١- اهمية قواعد البيانات

تعد قواعد البيانات وأنظمة قواعد البيانات مكونًا أساسيًا للحياة في المجتمع الحديث: فمعظمنا يتفاعل مع قاعدة البيانات كل يوم من خلال عدة أنشطة، على سبيل المثال :

- إذا ذهبنا إلى البنك لإيداع الأموال أو سحبها أو
- إذا قمنا بحجز فندق أو شركة طيران أو
- إذا وصلنا إلى كتالوج مكتبة محوسب للبحث عن عنصر ببليوغرافي أو
- إذا اشترينا شيئًا عبر الإنترنت - مثل كتاب ، لعبة ، أو كمبيوتر وقد تكون أنشطتنا هذه عبر شخصًا ما أو بعض برامج الكمبيوتر للوصول إلى قاعدة بيانات. وحتى شراء المواد من السوبر ماركت غالبًا ما يقوم تلقائيًا بتحديث قاعدة البيانات التي تحتوي على مخزون عناصر البقالة.

وهذه الامثلة هي من **تطبيقات قواعد البيانات التقليدية** ، حيث تكون معظم المعلومات التي يتم تخزينها والوصول إليها إما نصية أو رقمية. ولكن حدث تطور في السنوات القليلة الماضية ، حيث أدى التقدم التكنولوجي إلى ظهور **تطبيقات جديدة** ومثيرة لأنظمة قواعد البيانات وحسب ماظهر من احتياج لها في عالم اليوم .

فعلى سبيل المثال ما نراه من انتشار مواقع الويب الخاصة بوسائل التواصل الاجتماعي ، مثل Facebook و Twitter و Flickr التي تحتاج الى إنشاء قواعد بيانات ضخمة تخزن البيانات غير التقليدية ، حيث ان هذه المواقع تتداول المنشورات والتغريدات والصور ومقاطع الفيديو، وهذه عبارة عن بيانات تحتاج الى إنشاء أنواع جديدة من أنظمة قواعد البيانات ، والتي يشار إليها غالبًا باسم أنظمة تخزين البيانات الضخمة ، أو أنظمة NOSQL ، لإدارة البيانات الخاصة بتطبيقات وسائل التواصل الاجتماعي.

ومثال اخر ما يتم استخدامه من الأنظمة من قبل شركات مثل Google و Amazon و Yahoo لإدارة البيانات المطلوبة في محركات بحث الويب الخاصة بهم ، وكذلك لتوفير التخزين السحابي Cloud Storage، حيث يتم

تزويد المستخدمين بقدرات تخزين على الويب لإدارة جميع أنواع البيانات بما في ذلك المستندات والبرامج والصور ومقاطع الفيديو ورسائل البريد الإلكتروني.

وهنالك بعض التطبيقات الأخرى لقواعد البيانات مثل :

- إمكانية نقل الصور ومقاطع الصوت وتدفق الفيديو رقميًا عبر الهواتف المحمولة والأجهزة الأخرى. وتعتبر هذه الأنواع من الملفات مكونًا مهمًا لقواعد بيانات الوسائط المتعددة Multi-media.
- أنظمة المعلومات الجغرافية (GISs) تخزين وتحليل الخرائط وبيانات الطقس وصور الأقمار الصناعية.
- استخدام مستودعات البيانات وأنظمة المعالجة التحليلية عبر الإنترنت (OLAP) لمساعدة العديد من الشركات في استخراج وتحليل معلومات الأعمال المفيدة من قواعد البيانات الكبيرة جدًا لدعم اتخاذ القرار فيها.

٢- مقدمة الى قواعد البيانات

لقد كان لقواعد البيانات وتقنية قواعد البيانات تأثير كبير على الاستخدام المتزايد لأجهزة الكمبيوتر. ومن الإنصاف القول إن قواعد البيانات تلعب دورًا مهمًا في جميع المجالات التي تستخدم فيها أجهزة الكمبيوتر تقريبًا ، بما في ذلك الأعمال والتجارة الإلكترونية ووسائل التواصل الاجتماعي والهندسة والطب وعلم الوراثة والقانون والتعليم وعلوم المكتبات .

البيانات Data : هي مجموعة من القيم المنفصلة التي تنقل المعلومات ، ,هي حقائق معروفة يمكن تسجيلها و يتم تمثيلها أو ترميزها بشكل ما مناسب للاستخدام أو المعالجة بشكل أفضل. والتي لها معنى ضمني. على سبيل المثال ، ضع في اعتبارك الأسماء وأرقام الهواتف وعناوين الأشخاص الذين تعرفهم. و تمثل البيانات الحقائق والأرقام الأولية التي يمكن استخدامها بهذه الطريقة من أجل الحصول على المعلومات المفيدة منها. يتم جمع البيانات تقليديًا باستخدام تقنيات مثل القياس أو الملاحظة أو الاستعلام أو التحليل ،

وعادة ما يتم تمثيلها كأرقام أو رموز يمكن معالجتها بشكل أكبر. • البيانات هي أصغر وحدات المعلومات الواقعية

قاعدة البيانات DataBase: هي مجموعة من البيانات ذات الصلة . وفي الوقت الحاضر ، عادةً ما يتم تخزين هذه البيانات في الهواتف المحمولة ، والتي لديها برنامج قاعدة بيانات بسيط خاص بها. يمكن أيضًا تسجيل هذه البيانات في دفتر عناوين مفهرس أو تخزينها على محرك أقراص ثابت باستخدام جهاز كمبيوتر شخصي وبرامج مثل Microsoft Access أو Excel. هذه المجموعة من البيانات ذات الصلة ذات المعنى الضمني هي قاعدة بيانات.

إن الاستخدام الشائع لمصطلح قاعدة البيانات عادة ما يكون أكثر تقييدًا ، حيث ان قاعدة البيانات تمتاز بمجموعة من الخصائص التالية:

■ تمثل قاعدة البيانات بعض جوانب العالم الحقيقي ، تسمى أحيانًا بالعالم المصغر **miniworld**. وتنعكس التغييرات على العالم الحقيقي في قاعدة البيانات.

■ قاعدة البيانات هي مجموعة متماسكة منطقيًا من البيانات مع بعض المعاني المتأصلة. لا يمكن الإشارة إلى مجموعة عشوائية من البيانات بشكل صحيح على أنها قاعدة بيانات.

■ تم تصميم قاعدة البيانات وإنشائها وملؤها بالبيانات لغرض محدد. لديها مجموعة مقصودة من المستخدمين وبعض التطبيقات المسبقة التي يهتم بها هؤلاء المستخدمون. بمعنى آخر ، تحتوي قاعدة البيانات على بعض المصادر التي تُشتق منها البيانات ، ودرجة معينة من التفاعل مع الأحداث في العالم الحقيقي ، وقد يقوم المستخدمون النهائيون لقاعدة البيانات بإجراء معاملات تجارية (على سبيل المثال ، يشتري العميل كاميرا) أو قد تحدث أحداث (على سبيل المثال ، لدى الموظف طفل) والتي تتسبب في تغيير المعلومات الموجودة في قاعدة البيانات. لكي تكون قاعدة البيانات دقيقة وموثوقة في جميع الأوقات ، يجب أن تكون انعكاسًا حقيقيًا للعالم المصغر الذي تمثله ؛ لذلك ، يجب أن تنعكس التغييرات في قاعدة البيانات في أسرع وقت ممكن.

يمكن أن تكون قاعدة البيانات بأي حجم وتعقيد. على سبيل المثال ، قد تتكون قائمة الأسماء والعناوين المشار إليها سابقًا من بضع مئات من السجلات فقط ، ولكل منها بنية بسيطة. من ناحية أخرى ، قد يحتوي الكتالوج المحوسب لمكتبة كبيرة على نصف مليون إدخال منظم تحت فئات مختلفة - حسب الاسم الأخير للمؤلف الرئيسي ، حسب الموضوع ، حسب عنوان الكتاب - مع تنظيم كل فئة أبجديًا.

سيتم الاحتفاظ بقاعدة بيانات ذات حجم وتعقيد أكبر من قبل شركة وسائط اجتماعية مثل Facebook ، التي لديها أكثر من مليار مستخدم. لذلك تحتفظ قاعدة البيانات بالمعلومات المتعلقة بالمستخدمين المرتبطين ببعضهم البعض كأصدقاء ، ومنشورات كل مستخدم ، والمستخدمين الذين يُسمح لهم برؤية كل منشور ، وكمية هائلة من أنواع المعلومات الأخرى اللازمة للتشغيل الصحيح لموقع الويب الخاص بهم .

بالنسبة لمواقع الويب هذه ، هناك حاجة إلى عدد كبير من قواعد البيانات لتتبع المعلومات المتغيرة باستمرار التي يتطلبها موقع الويب الخاص بوسائل التواصل الاجتماعي.

مثال أخر على قاعدة بيانات تجارية كبيرة هو مايرتبط بـ Amazon.com. التي تحتوي على بيانات لأكثر من ٦٠ مليون مستخدم نشط ، وملايين الكتب ، والأقراص المدمجة ، ومقاطع الفيديو ، وأقراص DVD ، والألعاب ، والإلكترونيات ، والملابس ، وعناصر أخرى. تشغل قاعدة البيانات أكثر من ٤٢ تيرابايت (تبلغ قيمة التخزين $١٠^{١٢}$ بايت) ويتم تخزينها على مئات أجهزة الكمبيوتر (تسمى الخوادم).

