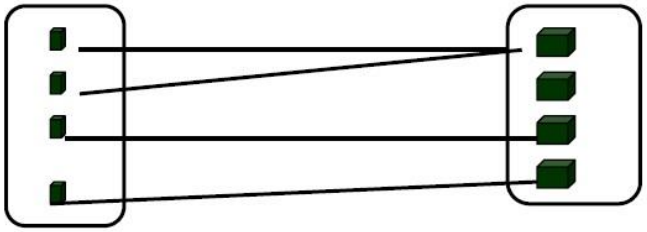
١. **واحد ــ واحد (١:١):** وهذا يعني أن قيمة واحدة في الجدول الأول تقابل قيمة واحدة فقط في الجدول الثاني



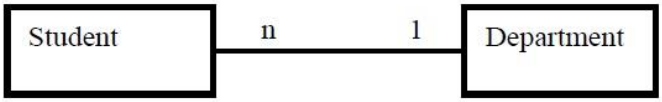
فمثلا يمكن أن نحدد على سبيل المثال أن لكل شخص جواز سفر واحد فقط وأن جواز السفر يعود لشخص واحد فقط .



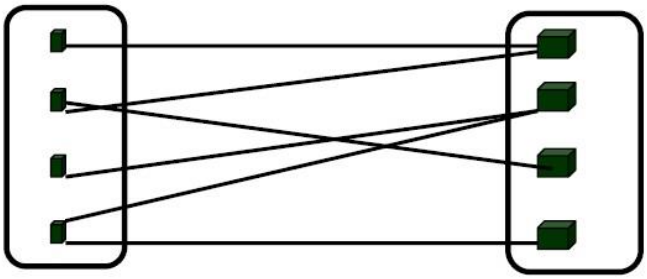
٢. **واحد ــ متعدد أو متعدد ــ واحد** (١:N أو N:١) وهذا يعني أن قيمة في الجدول الأول تقابل قيمة في الجدول الثاني وأن القيمة في الجدول الثاني يمكن أن يقابلها قيمة أو أكثر في الجدول الأول.



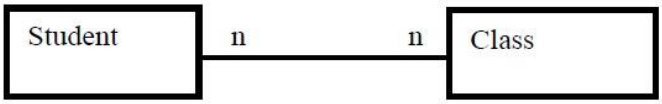
فمثلا يجب أن يتبع المتدرب لقسم واحد فقط وفي الوقت نفسه يمكن أن يكون هنالك أكثر من طالب ينتمي لهذا القسم .



٣. **متعدد ــ متعدد(N:N)** : وهذا يعني أن قيمة في الجدول الأول تقابل قيمة أو أكثر في الجدول الثانِي وأن القيمة في الجدول الثاني يمكن أن يقابلها قيمة أو أكثر في الجدول الأول.



فمثلا يمكن للطالب أن يسجل في أكثر من شعبة وكذلك الشعبة يمكن أن يسجل فيها أكثر من طالب.



لغات نظم ادارة قواعد البيانات DB Languages:

- **لغة تعريف البيانات Data Definition Language (DDL):**

هي مجموعة من الكلمات المحجوزة التي تقوم بإدارة الكائنات في قاعدة البيانات سواء بالإنشاء أو التعديل أو الحذف.

- **لغة معالجة البيانات:Data Manipulation Language (DML)**

هي عبارة عن تجمع من لغات الكمبيوتر المستخدمة من قبل برامج الكمبيوتر لمعالجة البيانات في قاعدة بيانات .

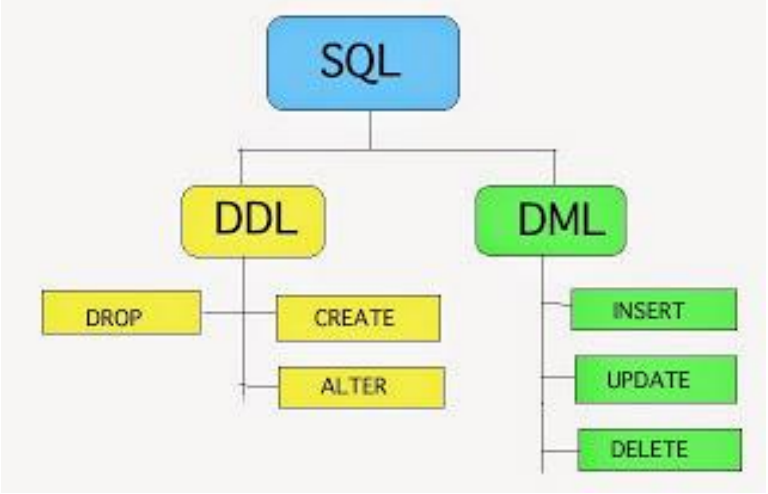
باستخدام DML ، يمكن القيام بما يأتي:

- استرجاع المعلومات المخزنة في قاعدة بيانات.
- أدخل معلومات جديدة في قاعدة البيانات.
- إزالة المعلومات من الجدول.

هناك نوعان من DML ، وهما الإجرائي وغير الإجرائي. يتطلب DML الإجرائي من المستخدم تحديد البيانات المطلوبة وكيفية الحصول عليها ، بينما يتطلب DML غير الإجرائي من المستخدم تحديد البيانات المطلوبة دون معرفة كيفية الحصول عليها.

- **SQL**

SQLاختصار لـ "لغة الاستعلام الهيكلي" (لغة الاستعلام الهيكلية). تستخدم عبارات SQL لتنفيذ العديد من المهام مثل: تحديث البيانات في قاعدة بيانات أو عرض البيانات من قاعدة بيانات. بعض برامج RDBMS التي يمكنها استخدام SQL ، مثل Oracle :و Sybase و Microsoft SQL Server و MySQL و Microsoft Accessو Ingres وما إلى ذلك. يحتوي كل برنامج قاعدة بيانات على لغة أوامر / بناء مختلفة ، ولكن من حيث المبدأ له نفس المعنى والوظيفة. تتضمن هذه الأوامر: "تحديد" و "إدراج" و "تحديث" و "حذف" و "إنشاء" ، والتي يمكن استخدامها للقيام بكل ما هو مطلوب تقريبًا لمعالجة قاعدة بيانات.



نموذج الكيانات والعلاقات ER

مقدمة:

إن هدف عملية التصميم هو الوصول إلى فهم صحيح للنظام للمساعدة في عملية تطوير هذا النظام، وهذا ليس بالأمر السهل إذ لابد من وجود مقياس صحيح للحكم على هذا الفهم. ومن هنا برزت الأهمية لاستخدام العديد من الأدوات التي تساعد المصمم لوضع التصور والفهم الصحيحين لعمل هذا النظام.ومن هذه الأدوات استخدام النماذج التمثيلية التي تصف مكونات النظام وكيفية ارتباطها مع بعضها.وسنقوم في هذا الفصل بدراسة كيفية تمثيل البيانات باستخدام **نموذج الكيانات والعلاقات** Entity Relationship (ER)Diagram.

النماذج:

ما هو النموذج ؟

النموذج عبارة عن وصف رسومي (تمثيلي) لوصف الحقائق التي لا يمكن رؤيتها مباشرة.

وبعبارة أخرى هو وصف مجرد للكائنات الحقيقة. **نموذج البيانات** هو عبارة عن تمثيل بسيط لوصف تراكيب البيانات المعقدة في واقع الحياة العملية على شكل رسومي دون النظر إلى مكان وكيفية تخزين أو الوصول إلى هذه البيانات.ويستخدم هذا النموذج كوسيلة اتصال مابين المصمم من جهة وبين المبرمجين والمستخدمين من جهة أخرى. إذ حتى لو كان لدينا العديد من المبرمجين المحترفين فلا نستطيع الحصول على نظام جيد دون أن يكون هذا النظام قد صمم بشكل صحيح .والشكل التالي يبين مواصفات لمنزل وهذا الشكل يكون كوسيلة اتصال مابين الشخص الذي يرغب في بناء المنزل (الزبون) وكذلك بين المهندس(المصمم) من جهة وبين المقاول (المنفّذ) الذي سيقوم ببناء المنزل،وفي بناء أنظمة قواعد البيانات يمثل الزبون صاحب النظام ويمثل المصمم (مصمم قاعدة البيانات) والمقاول المنفذ هو مجموعة المبرمجين التي تقوم ببناء النظام .