يدخل ملايين الزوار إلى Amazon.com يوميًا ويستخدمون قاعدة البيانات لإجراء عمليات شراء. يتم تحديث قاعدة البيانات باستمرار حيث تتم إضافة كتب جديدة وعناصر أخرى إلى المخزون ، ويتم تحديث كميات المخزون عند إجراء عمليات الشراء. قد يتم إنشاء قاعدة بيانات وصيانتها يدويًا أو قد

تكون محوسبة. على سبيل المثال ، فهرس بطاقات المكتبة هو قاعدة بيانات يمكن إنشاؤها وصيانتها يدويًا.

٣- ادارة قواعد البيانات

يمكن إنشاء قاعدة بيانات محوسبة وصيانتها إما عن طريق مجموعة من برامج التطبيق المكتوبة خصيصًا لهذه المهمة أو عن طريق نظام إدارة قواعد البيانات..

نظام إدارة قواعد البيانات (DBMS) : هو نظام محوسب يمكّن المستخدمين من إنشاء قاعدة بيانات والحفاظ عليها. يعد DBMS نظامًا برمجيًا للأغراض العامة ويشتمل على الوظائف التالية:

١- عمليات تحديد قواعد البيانات وإنشائها: يتضمن تحديد قاعدة البيانات تحديد أنواع البيانات والهياكل والقيود الخاصة بالبيانات المراد تخزينها في قاعدة البيانات. يتم أيضًا تخزين تعريف قاعدة البيانات أو المعلومات الوصفية بواسطة DBMS في شكل كتالوج أو قاموس قاعدة البيانات ؛ يطلق عليه البيانات الوصفية **Meta-data** . بناء قاعدة البيانات هو عملية تخزين البيانات على بعض وسائط التخزين التي يتحكم فيها نظام إدارة قواعد البيانات

٢- معالجة قاعدة البيانات واجراء العمليات المختلفة عليها: تتضمن معالجة قاعدة البيانات وظائف مثل الاستعلام عن قاعدة البيانات لاسترداد بيانات محددة ، وتحديث قاعدة البيانات لتعكس التغييرات في العالم المصغر ، وإنشاء تقارير من البيانات.

٣- مشاركتها بين مختلف المستخدمينUsers والتطبيقات Applications: . تسمح مشاركة قاعدة البيانات للعديد من المستخدمين والبرامج بالوصول إلى قاعدة البيانات في وقت واحد.

برنامج التطبيق Application Program وهو البرنامج الذي يصل إلى قاعدة البيانات عن طريق إرسال استعلامات أو طلبات للبيانات إلى نظام إدارة قواعد البيانات. أما **الاستعلام Query وهو** عادةً يشير إلى الطلب من قاعدة البيانات للحصول على بعض البيانات

؛ وقد تؤدي هذه المعاملة إلى قراءة بعض البيانات وكتابة بعض البيانات في قاعدة البيانات.

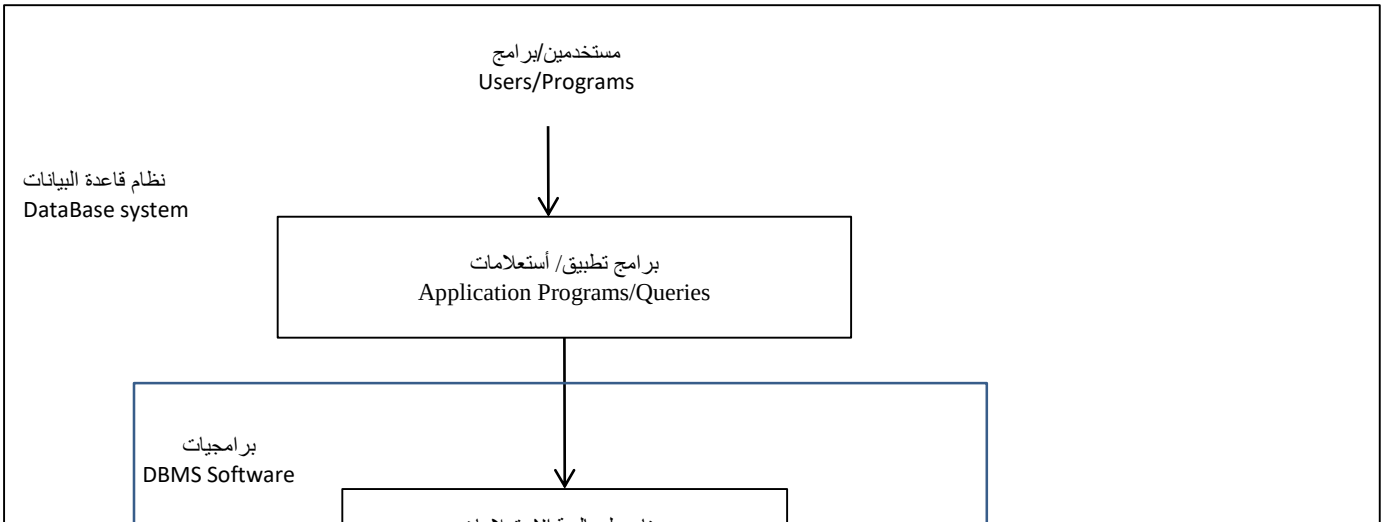
٤-حماية قاعدة البيانات والحفاظ عليها لفترة طويلة من الزمن. تشمل الحماية حماية النظام ضد عطل الأجهزة أو البرامج (أو الأعطال) وحماية الأمان من الوصول غير المصرح به أو الضار. قد يكون لقاعدة البيانات الكبيرة النموذجية دورة حياة لعدة سنوات ، لذلك يجب أن يكون نظام إدارة قواعد البيانات قادرًا على الحفاظ على نظام قاعدة البيانات من خلال السماح للنظام بالتطور مع تغير المتطلبات بمرور الوقت.

أن معظم نظم إدارة قواعد البيانات.DBMS هي أنظمة برمجية معقدة للغاية ويمكن بناءها على نوعين تبعا لطريقة الاستخدام :

١-مخصص للأغراض العامة لتنفيذ قاعدة بيانات حسب الاحتياجات مثل أوراكل Oracle ومايكروسوفت أكسس ACCESS .

٢-إنشاء برنامج DBMS لأغراض خاصة لتطبيق معين : من الممكن كتابة مجموعة مخصصة من البرامج لإنشاء قاعدة البيانات وصيانتها ، مثل حجوزات شركات الطيران.

نظام قاعدة البيانات: وهو عبارة عن اجتماع قاعدة البيانات مع برنامج DBMS . يوضح الشكل ١,١ بعض المفاهيم التي ناقشناها حتى الآن.



مثال

الكثير منا على دراية بمثال بسيط وهو : قاعدة بيانات الجامعة للحفاظ على المعلومات المتعلقة بالطلاب والدورات والدرجات في بيئة الجامعة. يوضح الشكل ١,٢ بنية قاعدة البيانات وعدد قليل من سجلات البيانات النموذجية. تم تنظيم قاعدة البيانات في خمسة ملفات ، كل منها تحتوي على سجلات البيانات من نفس النوع .

الطالب

اسم الطالب	رقم الطالب	المرحلة	القسم
Smith	17	1	CS
Brown	8	2	CS

المتطلبات الاساسية

رقم الكورس	رقم المتطلب
CS33380	CS3320
CS3380	MATH2410
CS3320	CS1310

الكورس

اسم الكورس	رقم الكورس	عدد الوحدات	القسم
Intro to Computer Science	CS1310	4	CS
Data Structure	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

تقرير الدرجات

رقم الطالب	معرف المقطع	الدرجة
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

المقطع

معرف المقطع	رقم الكورس	الفصل	السنة	مدرس الكورس
85	MATH2410	Fall	2020	King
92	CS1310	Fall	2020	Anderson
102	CS3320	Spring	202	Kinith
112	MATH2410	Fall	2021	Chang
119	CS1310	Fall	2021	Anderson
135	CS3380	Fall	2021	Stone

الشكل(٢) : مثال قاعدة بيانات لخزن بيانات عن الطلبة ومواهم الدراسية

يخزن ملف الطالب البيانات عن كل طالب ، يخزن ملف الدورة التدريبية البيانات في كل دورة ، ويخزن ملف القسم البيانات في كل قسم من الدورة ، يخزن ملف "تقرير الدرجات" الدرجات التي يتلقاها الطلاب في الأقسام المختلفة التي أكملوها ، ويخزن الملف "المتطلبات" المتطلبات الأساسية لكل دورة تدريبية.