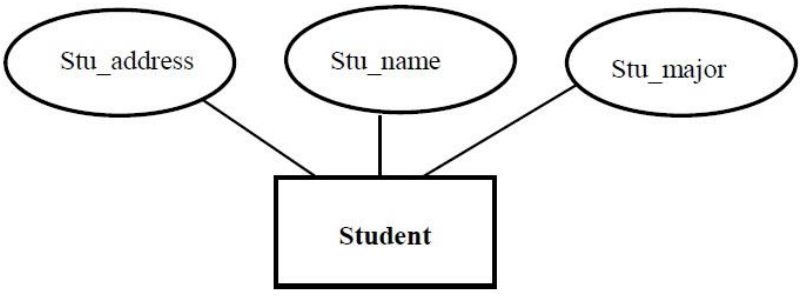
مخطط منزل

نموذج الكيانات والعلاقات:

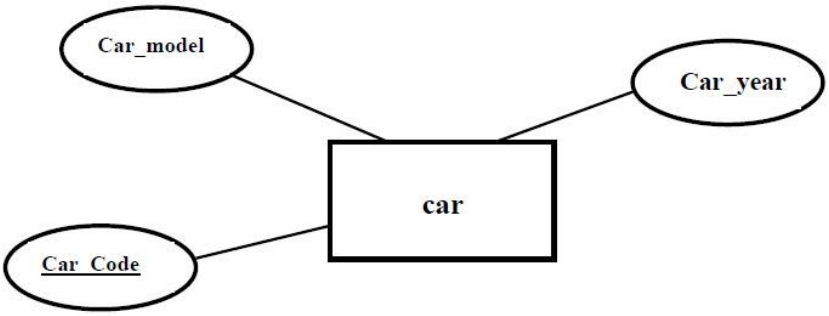
هو عبارة عن نموذج لتمثيل كيانات النظام وصفاتها وكيفية ارتباط هذه الكيانات مع بعضها باستخدام رموز رسومية..ولنتعرف الآن على عناصر هذا النموذج:

مجموعة الكيانات (Entity Set) وتمثل المجموعة التي تنتمي إليها مجموعة الكائنات(Objects) المتشابهة وتمثل بجدول في قاعدة البيانات العلائقية . و **الكيان**(Entity) هو عبارة عن كائن أو شيء محط الاهتمام في النظام وعلينا أن نقوم بجمع وتسجيل البيانات عن هذا الكيان. مثلا المتدرب ، المقرر، المدرس و الشعبة تعتبر كيانات مهمة في نظام قاعدة البيانات لجامعة .ويمثل الطبيب و المريض و وصفة العلاج كيانات مهمة في قاعدة بيانات لمستشفى . ويرمز لمجموعة الكيانات بمستطيل يحتوي على اسم الكيان .

الخصائص أو الصفات (Attributes): هي عبارة عن الصفات المميزة للكيان، وبعبارة أخرى هي المعلومات الواجب تخزينها عن كائن معين وتمثل بأعمدة الجدول في قاعدة البيانات العلائقية..فمثلا لكل طالب يجب أن نسجل الاسم، الرقم، تاريخ الميلاد، التخصص، ولمنتج معين يكون الرقم الوصف، الطول، العرض، اللون.ويرمز للصفة بشكل بيضاوي يحتوي على اسم الصفة وتربط الصفة مع الكيان بواسطة خط مستقيم.



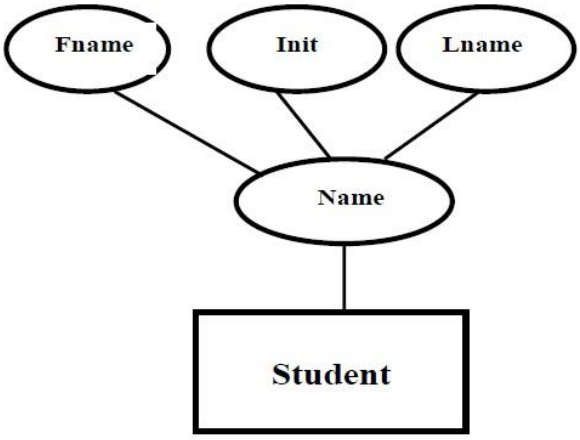
ولكل صفة يجب أن نحدد **مجال القيم (Domain)**: وهو مجموعة القيم لهذه الصفة فمثلا رقم المتدرب يجب أن يكون عدداً صحياً من عشر خانات، واسم المتدرب يجب أن يحتوي على قيم رمزية بطول ٣٠ حرف، والمعدل التراكمي يجب أن يحتوي على عدد كسري مابين ٠.. ٥ مثلا (٣,٥). تاريخ الميلاد يجب أن يكون مقبولاً بحيث لا يتجاوز عمر المتدرب عند القبول ٢٢ سنة. وبعض الصفات يمكن أن تشترك في نفس مجال القيم فمثلا القسم الدراسي للطالب والمدرس يكون اسماً من أسماء الأقسام في الجامعة. والصفة (مجموعة الصفات) التي تم اختيارها **كمفتاح رئيس (primary key)** تُمثل كأي صفة ولكن يوضع خط تحت الاسم.



وفي عملية تحديد الصفات للكيانات لابد من أن نحدد

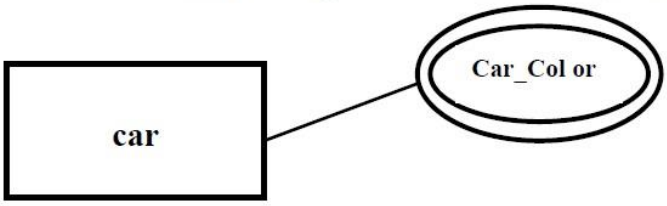
أ - الصفات البسطة والمركبة Simple and Composite Attributes :

وتقسم إلى صفات بسيطة أي لا يمكن تجزئتها مثل رقم الطالب، الجنس تاريخ الميلاد. أو مركبة أي يمكن تجزئتها كالاسم (الاسم الأول، الثاني، واسم العائلة)، العنوان (المدينة، الحي، الشارع، رقم المنزل). ويرمز للصفة المركبة بشكل بيضاوي ترتبط معه أشكال بيضاوية أخرى يحتوي كل منها على اسم الصفة الفرعية وتربط الصفات الفرعية مع الصفة الرئيسة بواسطة خط مستقيم .



ب -صفات وحيدة أو متعددة القيم Single-Valued or Multiple-Valued Attributes :

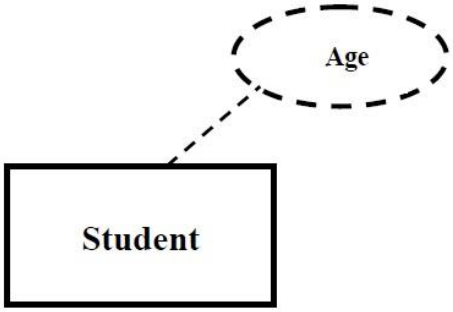
الصفات التي تحتوي على قيمة واحدة مثل (رقم السيارة، تاريخ الصنع) أو عدة قيم مثل لون السيارة (فيمكن أن يكون هناك لون للسقف، الجسم، الجوانب) وكذلك يمكن أن يكون للمدرس أكثر من رقم هاتف أو أكثر من بريد إلكتروني. ويرمز للصفة متعددة القيم بشكل بيضاوي داخل شكل بيضاوي آخر يحتوي على اسم الصفة وتربط الصفة مع الكيان بواسطة خط مستقيم.



ج ـ الصفات المشتقة (Derived Attributes):

وهي الصفات التي يمكن اشتقاقها من صفات أخرى ويرمز لها بشكل بيضاوي متقطع يحتوي على اسم الصفة وترتبط مع الكيان بخط مستقيم متقطع أيضا كما في الشكل التالي. مثل عُمر المتدرب يمكن حسابه على أنه الفرق بين تاريخ الميلاد والتاريخ الحالي.

العمر = التاريخ الحالي – تاريخ الميلاد.

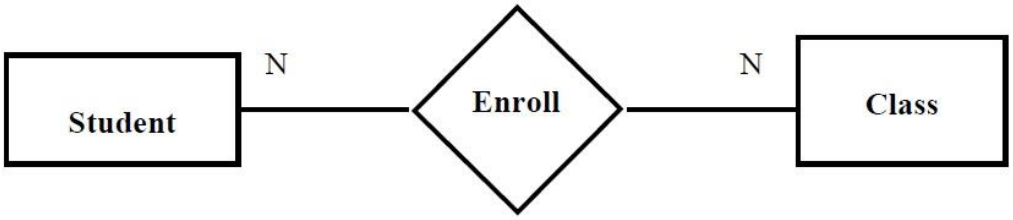


الصفات المشتقة يجب أن لا تخزن ولكن توضع طريقة لحسابها عند عملية الاسترجاع.ولكن قد نخزن بعض الصفات المشتقة إذا كانت عملية حسابها تأخذ وقتا كبيرا وفي نفس الوقت يتم طلبها بشكل كبير مثل المعدل التراكمي للطالب.

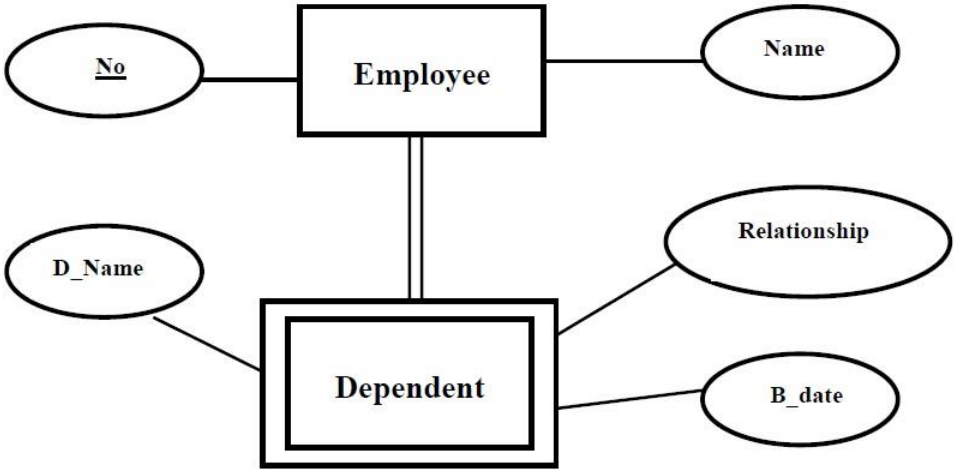
الروابط أو العلاقات (Relationships):

وهي عبارة عن الرابط أو العلاقة مابين الكيانات واسم هذه الرابطة يجب أن يعبر عن كيفية هذا الترابط ويكون على شكل فعل (ينتمي، يحتوي، يسجل، يتكون من....). ويرمز لها بشكل معين يحتوي على اسم الرابط أو العلاقة.وكذلك لكل علاقة درجة تشاركية. وتبين مقدار التشارك مابين الكيانات إما واحد ـ واحد (١:١) أو واحد ـ متعدد(**1: N**) أو متعدد ـ متعدد (**N:N**).

فالطالب يسجل في شعبة أو أكثر والشعبة يسجل فيها مجموعة من الطلاب .



الكيانات الضعيفة: وهي عبارة عن الكيانات التي لا توجد مستقلة بنفسها في النظام وبعبارة أخرى فإن وجودها يعتمد على وجود كيان آخر فمثلا لنفرض أن مؤسسة ما تسجل معلومات عن أسماء الأشخاص التابعين للموظف مثل الأبناء، الزوجة أو الوالدين. فوجود معلومات التابع مرتبط بوجود الموظف وفي هذه الحالة يختار المفتاح الرئيس للكيان الرئيس مع صفة من صفات التابع (مثل الاسم) لتشكل مفتاحا رئيسا للكيان التابع ويوضع تحته خط مقطع. ويرمز للكيان الضعيف بمستطيل داخل مستطيل يحتوي على اسم الكيان الضعيف ويرتبط مع الكيان الرئيس بخطين مستقيمين (يعني أن وجود الكيان الأول شرط لوجود الكيان الآخر وليس بالضرورة للكيانات الضعيفة فقط).

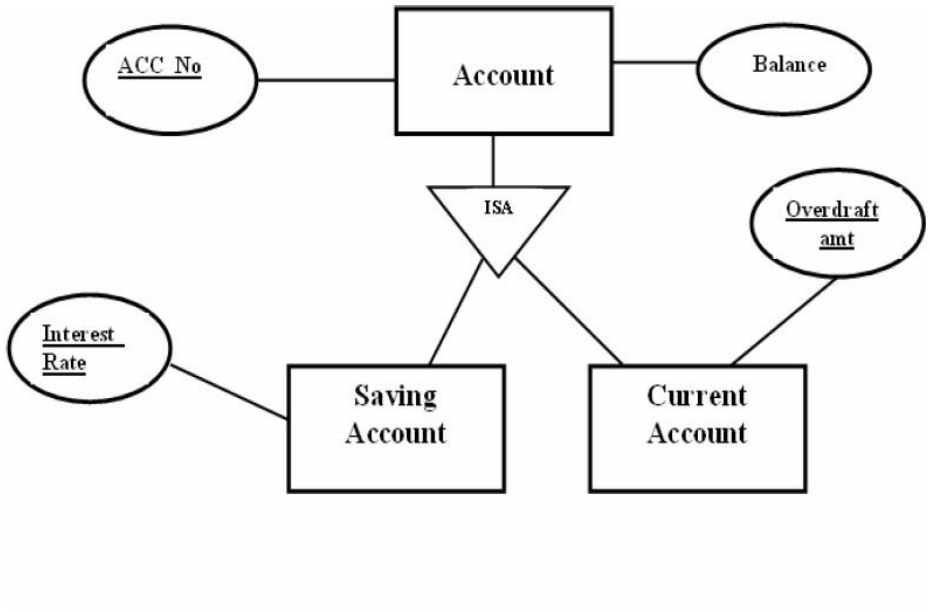


تمثيل الأنواع الرئيسة والأنواع الفرعية (Supertype and Subtype) :

هناك بعض الكيانات الفرعية التي تتبع إلى نوع رئيس (أعلى) Supertype فمثلا بالنسبة للحساب البنكي يمكن أن يكون هناك أكثر من نوع للحسابات ولكن جميع هذه الحسابات تشترك في الكثير من الصفات ففي هذه الحالة نقوم بإنشاء كيان الحساب البنكي Account بحيث يحتوي على جميع هذه الصفات ،ثم بعد ذلك نقوم بإنشاء كيانات فرعية للحسابات يحتوي كل منها على الصفات الخاصة بهذا النوع فقط.

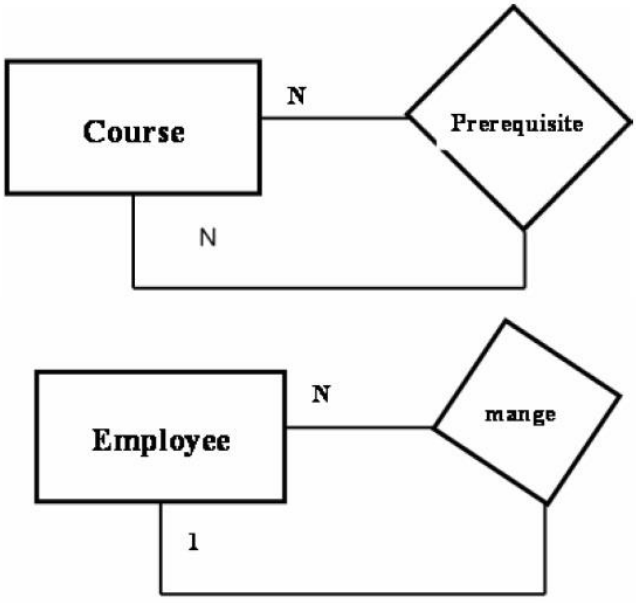
مثال: لنفرض أن لكل الحساب حقل يمثل رقم الحساب وحقل يمثل الرصيد الحالي وفي نفس الوقت لدينا نوعين من الحسابات: الحساب الجاري (Current Account) وفيه الصفة (Overdraft Amount) وهي أعلى قيمة يسمح لصاحب الحساب أن يسحبها عندما لا يكون لديه رصيد. والنوع الثاني حساب التوفير وفية صفة معدل الفائدة(Interest Rate) .

وتمثل العلاقة بين الأنواع الرئيسة العليا والأنواع الفرعية بمثلث مقلوب يحتوي على (**ISA**) بمعنى يكون.

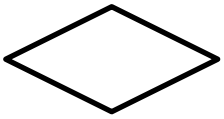
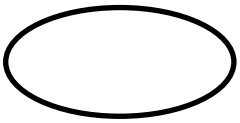
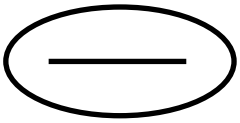
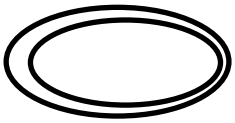
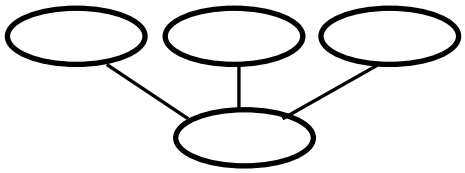


تمثيل علاقة الكيان مع نفسه (Recursive):

وفي هذه الحالة نبين كيفية تمثيل ارتباط الكيان مع نفسه، فمثلا نفرض أن المقرر الدراسي يمكن أن يكون لديه متطلب سابق أو أكثر (وهذا المتطلب هو عبارة عن مقرر) وكذلك يجب أن يكون للموظف مدير واحد فقط (والمدير بدوره هو أيضا موظف)



الرموز القياسية المستخدمة في تصميم الكيان والعلاقة الرابطة:

المفهوم	الرمز
الكيان	
العلاقة الرابطة	
الخاصية او الصفة	
صفة تمثل مفتاح اساسي	
صفة ممكن ان يكون لها اكثر من قيمة	
صفة مركبة	

التحويل من مخطط الكيان والعلاقة الرابطة ERd الى جداول

أنواع المفاتيح في الجداول

❑ **المفتاح الرئيسي :**

هو صفة مميزة في جدول الكيان يتميز بأن قيمته فريدة ولا يمكن أن تتكرر أو تكون خالية .

مثال : الطالب : (اسم الطالب , الرقم الجامعي ، الهاتف)

❑ **المفتاح الأجنبي :**

هو صفة موجود في جدول لكنه لا يمثل واحدة من صفاته .

و يسمى مفتاح أجنبي لأنه يمثل مفتاحاً رئيسياً في جدول آخر .

وظيفته الربط بين الجداول .

الطالب : (اسم الطالب , الرقم الجامعي ، الهاتف، رقم القسم)

تتم عملية التحويل من مخطط الكيان و العلاقة الرابطة ERD إلى جداول ، بتطبيق مجموعة من الخطوات البسيطة، تسمى Mapping .

التحويل من مخطط الكيان و العلاقة الرابطة ERD إلى جداول :

❖ الخطوة الأولى :

1- كل كيان في نموذج ERD يتحول إلى جدول ، ويكون اسم الجدول بنفس اسم الكيان .