لتعريف قاعدة البيانات هذه ، يجب علينا تحديد هيكل سجلات كل ملف عن طريق تحديد الأنواع المختلفة من عناصر البياناتitem التي سيتم تخزينها في كل سجل **record**. في الشكل ١,٢ ، يتضمن كل سجل الطالب بيانات تمثل اسم الطالب ورقم الطالب والفصل الدراسي (مثل طالب جديد أو '١' أو طالبة أو '٢' وما إلى ذلك) ، والتخصص (مثل الرياضيات أو 'MATH' والكمبيوتر العلوم أو "CS") ؛ يتضمن كل سجل من سجلات الدورة التدريبية بيانات لتمثيل اسم الدورة التدريبية ورقم الدورة وساعات الائتمان والقسم (القسم الذي يقدم الدورة التدريبية) وما إلى ذلك. يجب علينا أيضًا تحديد نوع البيانات لكل عنصر بيانات داخل السجل.
على سبيل المثال ، يمكننا تحديد أن "اسم الطالب" عبارة عن سلسلة من الأحرف الأبجدية ، و "رقم الطالب" هو عدد صحيح ، و "الدرجة" في ملف "تقرير الدرجات" هو حرف واحد من المجموعة {'A' و 'B' و 'C' و " D، ' ' I،F ' '}. قد نستخدم أيضًا مخطط تشفير لتمثيل قيم عنصر البيانات. على سبيل المثال ، في الشكل ١,٢ ، نعرب عن استياءنا من فئة الطالب على أنه ١ للطالب الجديد ، و ٢ للطالب الثاني ، و ٣ للمبتدئين ، و ٤ للكبار ، و ٥ للطالب الخريجين.

لإنشاء قاعدة بيانات UNIVERSITY جامعة ، نقوم بتخزين البيانات لتمثيل كل طالب ودورة تدريبية وقسم وتقرير تقدير ومتطلب أساسي كسجل

في الملف المناسب. لاحظ أن السجلات الموجودة في الملفات المختلفة قد تكون ذات صلة. على سبيل المثال ، يرتبط سجل Smith في ملف "الطالب" بسجلين في ملف "تقرير الدرجات" يحددان درجات Smith في قسمين.
وبالمثل ، يرتبط كل سجل في الملف "المتطلبات" بسجلين:
أحدهما يمثل الدورة والآخر يمثل الشرط الأساسي. تتضمن معظم قواعد البيانات المتوسطة الحجم والكبيرة أنواعًا عديدة من السجلات ولها العديد من العلاقات بين السجلات.
ويمكن اجراء معالجة لقاعدة البيانات من خلال الاستعلام والتحديث و فيما يلي أمثلة على الاستعلامات:

- استرداد نسخة - قائمة بجميع الدورات والدرجات - من "سميث"
- قائمة بأسماء الطلاب الذين أخذوا قسم "قاعدة البيانات" الدورة التدريبية المقدمة في خريف ٢٠٠٨ ودرجاتهم في هذا القسم
- سرد المتطلبات الأساسية لـ "

تتضمن أمثلة دورات قواعد البيانات على التحديثات ما يلي:

- تغيير فئة "سميث" إلى طالب السنة الثانية
- إنشاء قسم جديد لدورة "قاعدة البيانات" لهذا الفصل الدراسي
- أدخل درجة "أ" لـ "سميث" في قسم "قاعدة البيانات" في الفصل الدراسي الأخير. هذه الاستفسارات والتحديثات غير الرسمية يجب تحديدها بدقة في لغة الاستعلام لنظام إدارة قواعد البيانات قبل أن تتم معالجتها.

٤-انشاء قاعدة البيانات

في هذه المرحلة ، من المفيد وصف قاعدة البيانات كجزء من مشروع أكبر يُعرف بنظام المعلومات داخل المنظمة. يقوم قسم تكنولوجيا المعلومات (IT) داخل المنظمة بتصميم وصيانة نظام معلومات يتكون من أجهزة كمبيوتر وأنظمة تخزين وبرامج تطبيقية وقواعد بيانات مختلفة. يبدأ تصميم تطبيق جديد لقاعدة بيانات موجودة أو تصميم قاعدة بيانات جديدة تمامًا بمرحلة تسمى تحديد المتطلبات والتحليل.

يتم توثيق هذه المتطلبات بالتفصيل وتحويلها إلى تصميم مفاهيمي يمكن تمثيله ومعالجته باستخدام بعض الأدوات المحوسبة بحيث يمكن صيانتها وتعديلها وتحويلها بسهولة إلى تطبيق قاعدة بيانات.
ثم يتم ترجمة التصميم إلى تصميم منطقي يمكن التعبير عنه في نموذج بيانات مطبق في نظام DBMS تجاري.
المرحلة النهائية هي التصميم المادي ، حيث يتم توفير المزيد من المواصفات لتخزين قاعدة البيانات والوصول إليها. يتم تنفيذ تصميم قاعدة البيانات وتزويدها بالبيانات الفعلية والمحافظة عليها باستمرار لتعكس حالة العالم المصغر.

نظام إدارة قواعد البيانات (DBMS) وتطبيقاته:

نظام إدارة قواعد البيانات هو نظام محوسب لحفظ السجلات. إنه مستودع أو حاوية لـ جمع ملفات البيانات المحوسبة. الغرض العام من DBMS هو السماح للمستخدمين بتحديد وتخزين واسترداد

وتحديث المعلومات الواردة في قاعدة البيانات عند الطلب. يمكن أن تكون المعلومات أي شيء له أهمية للفرد أو المنظمة. تمس قواعد البيانات جميع جوانب حياتنا. وبعض مجالات التطبيق الرئيسية لها هي كما يلي:

١. المصرفية
٢. شركات الطيران
٣. الجامعات
٤. التصنيع والبيع
٥. الموارد البشرية

أمثلة

معلومات المؤسسة

◦ المبيعات: للعميل والمنتج ومعلومات الشراء.

المحاسبة: للمدفوعات والإيصالات وأرصدة الحسابات والأصول والمعلومات المحاسبية الأخرى.

◦ الموارد البشرية: للحصول على معلومات حول الموظفين والرواتب وضرائب الرواتب والمزايا والتوليد

البنوك والتمويل

◦ المصرفية: لمعلومات العملاء والحسابات والقروض والمعاملات المصرفية.

◦ معاملات بطاقات الائتمان: للمشتريات على بطاقات الائتمان وإصدار كشوفات حساب شهرية.

◦ التمويل: لتخزين المعلومات حول مقتنيات ومبيعات وشراء الأدوات المالية مثل

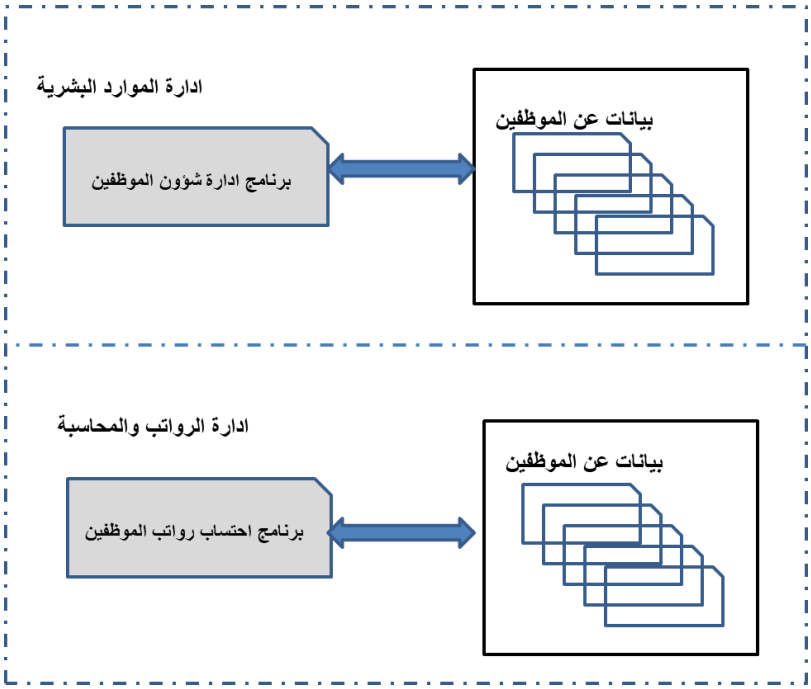
الاسهم والسندات؛ أيضًا لتخزين بيانات السوق في الوقت الفعلي لتمكين التداول عبر الإنترنت من قبل العملاء و
التداول الآلي من قبل الشركة.
• الجامعات: للحصول على معلومات الطلاب وتسجيلات الدورات والدرجات (بالإضافة إلى المؤسسة القياسية
معلومات مثل الموارد البشرية والمحاسبة).
• الخطوط الجوية: للحجز ومعلومات الجدول الزمني. كانت شركات الطيران من بين أول من استخدم قواعد البيانات في أ
بطريقة موزعة جغرافيا.
• الاتصالات السلكية واللاسلكية: للاحتفاظ بسجلات المكالمات التي تم إجراؤها ، وإنشاء فواتير شهرية ، والحفاظ على أرصدة
بطاقات الاتصال مسبقة الدفع ، وتخزين المعلومات الخاصة بشبكات الاتصال.

ماهي الحاجة لأنظمة قواعد البيانات

نشأت أنظمة قواعد البيانات استجابة للأساليب التي ظهرت في إدارة محوسبة للبيانات التجارية. و مثال على هذه الأساليب ، وكانت الطريقة الوحيدة المتوفرة للحفاظ على المعلومات الموجودة على جهاز الكمبيوتر لتخزينها في ملفات نظام التشغيل.
وهذا الاسلوب المستخدم يطلق عليه بـ"نظام معالجة الملف File processing system.