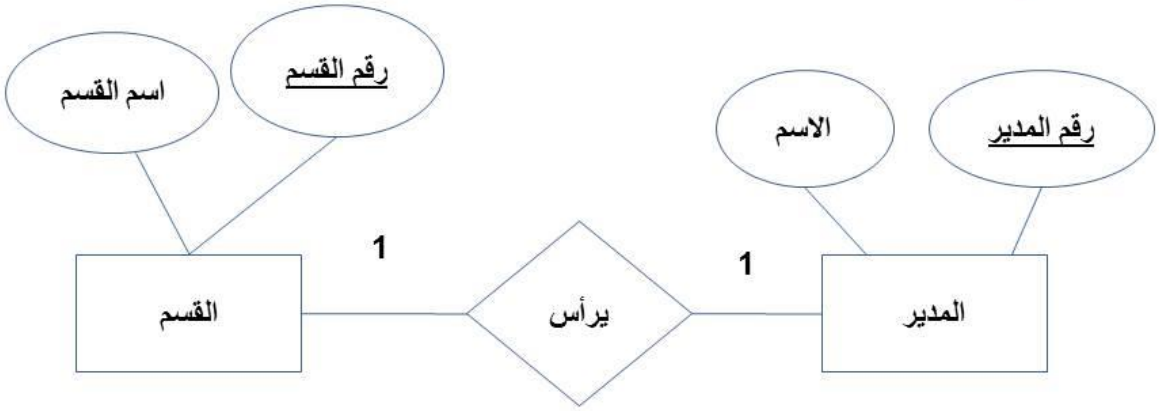
2- خصائص الكيان تصبح هي حقول الجدول أي عناوين أعمدته ،

ولا ننسى أن نضع خطاً تحت المفتاح الأساسي .

نضع اسم الجدول ثم أسماء الحقول بين قوسين كالتالي :

اسم الكيان (صفة 1 , صفة 2 مفتاح رئيسي ، صفة 3 ، ...)

مثال : حول نموذج الكيان والعلاقة الرابطة التالي إلى جداول:



أولا :

المدير (رقم المدير , الاسم)

القسم (رقم القسم , اسم القسم)

التحويل من مخطط الكيان و العلاقة الرابطة ERD إلى جداول :

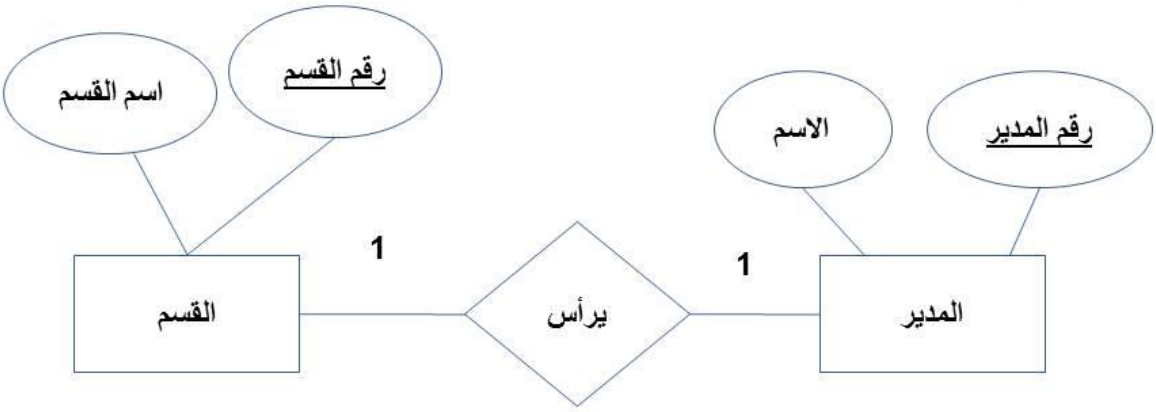
❖ الخطوة الثانية :

الربط بين هذه الجداول عن طريق المفاتيح ويكون ذلك حسب العلاقات :

- الحالة الأولى عندما تكون العلاقة بين الكيانين هي واحد إلى واحد :

فأننا عند تحويلها إلى جداول فإن المفتاح الأساسي لأحد الجدولين يظهر كحقل إضافي عند الجدول الآخر – تحديد المفتاح اختياري- ، ويسمى مفتاح أجنبي لأنه مفتاح أساسي لجدول وظهر في جدول آخر ونضع تحته خط متقطع .

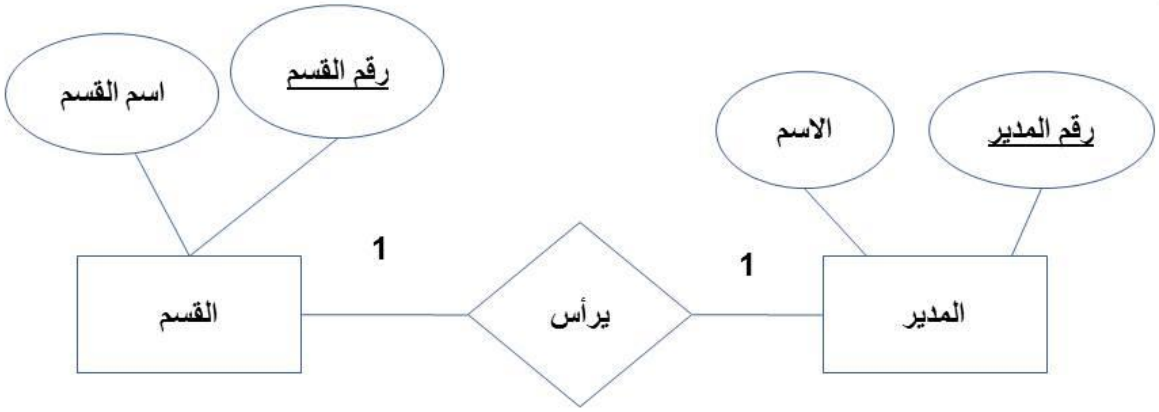
مثال : حول نموذج الكيان والعلاقة الرابطة التالي إلى جداول:



أولا :

المدير (رقم المدير , الاسم)

القسم (رقم القسم , اسم القسم)



- ثانيا اربط الجدولين بعلاقة واحد إلى واحد , وتمثل كما يلي :

المدير (رقم المدير, الاسم , **رقم القسم**)

القسم (رقم القسم, اسم القسم)

أو العكس خياري :

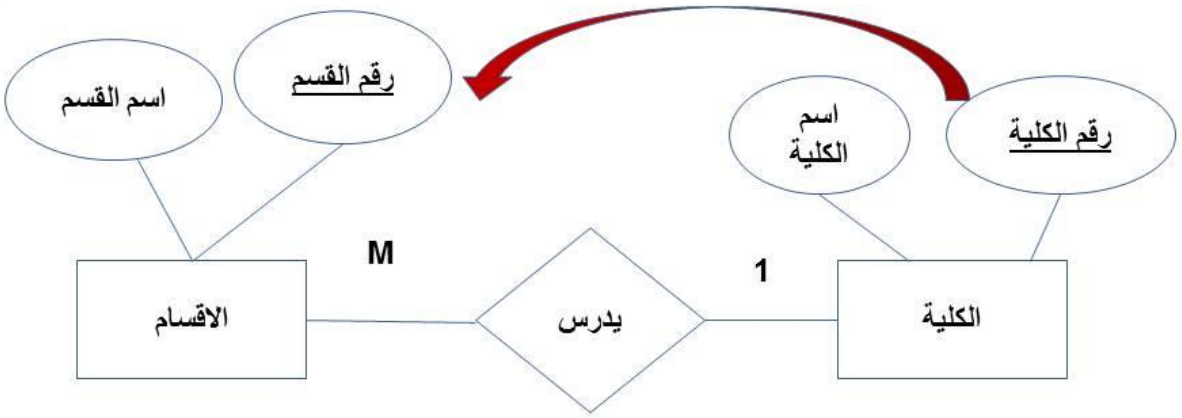
المدير (رقم المدير,الاسم)

القسم (رقم القسم, اسم القسم, **رقم المدير**)

التحويل من مخطط الكيان و العلاقة الرابطة ERD إلى جداول :

- الحالة الثانية :عندما تكون العلاقة بين الكيانين هي واحد إلى متعدد :

فإننا عند تحويلها إلى جداول نأخذ المفتاح الأساسي للكيان الذي تكون العلاقة من جهته واحد ونضع نسخه منه كمفتاح أجنبي للكيان التي تكون العلاقة من جهته متعدد ويكون ذلك إجباري .



أولا : نحول الكيانات إلى جداول :

الكلية (رقم الكلية, اسم الكلية)

الاقسام (رقم القسم, اسم القسم)

ثانيا : نربط بين الجدولين :

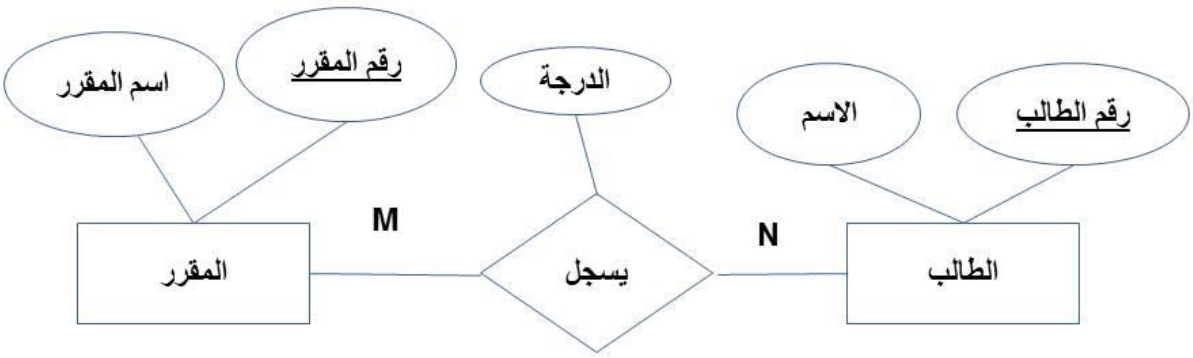
الكلية (رقم الكلية, اسم الكلية)

القسم (رقم القسم, اسم القسم , **رقم الكلية**)

التحويل من مخطط الكيان و العلاقة الرابطة ERD إلى جداول :

❖ الحالة الثالثة :عندما تكون العلاقة بين الكيانين هي متعدد إلى متعدد :

لابد من تعريف جدول ثالث يسمى جدول الربط يسمى بأسم العلاقة و يتكون مفتاحه الأساسي من حقلين عبارة عن (المفتاحين الأساسين للجدولين المرتبطين) وقد يحوي حقول أخرى ضرورية .



أولا : نحول الكيانات إلى جداول :

الطالب (رقم الطالب, الاسم)

المقرر (رقم المقرر, اسم المقرر)

ثانيا : نربط بين الجدولين :

الطالب (رقم الطالب, الاسم)

المقرر (رقم المقرر, اسم المقرر)

التسجيل (رقم الطالب،رقم المقرر, الدرجة)

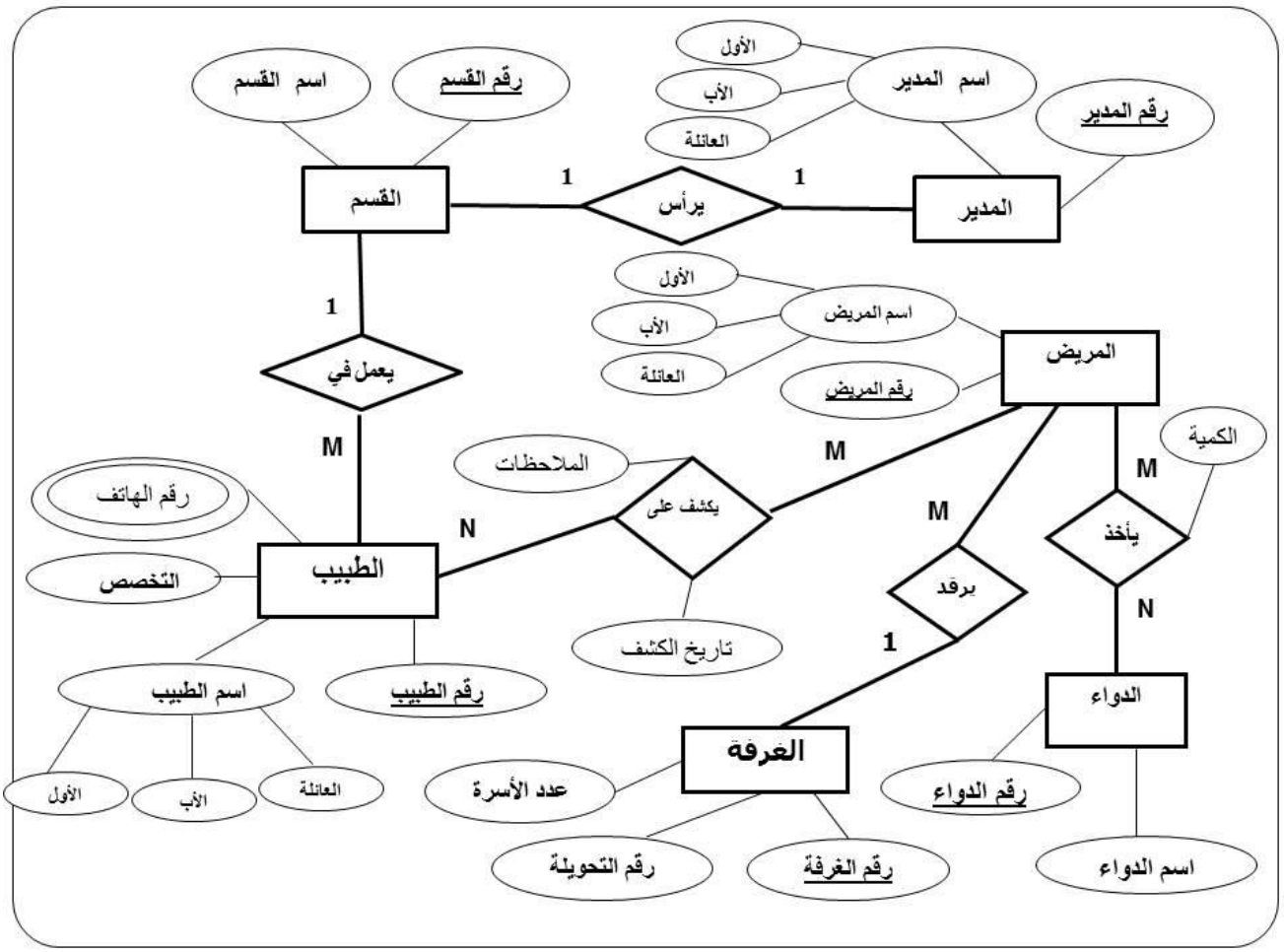
التحويل من مخطط الكيان و العلاقة الرابطة ERD إلى جداول :

❖ الخطوة الثالثة :

تحويل الصفة متعددة القيم يكون بإنشاء جدول يحتوي على حقل لهذه الصفة مع المفتاح الأساسي للكيان الذي تمثله تلك الصفة .

مثال : الهاتف (رقم الاستاذ ، رقم الهاتف)

الصفة المشتقة تلغى من الجدول ويحصل عليها عن طريق الاستعلامات لاحقا .



1. المدير (رقم المدير, الاسم الأول ،اسم الأب ، اسم العائلة)
2. القسم (رقم القسم ,اسم القسم, رقم المدير)
3. الطبيب (رقم الطبيب, الاسم الأول ، اسم الأب ، اسم العائلة , التخصص , رقم الهاتف, رقم القسم)
4. المريض (رقم المريض ،الاسم الأول ، اسم الأب ،اسم العائلة , رقم الغرفة)
5. الغرفة(رقم الغرفة, عدد الأسرة , التحويلة)
6. الدواء(رقم الدواء ,اسم الدواء)
7. الوصفة (رقم المريض , رقم الدواء , الكمية)
8. المعالجة (رقم المريض ، رقم الطبيب , تاريخ الكشف , الملاحظات)
9. الهاتف (رقم الطبيب ، رقم الهاتف)

لغة SQL وماذا تقدم:

SQL : هي اختصارٌ لـ Structured Query language وترجمتها هي "لغة الاستعلام البنائية" لغة SQL هي لغة ذات غرض متخصص هدفها إعطاء القدرة على إدارة البيانات الموجودة في قواعد البيانات العلاقية والتعامل معها، وتخضع هذه اللغة لمعايير دولية متفق عليها، ويقوم المعهد الوطني الأمريكي للمعايير (ANSI) بإدارة وإصدار المعايير الخاصة بـ SQL .

نستطيع باستخدام لغة SQL أن تقوم بالاتي:

١. الاستعلام عن البيانات وجلبها من قاعدة البيانات.
٢. إضافة، تعديل السجلات في قاعدة البيانات وحذفها منها.
٣. الحفاظ على سلامة ودقة البيانات في قاعدة البيانات.
٤. تحديد الصلاحيات والأذونات الخاصة بمستخدمي قاعدة البيانات.

أنواع أوامرSQL :

تنقسم جمل وأوامر SQL إلى ثلاث مجموعات، وذلك حسب الدور الذي يقوم به الأمر:

- **لغة التعامل مع البيانات :Data Manipulation Language (DML)** تحتوي هذه المجموعة على جمل غرضها إعطاء القدرة على التعامل مع البيانات دون التأثير على هيكليتها وشكلها العام، بحيث تستطيع الاستعلام عن البيانات، إضافة سجلّات، حذفها أو تعديلها.
- **لغة تعريف البيانات :Data Definition Language (DDL)**تُقدم الأوامر التي تندرج تحت هذه المجموعة القدرة على تعريف البيانات وشكلها وطريقة ربطها ببعضها عبر استخدام أوامر لإنشاء الجداول وإنشاء قاعدة البيانات.
- **لغة التحكم بالبيانات :Data Control Language (DCL)**تساعد هذه المجموعة من الأوامر في تحديد الصلاحيات التي يمكن منحها أو سلبها من المستخدمين الموجودين في قاعدة البيانات.

يبين الجدول التالي أهم الأوامر التي تُمَكّن المبرمج من إنشاء قاعدة البيانات الخاصة به وتنفيذ الجمل الخاصة بإدارتها وإدارة بياناتها.