نظام معالجة الملف

وهو النظام المستخدم قبل ظهور نظم ادارة قواعد البيانات DBMS. حيث يتم دعم نظام معالجة الملفات هذا بواسطة نظام تشغيل تقليدي. ويخزن النظام السجلات الدائمة للبيانات في ملفات مختلفة ، ويحتاج إلى برامج تطبيقية مختلفة لاستخراج السجلات منها وإضافتها إلى الملفات المناسبة كما في المثال الموضح في الشكل التالي:



في هذا المثال ، تحتفظ ادارة الموارد البشرية ، بملفات عن الموظفين لغرض متابعة شؤنهم الوظيفية واستخدامها في استخراج تقارير عن اجازات وغيابات الموظفين كجزء من برنامج يعمل في ادارة شؤون الموظفين . بينما نرى مستخدم ثاني مثل ادارة الرواتب والمحاسبة تتعامل ببيانات مشابهة لهذه البيانات عن الموظفين ولكنها ستقوم بتحديد البيانات التي تحتاجها في برنامج حساب الرواتب وتقوم بتنفيذ ملفات بيانات تختلف عن بيانات المستخدم الاول. على الرغم من اهتمام كلا المستخدمين بالبيانات المتعلقة بالموظفين ، فإن كل مستخدم يحتفظ بملفات منفصلة - وبرامج لمعالجة هذه الملفات - لأن كل منهما يتطلب بعض البيانات غير المتوفرة من ملفات المستخدم الآخر. ينتج عن هذا التكرار في تعريف البيانات وتخزينها مساحة تخزين ضائعة وجهود زائدة عن الحاجة للحفاظ على بيانات محدثة مشتركة. في نهج قاعدة البيانات ، يحتفظ المستودع الفردي بالبيانات التي تم تحديدها مرة واحدة ثم الوصول إليها من قبل العديد من المستخدمين بشكل متكرر من خلال الاستعلامات والمعاملات وبرامج التطبيقات.

إن الاحتفاظ بالمعلومات التنظيمية في نظام معالجة الملفات له عدد من العيوب الرئيسية:

١- تكرار البيانات وعدم تناسقها Data Redundancy and inconsistency. نظرًا لأن المبرمجين المختلفين يقومون بإنشاء الملفات

وبرامج التطبيق على مدى فترة طويلة ، من المحتمل أن يكون للملفات المختلفة هياكل مختلفة ويمكن كتابة البرامج بها بعدة لغات برمجة. علاوة على ذلك ، يمكن تكرار نفس المعلومات في عدة أماكن (ملفات).

على سبيل المثال ، فان معلومات الموظف (العنوان ورقم الهاتف) يمكن ان تتكرر في اكثر من ملف (ملف الموارد البشرية وملف الرواتب). هذا التكرار يؤدي إلى ارتفاع تكلفة التخزين والوصول. بالإضافة إلى ذلك ، قد يؤدي إلى عدم اتساق البيانات وهذا هو Data inconsistency ؛ ، ويعبر عنه بوجود النسخ المختلفة لنفس البيانات ولكن غير متناسقة . على سبيل المثال ،لو ان "عنوان الموظف" تم تغييره في سجلات الموارد البشرية ولكن لم يتم تغييره في ملف الرواتب فيحدث عدم اتساق بالعنوان .

٢- **صعوبة في الوصول إلى البيانات ومعالجتها** : ولغرض السماح للمستخدمين بالوصول المعلومات ، فهنالك عدد من البرامج التطبيقية التي تتعامل مع الملفات ، بما في ذلك البرامج من أجل:

- إضافة موظفين جدد او تحديث معلوماتهم
- وإنشاء قوائم بالرواتب
- تعيين غيابات الموظفين واحتساب الغيابات.

ولأجل هذا الغرض يجب على مبرمجي النظام كتابة برامج التطبيق لتلبية هذه الاحتياجات . وكلما دعت الحاجة فانه يجب إضافة برامج تطبيق جديدة إلى النظام.

٣- عزل البيانات Data isolation : لأن البيانات مبعثرة في ملفات مختلفة ، وقد تكون الملفات مخزونة بتنسيقات مختلفة ،ولذلك فان برامج تطبيقية لاسترجاع البيانات المناسبة سيكون مهمة صعبة.

٤- مشاكل التكامل Integrity problem . يجب أن تخضع قيم البيانات المخزنة في قاعدة البيانات بأنواع معينة من شروط تناسق القيم. فعلى سبيل المثال فلنفترض أن الجامعة تحتفظ بحساب لكل قسم وتسجيل رصيد مبلغ في كل حساب. و رصيد حساب القسم يجب ان لا تكون قيمته تحت الصفر أبدًا. ولذلك يقوم المبرمجون بوضع هذه القيود في النظام عن طريق إضافة

التعليمات البرمجية المناسبة في برامج التطبيق المختلفة. ومع ذلك ، عند إضافة قيود جديدة ، من الصعب تغيير البرامج.

٥- مشاكل الأمنيةSecurity Problem . لا ينبغي أن يكون كل مستخدم لنظام قاعدة البيانات قادرًا على الوصول إلى جميع البيانات. فمثلا، في إحدى الجامعات ، يحتاج موظفو كشوف المرتبات إلى رؤية ذلك الجزء فقط من قاعدة البيانات الذي يحتوي على معلومات مالية. وهم لا يحتاجون إلى الوصول إلى معلومات حول السجلات الأكاديمية ودخولهم لها في هذه الحالة هو خرق امني للبيانات. و بما أن برامج التطبيق تضاف إلى ملف نظام معالجة الملفات بطريقة مخصصة ، وفرض مثل هذه القيود الأمنية أمر صعب.

هذه الصعوبات ، من بين أمور أخرى ، دفعت إلى تطوير أنظمة قواعد البيانات. فيما يلي سنرى المفاهيم والخوارزميات التي تمكن أنظمة قواعد البيانات من حل مشاكل أنظمة معالجة الملفات.

<u>مزايا نظم إدارة قواعد البيانات:</u>

١- التحكم في التكرار Controlling of Redundancy : يشير تكرار البيانات إلى تكرار البيانات (أي تخزين البيانات نفسها عدة مرات). في نظام قاعدة البيانات ، من خلال وجود قاعدة بيانات مركزية والتحكم المركزي في البيانات من قبل DBA ، فإن يتم تجنب الازدواجية غير الضرورية في البيانات. كما أنه يلغي الوقت الإضافي لمعالجة الحجم الكبير من ملفات البيانات. وينتج عنه توفير مساحة التخزين.

٢- تحسين مشاركة البيانات Improved Data Sharing: يسمح نظام إدارة قواعد البيانات (DBMS) للمستخدم بمشاركة البيانات في أي عدد من برامج التطبيق.

٣- تكامل البيانات Integrity Data: تعني التكامل أن البيانات الموجودة في قاعدة البيانات دقيقة. ويساعد التحكم المركزي في البيانات في السماح للمسؤول بتعريف قيود تكامل البيانات في قاعدة البيانات.

٤- الأمان Security: امتلاك سلطة كاملة على البيانات التشغيلية ، يمكّن مسؤولي قاعدة البيانات من ضمان أن يكون الوصول إلى قاعدة البيانات من خلال القنوات المناسبة. يمكن ايضا تحديد عمليات التحقق من التفويض ويتم تنفيذها عند محاولة الوصول إلى البيانات الحساسة.

٥- اتساق البيانات Data Consistency: من خلال التخلص من تكرار البيانات ، فإننا نحد بشكل كبير من فرص عدم الاتساق. على سبيل مثال: يتم تخزين عنوان العميل مرة واحدة فقط ، ولا يمكن أن يكون لدينا خلاف على القيم المخزنة. ايضا يتم تبسيط تحديث قيم البيانات بشكل كبير عندما يتم تخزين كل قيمة في مكان واحد فقط. أخيرًا ، نتجنب التخزين المهدر الناتج عن تخزين البيانات الزائدة عن الحاجة.

٦- الوصول الفعال إلى البيانات: في نظام قاعدة البيانات ، تتم إدارة البيانات بواسطة DBMS ويتم الوصول إلى البيانات بالكامل من خلال نظام إدارة قواعد البيانات (DBMS) الذي يوفر مفتاحًا للمعالجة الفعالة للبيانات

٧- إنفاذ المعايير: من خلال مركزية البيانات ، يمكن لـمسؤول قاعدة البيانات إنشاء معايير البيانات وتنفيذها والتي قد تشمل اصطلاحات التسمية ومعايير جودة البيانات وما إلى ذلك.

٨- استقلالية البيانات: يوفر نظام إدارة قاعدة البيانات الواجهة بين برامج التطبيق والبيانات. عندما يتم إجراء تغييرات على تمثيل البيانات ، والبيانات الوصفية يتم الحصول عليها بواسطة DBMS ولكن يستمر DBMS في توفير البيانات لبرنامج التطبيق في الطريقة المستخدمة سابقا. تتولى DBMs مهمة تحويل البيانات عند الضرورة.