عنوان الأمر	الأمر
إنشاء قاعدة بيانات	CREATE DATABASE
تعديل قاعدة البيانات	ALTER DATABASE
إنشاء جدول	CREATE TABLE
تعديل الجدول	ALTER TABLE
حذف الجدول	DROP TABLE
إنشاء فهرس	CREATE INDEX
حذف فهرس	DROP INDEX
الاستعلام عن البيانات	SELECT
إضافة بيانات	INSERT
تعديل بيانات	UPDATE
حذف بيانات	DELETE

جملة إنشاء قاعدة البيانات

في الواقع لا يوجد معيار لأمر إنشاء قاعدة البيانات في معيار SQL المقدم من معهد ANSI ولكن برمجيات قواعد البيانات العلاقية تقدم نسخة من هذا الأمر، وتختلف الصيغة من نظام إلى آخر.

الصيغة العامة لأمر إنشاء قاعدة البيانات هي كالتالي:

```
CREATE DATABASE database_name;
```

ملاحظات هامة

- يُنشئ الأمر السابق قاعدة بيانات فارغة بالاسم المُمَرَّر (أي database_name) في الجملة أعلاه.
- تتطلب أغلب نظم إدارة قواعد البيانات وجود صلاحيات المسؤول للمستخدم الذي ينشئ قاعدة البيانات.
- بمجرد إنشاء قاعدة البيانات، يستطيع المستخدم أو من له صلاحية، البدء بإضافة عناصر إلى قاعدة البيانات من جداولTables ، مشاهدViews ، دوال وحزم، وإضافة سجلات وبيانات إلى الجداول المُنشأة.
- أثناء تنفيذ أمر إنشاء قاعدة البيانات وبعده، تُنشَأ ملفات خاصة بقاعدة البيانات الجديدة حسب النظام المستخدم، وتُدار هذه الملفات وتُسمَّى إما تلقائيًّا أو من قبل المستخدم.

- لأن أمر إنشاء قاعدة البيانات غير معياري، فإن خيارات هذا الأمر متعددة وكثيرة وتأتي حسب نوع النظام المستخدم، وحسب نوع نظام إدارة قواعد البيانات.
- يظهر أمر الإنشاء بأحرف كبيرة .Upper case ليس هذا ضروريّاً في أغلب برامج إدارة قواعد البيانات، إلا أنها عادة في التوثيقات Documentations لتمييز الكلمات المفتاحية التي تعدّ جزءًا من SQL.

جملة إنشاء جدول:

يُعد إنشاء الجدول في قاعدة البيانات أول الخطوات في طريق بناء قاعدة البيانات وملئها بالسجلات، وهذا الأمر يقوم به مسؤول قواعد البيانات أو المبرمج على حد سواء.

ستحتاج قبل أن تتمكّن من البدء في تنفيذ أوامر على القاعدة إلى تحديد قاعدة البيانات التي تريد العمل عليها، أي تلك التي ستُنشِئ الجداول فيها. تختلف طريقة تحديد قاعدة البيانات حسب طريقة الاتصال ببرنامج إدارة قواعد البيانات:

- في سطر أوامر MySQL يُنفَّذ الأمر بالطريقة التالية:

```
USE  database_name;
```

الصيغة العامة لجملة إنشاء الجدول في:SQL

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

شرح الصيغة:

```
CREATE  TABLE table_name (
```

- CREATEتعني إنشاء العنصر (الجدول هنا) وهي بداية الأمر.
- TABLEلتحديد أن هذه الجملة لإنشاء جدول.
- table_nameوهو الاسم الذي نريد إطلاقه على الجدول الجديد الذي نريد بناءه.

- القوس المفتوح باتجاه اليسار يعني البدء بكتابة هيكل الجدول والذي يتضمن الأعمدة ونوعها والقيود التي من الممكن أن نضيفها وبعض الإعدادات الأخرى.

```
column1 datatype [constraint],
```

- column1: هو الاسم الذي سوف نعطيه للعمود الأول.
- Datatype: يعني نوع العمود (نصي, رقم, تاريخ. الخ.
- [Constraint]: تعني – اختيارياً - تستطيع تحديد قيود على مستوى هذا العمود.
- الفاصلة تعني وجود عمود آخر سوف نعرّفه بعد هذا العمود.
- عند كتابة العمود الأخير لا نضيف فاصلة، ومن ثم نضيف القوس المعاكس للقوس الذي فُتح عند بداية كتابة الأعمدة، ونختم الأمر بفاصلة منقوطة.

ملاحظات هامة:

- تبدأ أسماء الجداول والأعمدة عموما بحرف وليس برقم ، ومن الممكن أن تُتبع بعد ذلك بالأرقام.
- يُفضَّل ألا يتجاوز طول اسم الجدول أو العمود ٣٠ حرفاCharacters ، حيث إن بعض النظم تمنع أن تتجاوز ذلك مثل نظام إدارة قواعد البيانات أوراكل.
- يجب ألا تُستخدَم كلمات محجوزة في تسمية الجدول أو العمود.

جملة تعديل الجدول:

بعد أن تكلمنا عن جملة إنشاء الجدول، يجب علينا توضيح جملة تعديل الجدول بعد إنشائه، حيث إن إجراء عمليات التغيير على الجدول يُعد أمراً مهما للمبرمج ومسؤول قواعد البيانات إذ يُنفَّذ باستمرار أثناء وفي بداية بناء النظم البرمجية والبرامج نظرا لتغير المتطلبات وعدم اكتمالها.

تُستخدَم جملة تعديل الجدول Alter Table عموما في الحالات التالية:

- إضافة عمود للجدول.
- حذف عمود من الجدول.
- تغيير نوع عمود في الجدول.
- إضافة قيد على العمود.
- حذف قيد عن العمود.

في ما يلي الصيغ العامة لجملة تعديل الجدول.

إضافة عمود:

```
ALTER TABLE table_name ADD column_name datatype;
```

حذف عمود:

```
ALTER TABLE table_name DROP COLUMN column_name;
```

تعديل عمود:

```
ALTER TABLE table_name MODIFY | ALTER COLUMN column_name
datatype;
```

إنشاء جداول وتعديلها:

لإنشاء جدول باسم Persons يحتوي على ٥ أعمدة تمثل معلومات أشخاص مثل رقم الشخص واسمه وعنوانه، نُنَفذ الجملة التالية:

```
CREATE TABLE Persons (
   PersonID int,
   Last_Name varchar(255),
   First_Name varchar(255),
   Address varchar(255),
   City varchar(255)
);
```

يتكوّن الجدول السابق:

- من العمود PersonID الذي هو من النوعint ، أي أن قيم هذا العموم يجب أن تكون أرقامًا؛
- الأعمدةCity ،Last_Name ، First_Name وAddressالتي هي من النوعvarchar ، أي سلسلة محارف، بطول 255 محرفا.

بعد تنفيذ جملة إنشاء الجدول السابقة، ينتج لدينا جدول فارغ بالشكل التالي:

```
+-------------+-----------------+-----------------+------------+--------+
| PersonID    | Last_Name       | First_Name      | Address    | City   |
+-------------+-----------------+-----------------+------------+--------+
```

نستطيع أن نملأ الجدول بالبيانات باستخدام جملة الإضافة والتي سوف نتكلم عنها لاحقاً.

لإنشاء نفس الجدول السابق بحيث يتضمن وجود قيود على مستوى الأعمدة، ننفذ الجملة التالية:

```
CREATE  TABLE  Persons (
   PersonID int PRIMARY KEY,
   LastName varchar(255) NOT NULL,
   FirstName varchar(255),
   Address varchar(255) NOT NULL,
   City varchar(255)
);
```

أضفنا في الجملة السابقة، قيودا على مستوى أعمدة الجدول، بحيث يُعرَّف العمود PersonID بأنه المفتاح الرئيسي للجدول، والأعمدة Last_Name و Address بأنها لا تستقبل القيم الفارغة.

في حال أردنا أن نضيف عمودًا جديدًا للجدول باسم) Age العمر) ومن نوع البيانات رقم نستخدم جملة التعديل التالية:

```
ALTER TABLE Persons
ADD Age int;
```

تمكن ترجمة الأمر على النحو التالي" :**عدّل الجدول Persons بإضافة عمود اسمه Age ونوعه.**"**int**

في حال أردنا أن نحذف عمود City من الجدول نستخدم الجملة التالية:

```
ALTER TABLE Persons
DROP COLUMN City;
```

أي" :**عدّل الجدول Persons بحذف العمود.**"**City**

إذا أردنا تعديل نوع عمود Age إلى نص بدلا من رقم نستخدم الجملة التالية:

```
ALTER  TABLE Persons
MODIFY  Age varchar(10);
```

التعامل مع البيانات (الإدخال، الحذف والتعديل) فيSQL :

بعد أن تعلمنا كيفية إنشاء الجدول في قواعد البيانات، وتعرفنا على أنواع البيانات المستخدمة غالبا، وكيفية إضافة القيود على الجدول، سوف نقوم بالتعرف على جمل التعامل مع البيانات Data Manipulation Language، بحيث ستكون لدينا المعرفة اللازمة لإضافة سجل بيانات على الجدول، تعديل سجل بيانات، وحذف سجل بيانات وذلك باستخدام جمل SQL اللازمة لذلكUPDATE:، INSERTو.DELETE

تغيّر جمل التعامل مع البيانات على البيانات في الجدول، لذلك فإن مسؤول قواعد البيانات لا بدّ أن يدير الصلاحيّات اللازمة للتغيير على البيانات بطريقة مُثلى. لابد التنويه إلى أن جمل التعامل مع البياناتUPDATE) ، INSERTو (DELETEلا تُرجع بيانات أو مُخرجات عند تنفيذها، ولكن أغلب أنظمة إدارة قواعد البيانات تطبع لك ملخص السجلات التي تأثرت بتنفيذ الجمل السابقة.