٩- تقليل وقت تطوير التطبيقات وصيانتها: يدعم نظام إدارة قواعد البيانات (DBMS) العديد من الوظائف المهمة التي شائعة في العديد من التطبيقات ، والوصول إلى البيانات المخزنة في DBMS ، مما يسهل التطوير السريع التطبيق.

عيوب نظم إدارة قواعد البيانات

١- إنه معقد بعض الشيء. نظرًا لأنه يدعم وظائف متعددة لمنح المستخدم أفضل البرامج الأساسية أصبح معقدًا. يجب أن يكون لدى المصممين والمطورين معرفة دقيقة بالبرنامج لتحقيق أقصى استفادة منه.

٢- نظرًا لتعقيدها ووظائفها ، فإنها تستخدم قدرًا كبيرًا من الذاكرة. كما يحتاج إلى ذاكرة كبيرة تشغيل بكفاءة.

٣- يعمل نظام DBMS على النظام المركزي ، أي ؛ كل المستخدمين من جميع أنحاء العالم يمكنهم الوصول إلى هذا قاعدة البيانات. ومن ثم فإن أي فشل في نظام إدارة قواعد البيانات سيؤثر على جميع المستخدمين.

٤- DBMS هو برنامج معمم ، أي ؛ إنه عمل مكتوب على الأنظمة بأكملها بدلاً من نظام محدد. ومن ثم بعض التطبيق سوف يعمل ببطء

الفاعلون في بيئة نظام قاعدة البيانات

ويقصد بهم الاشخاص الذيم لهم دور في نظام قاعدة البيانات، وهم يختلفون في وظائفهم ومهاراتهم تبعا للدور الذي يقومون به مع ملاحظة ان حجم نظام قاعدة البيانات هو الذي يحدد عدد الأشخاص وادوارهم فعلى سبيل المثال : في نظام قاعدة بيانات صغير يقوم شخص واحد بجميع الادوار التي يحتاجها النظام ، ولا توجد مشاركة مع اشخاص اخرين . ومع ذلك ، في المؤسسات الكبيرة ، يشارك العديد من الأشخاص في تصميم واستخدام وصيانة قاعدة بيانات كبيرة تضم مئات أو آلاف المستخدمين. وبشكل عام يمكن تصنيف هؤلاء الأشخاص الى :

١. مديرو قواعد البيانات DataBase Administrator: ويكون اختصار الأسم الى (DBA) حيث انه في أي مؤسسة يستخدم فيها العديد من الأشخاص نفس الموارد ، هناك حاجة إلى مسؤول رئيسي للإشراف على هذه الموارد وإدارتها. وتقع مسؤولية إدارة هذه الموارد على عاتق مسؤول قاعدة البيانات .(DBA) وهو مسؤول عن السماح بالوصول إلى قاعدة البيانات ، وتنسيق ومراقبة استخدامها ، كذلك هو مسؤول عن مشاكل مثل الخروقات الأمنية وضعف وقت استجابة النظام. و في المؤسسات الكبيرة ، يتم مساعدة مدير قاعدة البيانات من قبل فريق عمل يقوم بهذه الوظائف.

٢. مصممي قواعد البياناتDataBase Designer

مصممو قواعد البيانات مسؤولون عن تحديد البيانات التي سيتم تخزينها في قاعدة البيانات واختيار الهياكل المناسبة لتمثيل هذه البيانات وتخزينها. يتم تنفيذ هذه المهام في الغالب قبل تنفيذ قاعدة البيانات فعليًا وملؤها بالبيانات. تقع على عاتق مصممي قواعد البيانات مسؤولية التواصل مع جميع مستخدمي قاعدة البيانات المحتملين من أجل فهم متطلباتهم وإنشاء تصميم يلبي هذه المتطلبات.

ويتفاعل مصممو قواعد البيانات عادةً مع كل مجموعة محتملة من المستخدمين ويطورون طرق عرض لقاعدة البيانات التي تلبي متطلبات البيانات والمعالجة لهذه المجموعات. ثم يتم تحليل كل طريقة عرض ودمجها مع آراء مجموعات المستخدمين الأخرى. يجب أن يكون التصميم النهائي لقاعدة البيانات قادرًا على دعم متطلبات جميع مجموعات المستخدمين.

٣ . المستخدمون النهائيون End Users : وهم الأشخاص الذين تتطلب وظائفهم الوصول إلى قاعدة البيانات للاستعلام عن التقارير وتحديثها وإنشاءها ؛ قاعدة البيانات موجودة في المقام الأول لاستخدامها. هناك عدة فئات من المستخدمين النهائيين:

■ المستخدمون النهائيون العاديون الذين يستخدمون واجهة استعلام قاعدة بيانات معقدة لتحديد طلباتهم وعادة ما يكونون مديرين من المستوى المتوسط أو العالي أو متصفحات عرضية أخرى.

■ يُشكل المستخدمون النهائيون قليلي الخبرة أو المحددون وهم يشكلون جزءًا كبيرًا من المستخدمين النهائيين لقاعدة البيانات. تدور وظيفتهم الرئيسية حول الاستعلام باستمرار عن قاعدة البيانات وتحديثها ، باستخدام أنواع قياسية من الاستعلامات والتحديثات - تسمى المعاملات المعلبة - التي تمت برمجتها واختبارها بعناية.
العديد من هذه المهام متاحة الآن كتطبيقات جوال للاستخدام مع الأجهزة المحمولة.
تتنوع المهام التي يؤديها هؤلاء المستخدمون. بعض الأمثلة هي: يقوم عملاء البنك والصرافون بفحص أرصدة الحسابات وترحيل السحوبات والودائع. يتحقق وكلاء الحجز أو العملاء لشركات الطيران والفنادق وشركات تأجير السيارات من توافر طلب معين وإجراء الحجوزات.

■ المستخدمون النهائيون اصحاب المهارات من المهندسين والعلماء ومحللي الأعمال وغيرهم ممن يتعرفون تمامًا على مرافق نظام إدارة قواعد البيانات من أجل تنفيذ تطبيقاتهم الخاصة لتلبية متطلباتهم المعقدة.

٤ . محللو النظام System Analysts ومبرمجو التطبيقات (مهندسو البرمجيات) : يحدد محللو النظام متطلبات المستخدمين النهائيين ، وخاصة المستخدمين النهائيين الساذجين والمعاملات ، ويطورون مواصفات للمعاملات القياسية التي تفي بهذه المتطلبات. يقوم مبرمجو التطبيقات بتنفيذ هذه المواصفات كبرامج ؛ ثم يقومون باختبار هذه المعاملات وتصحيحها وتوثيقها وصيانتها. يجب أن يكون هؤلاء المحللين والمبرمجين - يشار إليهم عادةً بمطوري البرامج أو مهندسي البرمجيات - على دراية بمجموعة كاملة من القدرات التي يوفرها نظام إدارة قواعد البيانات (DBMS) لإنجاز مهامهم.

٥. **اشخاص وراء الكواليس**

وهم الأشخاص الذين يرتبط عملهم بتصميم وتطوير وتشغيل برنامج DBMS وبيئة النظام. لا يهتم هؤلاء الأشخاص عادةً بمحتوى قاعدة البيانات نفسها. نسميهم العمال وراء الكواليس ، وهم يشملون الفئات التالية:

■ مصممي ومنفذي نظام DBMS: تصميم وتنفيذ وحدات وواجهات DBMS كحزمة برامج. نظام إدارة قواعد البيانات (DBMS) هو نظام برمجي معقد للغاية يتكون من العديد من المكونات ، أو الوحدات النمطية ، بما في ذلك الوحدات النمطية لتنفيذ الكتالوج ، ومعالجة لغة الاستعلام ، ومعالجة الواجهة ، والوصول إلى البيانات وتخزينها مؤقتًا ، والتحكم في زمن التوافق ، ومعالجة استعادة البيانات وأمانها. يجب أن يتفاعل نظام DBMS مع برامج النظام الأخرى ، مثل نظام التشغيل والمجمعين للغات البرمجة المختلفة.

■ مطورو الأدوات ويقومون بتصميم الأدوات وتنفيذها: ـــحزم البرامج التي تسهل نمذجة قاعدة البيانات وتصميمها ، وتصميم نظام قاعدة البيانات ، وتحسين الأداء. الأدوات عبارة عن حزم اختيارية يتم شراؤها غالبًا بشكل منفصل. وهي تشمل حزمًا لتصميم قاعدة البيانات ومراقبة الأداء واللغة الطبيعية أو واجهات الرسوم والنماذج الأولية والمحاكاة وتوليد بيانات الاختبار. في كثير من الحالات ، يقوم بائعو البرامج المستقلون بتطوير وتسويق هذه الأدوات.

■ المشغلون وموظفو الصيانة: (موظفو إدارة النظام) مسؤولون عن التشغيل الفعلي وصيانة بيئة الأجهزة والبرامج لنظام قاعدة البيانات.

على الرغم من أن هذه الفئات من العاملين وراء الكواليس مفيدة في جعل نظام قاعدة البيانات متاحًا للمستخدمين النهائيين ، إلا أنهم عادةً لا يستخدمون محتويات قاعدة البيانات لأغراضهم الخاصة.