هيكلية الجدول:

قبل البدء بشرح جمل التعامل مع البيانات ومعرفة كيفية تنفيذها، لابد أن تكون لدينا المعرفة الكاملة بهيكلية الجدول الذي نريد إضافة سجل إليه أو تعديله أو الحذف منه، والتعرف على هيكلية الجدول يتضمن:

- معرفة ترتيب الأعمدة الموجودة في الجدول،
- معرفة أسماء الأعمدة،
- معرفة نوع البيانات الخاصة بكل عمود،
- التعرف على الأعمدة المطبق عليها قيود ومعرفة هذه القيود وطبيعتها،
- معرفة القيم المبدئية إذا وجدت.

تُقدم أغلب نظم إدارة قواعد البيانات الأمر اللازم لمعرفة هيكلية الجدول، والصيغة العامة لهذا الأمر تكون كالتالي:

```
DESCRIBE table_name;
```

أو تكون بالصيغة التالية:

```
DESC table_name;
```

جُملة إضافة السجلات:

توجد طريقتان لإضافة سجلات البيانات على الجدول في قاعدة البيانات، الأولى تتضمن كتابة أسماء الأعمدة التي نريد إعطاءها قيمًا بالإضافة للقيم نفسها التي نريد حفظها، وتسمى هذه الطريقة **الإضافة بأسماء الأعمدة**، والطريقة الثانية، في حال أردنا أن نضيف بيانات لكل الأعمدة في الجدول بنفس الترتيب الموجود في الجدول، فهنا لا نكتب أسماء الأعمدة، وتسمى هذه الطريقة **الإضافة بمواقع الأعمدة**.

الصيغة العامة للطريقة الأولى:

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

الصيغة العامة للطريقة الثانية:

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

لنفترض وجود جدول Persons بالهيكلية والبيانات التالية:

Person_ID	First_Name	Last_Name	Age
101	Ibrahim	Mohammed	31
102	Mohammed	Khaled	25

لإضافة سجل إلى هذا الجدول نستخدم الجملة التالية:

```
INSERT  INTO  Persons
VALUES (103, 'Saleem', 'Yaser',20);
```

ملاحظة :تُكتَب النصوص في بعض أنظمة قواعد البيانات بين علامتي التنصيص المزدوجة. فمثلا لكتابة اسم Ibrahimفي جملة الإضافة نكتبها هكذا"Ibrahim" ، بينما توجد أنظمة أخرى تعتمد التعامل مع النصوص بين علامتي تنصيص منفردة، هكذا.'Ibrahim'

في جملة إضافة السجل السابقة لم نذكر اسم أي من الأعمدة، لأننا أدخلنا قيمًا لكل الأعمدة. القيمة الأولى (103) هي للعمود الأول (Person_ID) والثانية ('Saleem') للعمود الثاني (First_Name) وهكذا. ولكن في بعض الحالات لا نضيف السجل بهذه الطريقة، ففي كثير من الأحيان نحتاج إلى إضافة قيم لأعمدة معينة.

مثال على تحديد الأعمدة:

الأعمدة التي يكون عليها قيد مفتاح رئيسي غالبا لا تُدخَل قيم إليها في جمل الإضافة، ويُستخدم بدلاً لذلك طرق لتوليد قيم لها مثل طريقة جعل العمود (ذي النوع الرقمي Auto Increment (الذي يزيد آخر رقم في العمود بالعدد واحد ثم يستخدمه للعمود الذي يوجد عليه المفتاح الرئيسي. في هذه الحالة تكون جملة الإضافة بالصيغة التالية:

```
INSERT INTO Persons (first_name, last_name, age)
VALUES ('Ali',Mohammed',25);
```

ويصبح الجدول بعد تنفيذ الجملتيْن السابقتيْن كالتالي:

Person_ID	First_Name	Last_Name	Age
101	Ibrahim	Mohammed	31
102	Mohammed	Khaled	25
103	Saleem	Yaser	20
104	Ali	Mohanmmed	25

القيم الفارغة:

لو افترضنا وجود قيد العمود غير الفارغ على الأعمدة First_Name و Ageمع وجود قيد المفتاح الرئيسي على العمود Person_Id وخاصية Auto Increment عليه، فإن أي جملة إضافة لا تشتمل على قيم لـ First_Nameأو Age سوف تُظهِر خطأ عند تنفيذها، ولكن لأننا لم نضف قيد العمود غير الفارغ على عمودLast_Name، فإننا نستطيع تجاهل هذا العمود عند الإضافة كالتالي:

```
INSERT  INTO  Persons (first_name, age)
VALUES ('Reem', 20);
```

تصبح البيانات في الجدول كالتالي عند تنفيذ الجملة السابقة:

Person_ID	First_Name	Last_Name	Age
101	Ibrahim	Mohammed	31
102	Mohammed	Khaled	25
103	Saleem	Yaser	20
104	Ali	Mohanmmed	25
105	Reem		20

من المهم فهم طبيعة القيم الفارغة Null في جداول قواعد البيانات، حيث إنها تختلف عن قيمة الصفر أو القيم النصية ."""القيم الفارغة هي التي تُركت بدون اعتبار عند إضافة السجل أو تعديله، كما أن القيم الفارغة لا تخضع للفحص أو المقارنة باستخدام عمليات مثل<> > < = ؛ ونستخدم بدلا منها عمليات IS NULLأو IS NOT NULL هذه النقطة سوف نشرحها في درس جملة الاستعلام.

جملة تعديل السجل:

نستخدم جملة تعديل السجل لإجراء عملية تغيير لقيم الأعمدة في سجل معين أو مجموعة سجلات أو على الجدول بأكمله.

الصيغة العامة لجملة تعديل السجل كالتالي:

```
UPDATE  table_name
SET  column1 = value1, column2 = value2, ...
WHERE  condition;
```

ملاحظات هامة:

- في حالة تعديل أكثر من عمود، يجب وضع فاصلة بين القيم الجديدة للأعمدة.
تُمثلcolumn1 و column2أسماء الأعمدة التي نريد تغيير قيمها، وتمثل value1 و value2 القيم الجديدة للأعمدة السابقة بالترتيب.
- نستطيع التعديل على أكثر من عمود في نفس جملة التعديل.
- لا بد من الحذر والانتباه الشديديْن عند تنفيذ جملة التعديل، حيث إن جملة الشرط (WHERE condition)تحدد السجلات التي سيُعدَّل عليها، وفي حالة عدم وجود جملة الشرط، فإن جميع السجلات في الجدول ستدخل في العملية.

تعديل سجل واحد:

لتعديل سجل واحد، يجب أن نُحدد الشرط الذي يميز هذا السجل على نحو فريد في جملة الشرط، وغالبا يُستخدَم عمود قيد المفتاح الرئيسي في الجدول. فمثلا، إذا أردنا أن نُعدل قيمة العمر Age من 31 إلى 21 للشخص صاحب الرقم101 في جدول Persons ننفذ الجملة التالية:

```
UPDATE  Persons
SET  Age = 21
WHERE  Person_Id = 101;
```

ويصبح السجل بالقيم التالية:

Person_ID	First_Name	Last_Name	Age
101	Ibrahim	Mohammed	21

لاحظ أننا في جملة الشرط استخدمنا عملية المقارنة = لتحديد رقمPerson_Id ، وهنا يجب أن ننوه أننا نستطيع استخدام جميع عمليات المقارنة في جملة الشرط بشرط أن تكون منطقية ومكتوبة بطريقة صحيحة، فمثلا، لو أردنا أن نُعدل جميع أعمار الأشخاص الذين أعمارهم 20 أو أقل، لتصبح 25 ننفذ الجملة التالية:

```
UPDATE  Persons
SET  Age = 25
WHERE  Age <= 20;
```

السجلات التي لها القيمة 103 و 105في العمود Person_Id هي التي ستتأثر بالجملة السابقة عند تنفيذها، وتصبح البيانات في الجدول على النحو التالي:

Person_ID	First_Name	Last_Name	Age
101	Ibrahim	Mohammed	21
102	Mohammed	Khaled	25
103	Saleem	Yaser	25
104	Ali	Mohanmmed	25
105	Reem		25

جملة حذف السجل:

تُستخدَم جملة الحذف لإجراء عملية مسح بيانات موجودة في جدول معين وتتحكّم فيها جملة الشرط والتي تحدد السجل أو السجلات التي ستُحذَف.