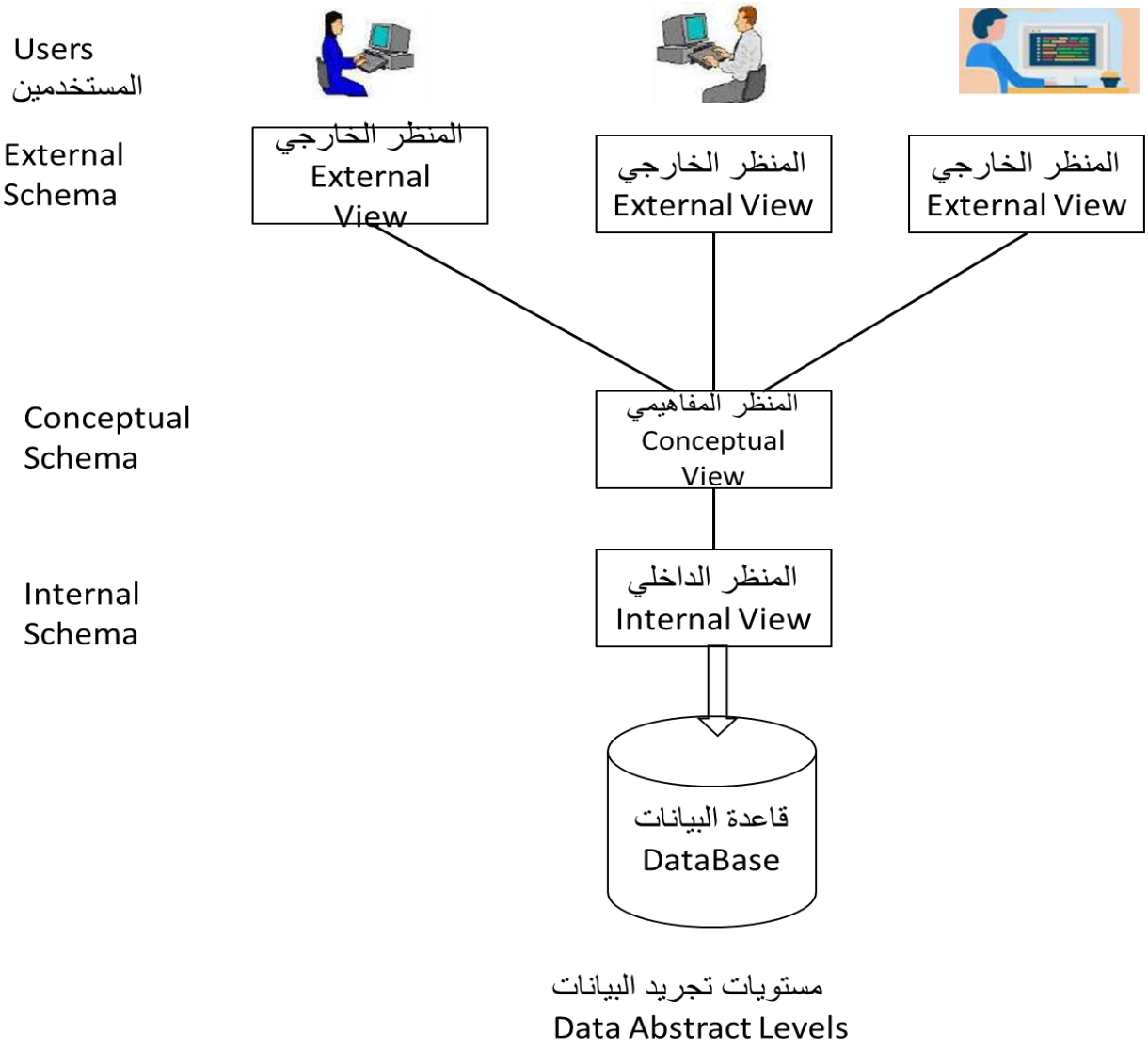
منظر البيانات**View of Data**

تجريد البيانات Data Abstract

تجريد البيانات: وهي تعني أن النظام يخفي تفاصيل معينة عن كيفية تخزين البيانات وصيانتها . ان الغرض الرئيسي من نظام قاعدة البيانات هو تزويد المستخدمين بنظرة مجردة للبيانات لكي يكون النظام قابلاً للاستخدام و يجب أن يسترجع البيانات بكفاءة. وهذه ادت إلى استخدام المصممين هياكل البيانات المعقدة لتمثيل البيانات في قاعدة البيانات. نظرًا لأن العديد من مستخدمي نظام قاعدة البيانات ليسوا مدربين على الكمبيوتر ، يقوم المطورون بإخفاء التعقيد عن المستخدمين من خلال عدة مستويات من التجريد ، لتبسيط تفاعلات المستخدمين مع النظام:

١- المستوى المادي **Physical level (or Internal View / Schema)**:(أو العرض الداخلي / المخطط): يصف أدنى مستوى من التجريد كيفية تخزين البيانات بالفعل. يصف المستوى المادي هياكل البيانات المعقدة منخفضة المستوى بالتفصيل.

٢- المستوى المنطقي (أو العرض المفاهيمي / المخطط) **Logical level (or Conceptual View / Schema)**: يصف المستوى الأعلى التالي من التجريد ما هي البيانات المخزنة في قاعدة البيانات ، وما هي العلاقات الموجودة بين تلك البيانات. وهكذا يصف المستوى المنطقي قاعدة بيانات كاملة من حيث عدد صغير من الهياكل البسيطة نسبيًا. و قد تتضمن الهياكل البسيطة على المستوى المنطقي هياكل معقدة على المستوى المادي ، مستخدم المستوى المنطقي لا يحتاج إلى إدراك هذا التعقيد. يشار إلى هذا باسم استقلال البيانات المادية. قاعدة البيانات. المسؤولون ، الذين يجب أن يقرروا ما هي المعلومات التي يجب الاحتفاظ بها في قاعدة البيانات ، يستخدمون المستوى المنطقي للتجريد.



٣- مستوى العرض (أو العرض / المخطط الخارجي) **View level (or External View / Schema)**: يصف أعلى مستوى من التجريد جزءًا فقط من قاعدة البيانات بأكملها. على الرغم من أن المستوى المنطقي يستخدم هياكل أبسط ، إلا أن التعقيد يظل بسبب تنوع المعلومات المخزنة في قاعدة بيانات كبيرة. لا يحتاج العديد من مستخدمي نظام قاعدة البيانات إلى كل هذه المعلومات ؛ بدلاً من ذلك ، يحتاجون إلى الوصول إلى جزء فقط من قاعدة

البيانات. يوجد مستوى عرض التجريد لتبسيط تفاعلهم مع النظام. قد يوفر النظام العديد من المشاهدات لنفس قاعدة البيانات. يوضح الشكل السابق العلاقة بين مستويات التجريد الثلاثة.

المثيلات Instances والمخططات Schema

المثيل Instance: هو مجموعة البيانات المخزونة في قاعدة البيانات في لحظة معينة، حيث ان قاعدة البيانات يمكن ان تتغير بمرور الوقت فيمكن ان يتم تحديث على المعلومات مثل حذف أو اضافة.

المخطط Schema : هو التصميم العام لقاعدة البيانات ، ويمكن ان تتغير المخططات بشكل غير منتظم وقليل.

وتوجد في انظمة قواعد البيانات العديد من المخططات مقسمة حسب مستويات التجريد:

١- المخطط المادي Physical Schema : وهذا النوع يقوم بوصف تصميم قاعدة البيانات على المستوى المادي.

٢- المخطط المنطقي Logical Schema: يوصف هذا المخطط قاعدة البيانات على المستوى المنطقي

٣- المخططات على مستوى العرض View Schema: وتسمى هذه المخططات احيانا بالمخططات الفرعية ، وقد تحتوي قاعدة البيانات على العديد منها، والتي تصف طرق عرض مختلفة لقاعدة البيانات .

يعتبر المخطط المنطقي هو الأهم من بين هذه المخططات من حيث تأثيره على التطبيقات لأن المبرمجين ينشئون تطبيقاتهم اعتمادا على المخطط المنطقي وأي تغيير بهذا المخطط يؤثر على التطبيقات، اما المخطط المادي فتغييره يكون سهل ولايؤثر على برامج التطبيق لانه معزول عنها .