الصيغة العامة لجملة الحذف:

```
DELETE  FROM  table_name
WHERE  condition;
```

لابد من الانتباه عند تنفيذ جملة الحذف، حيث إنه في حالة عدم تحديد جملة الشرط، فإن كل البيانات في الجدول ستُحذَف.

حذف سجل واحد:

لحذف السجل الخاص بالشخص الذي رقمه ١٠٣ ننفذ الجملة التالية:

```
DELETE  FROM Persons
WHERE  Person_Id = 103;
```

لحذف السجل الخاص بالشخص ذي القيمة الفارغة في الحقل Last_Name ننفذ الجملة التالية:

```
DELETE  FROM Persons
WHERE  Last_Name IS NULL;
```

حذف بيانات الجدول:

لحذف البيانات بالكامل من الجدول Persons ننفذ إحدى الجملتيْن التاليتيْن:

```
DELETE  FROM Persons;
DELETE * FROM Persons;
```

ملاحظة :العلامة * تعني جميع السجلات .

الاستعلام عن البيانات فيSQL

تعرّفنا في الدروس السابقة على إنشاء الجدول في قاعدة البيانات وإضافة البيانات إليه والتعامل معها من حيث التعديل والإضافة. سوف نبدأ بالتعرف على أشهر جمل لغة الاستعلام البنائية، وهي جملة الاستعلام Select Statement، حيث سنتكلم عن كيفية كتابة جملة الاستعلام، وأشكالها، وكيفية ترشيح البيانات وتحديد الأعمدة التي نريدها وغيرها من المواضيع.

جملة الاستعلام:

تجلب جملة الاستعلام SELECT بيانات جدول أو أكثر بعد الاستعلام عن وجود هذه الجداول في قاعدة البيانات، ونقصد بالاستعلام هنا ماذا نريد؟ ومن أين؟

ماذا نريد من أعمدة وسجلات، ومن أين، أي من أي الجداول نأتي بالمعلومات. البيانات الناتجة عن تنفيذ جملة الاستعلام تسمى مجموعة البيانات الناتجة.Result Data-Set

الصيغة العامة لجملة الاستعلام:

```
SELECT column1, column2, ...
FROM table_name
[WHERE where_condition]
[GROUP BY group_by_expression]
[ORODER BY order_by_expression];
```

في بداية كل جملة استعلام نكتب كلمة) SELECT جمل SQL غير حساسة لحالة الأحرف) ومن ثم نُتبعها بأسماء الأعمدة التي نريد الاستعلام عنها، أو نستبدل أسماء الأعمدة برمز * والذي يعني كل الأعمدة، ثم نكتب كلمة From والتي يليها اسم الجدول أو أسماء الجداول التي تحتوي على البيانات التي نريدها.

ما بين الأقواس المعكوفة هي جمل إضافية تقوم بمهام معينة في جملة الاستعلام وهي كالتالي:

Where: هي جملة الشرط والتي ترشّح البيانات بناءً على الشرط الموجود بعدها.

Group By: تجمّع البيانات الناتجة من تنفيذ جملة الاستعلام بناءً على جملة التجميع التي تليها.

Order By: ترتّب البيانات تصاعديا أو تنازليا بناءً على جملة الترتيب التي تليها.

مثال على جملة الاستعلام:

لو أردنا الاستعلام عن كافة البيانات الموجودة في جدول Persons ، ننفذ الجملة التالية:

```
SELECT *
FROM Persons;
```

، فإن * تعني عرض جميع البيانات، حيث ستُظهر كافة الأعمدة الموجودة في الجدول والسجلات التي يحتويها. وسيكون ناتج الجملة البيانات التالية:

Person_ID	First_Name	Last_Name	Age
101	Ibrahim	Mohammed	21
102	Mohammed	Khaled	25
103	Saleem	Yaser	25
104	Ali	Mohammed	25
105	Reem		25

لو أردنا أن نستعلم عن اسم الشخص الأول وعمره، نقوم بتنفيذ الجملة التالية:

```
SELECT  First_Name, Age
FROM  Persons;
```

لاحظ أننا فصلنا بين الأعمدة التي نريد إظهارها بفاصلة عادية ,، والعمود الأخير لا نكتب بعده فاصلة، بل كلمة From مباشر.

وسيكون ناتج الجملة البيانات التالية:

First_Name	Age
Ibrahim	21
Mohammed	25
Saleem	25
Ali	25
Reem	25

الاستعلام عن السجلات الفريدة:

في بيانات الجدول، ستجد في كثير من الأحيان أن هناك تكراراً للقيم في عمود ما، وقد تحتاج إلى الاستعلام عن القيم دون تكرار، فمثلا، في جدول الأشخاص Persons السابق، ستلاحظ أن عمود العمر Age يحتوي على ٥ قيم، ولكن توجد ٤ سجلات من نفس القيمة وهي ٢٥، وهنا يأتي دور جملة الاستعلام الفريد DISTINCT Select.

تُرجع جملة الاستعلام عن السجلّات الفريدة سجلات دون تكرار في القيم وصيغتها العامة:

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

لو نفذنا الجملة التالية:

```
SELECT Age
FROM Persons;
```

سيكون الناتج:

Age
21
25
25
25
25

ولكن لو استخدمنا جملة الاستعلام عن السجلّات الفريدة

```
SELECT DISTINCT Age
FROM Persons;
```

ستكون النتيجة كالتالي:

Age
21
25

ترشيح السجلات:

لإجراء عملية ترشيح السجلات، سنضيف إلى جملة الاستعلام جملة شرطية تبدأ بالكلمة Where ويتبعها الشرط (أو مجموعة الشروط) الذي نريد والذي سيُرشِّح السجلات بحيث تبقى السجلات التي تحقق الشرط في مجموعة البيانات الراجعة، وتُستبعد السجلات التي لا تحقق الشرط.

الصيغة العامة لجملة الاستعلام والتي تحتوي على شرط لترشيح السجلات:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

أمثلة على ترشيح البيانات في جدولPersons :

الحصول على البيانات الكاملة للشخص الذي له Person_ID يساوي ١٠١:

```
SELECT *
FROM  Persons
WHERE  Person_ID = 101;
```

الاستعلام عن أسماء الأشخاص الذين تساوي أعمارهم ٢٥ سنة أو تزيد عليها:

```
SELECT First_Name, Last_Name
FROM Persons
WHERE Age >= 25;
```

الاستعلام عن الاسم الأول والعمر للأشخاص الذين ليس لديهم قيمة للعمود Last_Name وأعمارهم فوق ٢٢:

```
SELECT  First_Name, Age
FROM  Persons
WHERE  Age > 22 AND Last_Name IS Null;
```

عمليات المقارنة في جملةWhere :

يلخص الجدول التالي العمليات التي من الممكن استخدامها في بناء شرط جملة:Where

العمليّة الوصفمثال:

العمليّة	الوصف	مثال
=	يساوي	Age = 20
<>	لا يساوي (في بعض النظم تكتب(=!	Age <> 20
>	أكبر من	Age > 20
<	أصغر من	Age < 20
>=	أكبر من أو يساوي	Age >= 20
<=	أصغر من أو يساوي	Age <= 20
BETWEEN … AND	بين قيمتين أو يساويهما	Age BETWEEN 20 AND 25
LIKE	مطابقة نمط	First_Name LIKE "%Ibr%"
IN	يوجد ضمن قيم معينة	Age in (20,23,25)

ملاحظة هامة: نستطيع الجمع بين أكثر من شرط في جملة Where وذلك باستخدام العمليات المنطقية) NOTللنفي(،) ANDوجوب تحقّق جميع الشروط) أو) OR يكفي تحقّق شرط واحد من الشروط.(

ترتيب السجلات:

نستطيع الحصول على البيانات الراجعة مرتبة تصاعديا أو تنازليا بعد تنفيذ جملة الاستعلام، وذلك باستخدام جملة.Order By

ترتّب الجملةُ السجلات تصاعديًّا وهو الخيار المبدئي، ولترتيبها تنازليًّا نستخدم الكلمة المحجوزة DESC، كما أنه يمكن الترتيب باستخدام عمود واحد أو أكثر.

الصيغة العامة لجملة الاستعلام مع جملة الترتيب هي:

```
SELECT  column1, column2, ...
FROM  table_name
ORDER  BY  column1, column2, ... ASC|DESC;
```

فمثلا لو أردنا أن نستعلم عن كل البيانات من جدول Persons بحيث تكون البيانات مرتبة ترتيبا تصاعديا حسب عمودFirst_Name ، نستخدم الجملة التالية:

```
SELECT *
FROM  Persons
ORDER  BY First_Name;
```

وستكون النتيجة:

Person_ID	First_Name	Last_Name	Age
104	Ali	Mohammed	25
101	Ibrahim	Mohammed	21
102	Mohammed	Khaled	25
105	Reem		25
103	Saleem	Yaser	25

في حال أردنا أن نرتب نفس البيانات بطريقة تنازلية نستخدم الجملة التالية:

```
SELECT *
FROM  Persons
ORDER  BY First_Name  DESC;
```

وتكون نتيجة الاستعلام كالتالي:

Person_ID	First_Name	Last_Name	Age
103	Saleem	Yaser	25
105	Reem		25
102	Mohammed	Khaled	25
101	Ibrahim	Mohammed	21
104	Ali	Mohammed	25