لغات قاعدة البيانات

يوفر نظام قاعدة البيانات لغات متعدد لتسهيل عملية ادارة فعاليات قواعد البيانات منها:

١- لغة تعريف البيانات لتحديد مخطط قاعدة البيانات **data-definition language** (**DDL**) ، ومن خلالها يتم تحديد مخطط قاعدة البيانات بواسطة مجموعة من التعريفات المعبر عنها. ويتم استخدام DDL أيضًا لتحديد خصائص إضافية للبيانات. نحدد بنية التخزين وطرق الوصول التي يستخدمها نظام قاعدة البيانات من خلال مجموعة من العبارات في نوع خاص من DDL يسمى تخزين البياناتData Storage ولغة التعريفDefinition . تحدد هذه العبارات تفاصيل تنفيذ مخططات قاعدة البيانات ، والتي عادة ما تكون مخفية عن المستخدمين. يجب أن تستوفي قيم البيانات المخزنة في قاعدة البيانات قيود اتساق معينة Consistency. على سبيل المثال ، افترض أن الجامعة تطلب ألا يكون عمر الطالب أكبر اويساوي ١٧ سنة. يوفر DDL تسهيلات لتحديد مثل هذه القيودConstraints. يتحقق نظام قاعدة البيانات من هذه القيود في كل مرة يتم فيها تحديث قاعدة البيانات.. وبالتالي ، فإن أنظمة قواعد البيانات تطبق قيود التكامل التي يمكن اختبارها بأقل قدر من الجهد. وهي مثل أي لغة برمجة أخرى ، يحصل على بعض التعليمات (العبارات) كمدخلات ويولد بعض المخرجات. يتم وضع إخراج DDL في قاموس البيانات Data Dictionary، والذي يحتوي على بيانات وصفية Metadata - أي بيانات حول البيانات. و يعتبر قاموس البيانات نوعًا خاصًا من الجداول لا يمكن الوصول إليه وتحديثه إلا بواسطة نظام قاعدة البيانات نفسه (وليس مستخدمًا عاديًا). يستشير نظام قاعدة البيانات قاموس البيانات قبل قراءة البيانات الفعلية أو تعديلها.

٢- لغة معالجة البيانات **data-manipulation language (DML)** وهي لغة للتعبير عن استعلامات قاعدة البيانات والتحديثات. و تمكّن المستخدمين من الوصول إلى البيانات أو معالجتها كما ينظمها نموذج البيانات المناسب. ويكون الوصول على انواع:

- استرجاع المعلومات المخزنة في قاعدة البيانات
- إدراج معلومات جديدة في قاعدة البيانات

- حذف المعلومات من قاعدة البيانات
- تعديل المعلومات المخزنة في قاعدة البيانات

من الناحية العملية ، لا يعتبر تعريف البيانات ولغتا معالجة البيانات لغتين منفصلتين ؛ وبدلاً من ذلك ، فإنهم يشكلون ببساطة أجزاءً من لغة قاعدة بيانات واحدة ، مثل لغة SQL المستخدمة على نطاق واسع.

الاستعلام Query : هو بيان يطلب استرجاع المعلومات. يسمى الجزء من DML الذي يتضمن استرداد المعلومات بلغة الاستعلام. فمن الشائع استخدام مصطلحات لغة الاستعلام ولغة معالجة البيانات بشكل مترادف.

قيود المجال Domain Constraints : يجب أن يرتبط مجال القيم الممكنة بكل سمة (على سبيل المثال ، أنواع الأعداد الصحيحة وأنواع الأحرف وأنواع التاريخ / الوقت). الإعلان عن سمة لمجال معين بمثابة قيد على القيم التي يمكن أن تتخذها. قيود المجال هي الشكل الأساسي لقيد التكامل. يتم اختبارها بسهولة من قبل النظام كلما تم إدخال عنصر بيانات جديد في قاعدة البيانات.

التفويض او التخويل Authorization. قد نرغب في التمييز بين المستخدمين فيما يتعلق بنوع الوصول المسموح به على قيم البيانات المختلفة في قاعدة البيانات. يتم التعبير عن هذه الفروق من حيث التفويض ، وأكثرهأنواع التخويل شيوعًا هي:

-إذن القراءة ، الذي يسمح بقراءة البيانات وليس تعديلها
- إذن إدخال ، الذي يسمح بإدخال بيانات جديدة ، ولكن ليس تعديل البيانات الموجودة
- إذن التحديث ، الذي يسمح بتعديل البيانات وليس حذفها
- إذن الحذف ، الذي يسمح بحذف البيانات.

و يجوز لنا تخصيص كل أنواع التفويضات هذه أو لا شيء أو مجموعة منها.

نماذج البيانات Data Models

أساس هيكل قاعدة البيانات هو نموذج البيانات. وهي مجموعة من الأدوات المفاهيمية لوصف "البيانات ،العلاقات ما بين البيانات ودلالات البيانات وقيود الاتساق" ، من خلال نموذج البيانات يمكن توصيف تصميم قاعدة بيانات على المستويات المادية والمنطقية والعرضية. وتعتمد هذه النماذج على مكونات اساسية وهي :

١- الكيان Entity :هو أي شيء (شخص أو مكان أو شيء أو حدث) تتمحور حوله البيانات التي سيتم جمعها وتخزينها. كيان يمثل نوع معين من الأشياء في العالم الحقيقي. نظرًا لأن الكيان يمثل نوعًا معينًا من الكائنات ، فإن الكيانات هي "يمكن تمييزه" أي أن مكرراتoccurances لكل كيان تكون فريدة ومتميزة. على سبيل المثال ، سيكون لدى كيان العميل العديد من الأحداث المميزة للعملاء ، مثل أحمد ، حسين، رؤى، ، وما إلى ذلك. يجوز للكيانات هي أشياء مادية ، مثل العملاء أو المنتجات ، ولكن قد تكون الكيانات أيضًا مجرد أشياء مجردة ، مثل درجات الطلاب.

٢- الصفة Attribute :هي خاصية مميزة للكيان. على سبيل المثال ، يمكن وصف كيان العميل بسمات مثل "اسم العائلة" للعميل و "الاسم الأول" للعميل و"هاتف" العميل و"عنوان" العميل و"حد ائتمان" العميل. الصفات هي متمثلة بالحقول في أنظمة الملفات.

٣- العلاقة ما بين البيانات : Relationshipتصف العلاقة ارتباطًا بين الكيانات. على سبيل المثال ، توجد علاقة بين العملاء و يمكن وصف الوكلاء على النحو التالي: يمكن للوكيل أن يخدم العديد من العملاء ، ويمكن أن يخدم كل عميل واحد وكيل. تستخدم نماذج البيانات ثلاثة أنواع من العلاقات: علاقات رأس بأطراف ، علاقة أطراف بأطراف ، علاقات رأس برأس. مصممي قواعد

البيانات عادةً ما تستخدم الرموز المختصرة ١: M أو ١ .. * ، M: N أو * .. * ، و ١: ١ أو ١..١ ، على التوالي. (على الرغم من أن تدوين M: N هو تسمية قياسية لعلاقة أطراف بأطراف ، يمكن أيضًا استخدام التسمية M: M.) الأمثلة التالية توضيح الفروق بين الثلاثة.

حددت المناقشة السابقة كل علاقة في كلا الاتجاهين ؛ أي العلاقات ثنائية الاتجاه:
يمكن لعميل واحد إنشاء العديد من الفواتير.
يتم إنشاء كل من الفواتير العديدة بواسطة عميل واحد فقط.

القيد Constraint : هو قيد يتم وضعه على البيانات. تعتبر القيود مهمة لأنها تساعد في ضمان سلامة البيانات. يتم التعبير عن القيود عادة في شكل قواعد. فمثلا:
يجب أن يتراوح راتب الموظف بين ٦٠٠٠ و ٣٥٠,٠٠٠.
يجب أن يكون المعدل التراكمي للطالب بين ٠,٠٠ و ٤,٠٠.
يجب أن يكون لكل فصل مدرس واحد فقط.

نموذج علاقة الكيان (ER)**Entity Relationship(ER) Model:**

نموذج ER: هي تقنية رسومية لفهم البيانات وتنظيمها بشكل مستقل عن التنفيذ الفعلي لقاعدة البيانات ويستند هذا النموذج الى مجموعة من المفاهيم التي يتم التعبير عنها بادوات رسوم تخطيطية كما سنوردها تباعا.

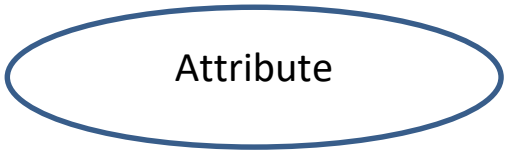
الكيان Entity: أي شيء له وجود مستقل ونجمع عنه بيانات. يُعرف أيضًا باسم نوع الكيان. في نموذج(ER)، يتم تمثيل الكيان بالشكل التالي:

مثيل الكيان Entity instance : هو عضو معين في نوع الكيان.
مثال لمثال الكيان: موظف معين

الكيان العادي Regular Entity : هو الكيان الذي له السمة الرئيسية الخاصة به هو كيان عادي. مثال للكيان العادي: الموظف.
الكيان الضعيف Weak entity : هو الكيان الذي يعتمد على كيان آخر في وجوده وليس لديه أي سمة رئيسية خاصة به هو كيان ضعيف.
مثال لكيان ضعيف: في علاقة أصل / طفل ، يُعتبر الوالد كيانًا قويًا والطفل كيانًا ضعيفًا. في نموذج ER ، وفي أدناه رسم للكيان الضعيف.

الصفات Attributes

الخصائص: وهي التي تصف الكيانات و يتم تمثيلها في نموذج (ER) من خلال الشكل البيضوي .



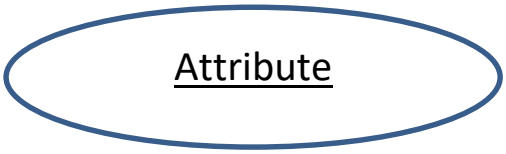
مجال الصفة attribute Domain

تسمى مجموعة القيم المحتملة التي يمكن أن تتخذها الصفة بمجال الصفة. على سبيل المثال : نأخذ الصفة "يوم " ، يمكن لصفة اليوم ان تأخذ أي قيمة من المجموعة التالية {السبت، الأحد،.....، الجمعة }. ومن ثم يمكن تسمية هذه المجموعة باسم مجال الصفة اليوم.

الصفة الرئيسية Attribute key أو المفتاح

وهي الصفة (أو مجموعة الصفات) الفريدة لكل مثيل كيان.

على سبيل المثال ، معرف الموظف للموظف (رقم الموظف)، ، إذا كانت الصفة الرئيسية تتكون من صفتين أو أكثر معًا ، فإنها تسمى المفتاح المركب. في نموذج (ER) ، يتم تمثيلها بالشكل التالي.



الصفة البسيطة Simple Attribute

إذا كان لا يمكن تقسيم الصفة إلى مكونات أبسط ، فهي صفة بسيطة.

مثال لسمة بسيطة: معرف الموظف للموظف.

الصفة المركبة Composite Attribute

إذا كان من الممكن تقسيم الصفة إلى مكونات ، فإنها تسمى صفة مركبة.

مثال على الصفة المركبة: اسم الموظف الذي يمكن تقسيمه إلى الاسم الأول والاسم الأوسط والاسم_الأخير.

الصفات ذات القيمة المفردة Single value Attribute

إذا كان بإمكان الصفة أن تأخذ قيمة واحدة فقط لكل مثيل كيان ، فهي صفة ذات قيمة واحدة.

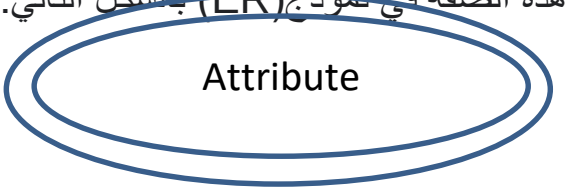
مثال على صفة ذات قيمة واحدة: عمر الطالب. يمكن أن يأخذ قيمة واحدة فقط لطالب معين.

الصفات متعددة القيم Multi value Attribute

إذا كان بإمكان الصفة أن تأخذ أكثر من قيمة واحدة لكل مثيل كيان ، فهي صفة متعددة القيم.

مثال للصفة متعددة القيم: رقم هاتف الموظف ، قد يكون للموظف عدة أرقام هواتف.

ويتم تمثيل هذه الصفة في نموذج(ER) بالشكل التالي:



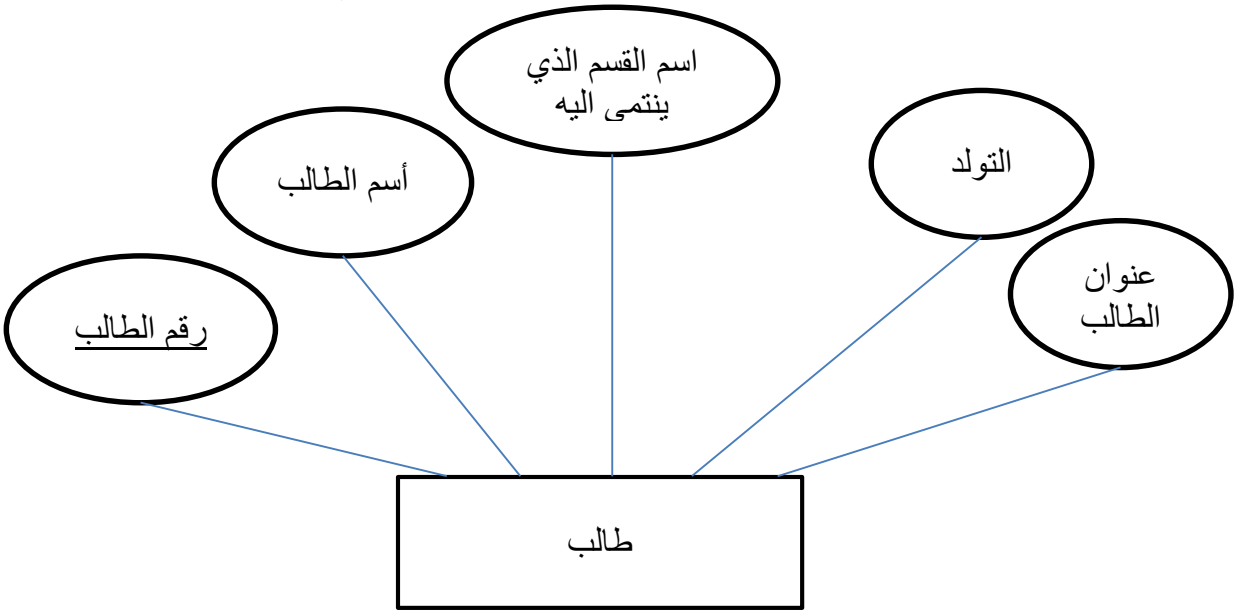
الصفة المخزنةStored Attribute

وهي الصفة التي يجب تخزينها بشكل دائم مثال على الصفة المخزنة: اسم الطالب

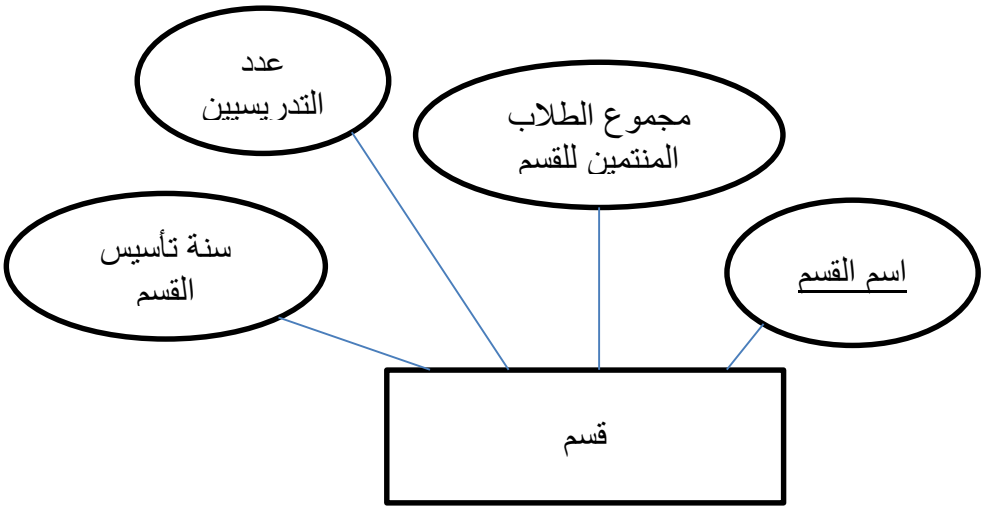
الصفة المشتقة Derived Attribute

وهي الصفة التي يمكن حسابها أو اشتقاقها بناءً على صفات أخرى هي صفة مشتقة.
مثال على الصفة المشتقة: عمر الموظف الذي يمكن حسابه من تاريخ الميلاد والتاريخ الحالي. ويتم تمثيلها في نموذج (ER) بالشكل التالي:

مثال: قاعدة بيانات كلية تتكون من كيانين هما "طالب" و "قسم"



(أ) كيان طالب

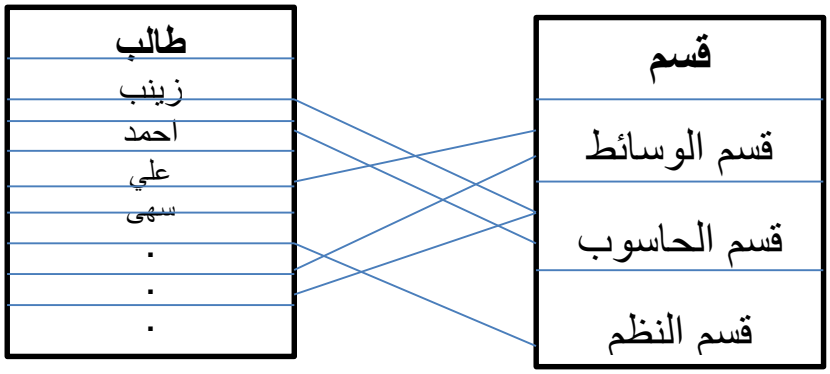


(ب) كيان قسم

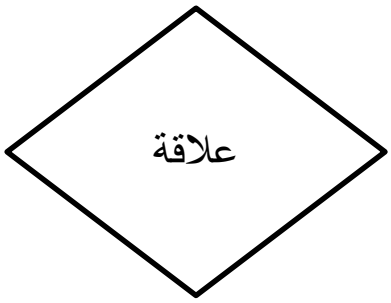
العلاقة Relationship

عندما تشير صفة من نوع كيان إلى نوع كيان آخر ، فهذ يعني وجود علاقة Relationship بينهما.

بالرجوع الى مثالنا في قاعد بيانات الكلية نجد ان صفة "اسم القسم الذي ينتمي اليه الطالب" في كيان طالب يشير إلى كيان "قسم" في نموذج (ER) ، وهذا يعني وجود علاقة مابين كيان "طالب" و "قسم" .



وسيتم تنقيح مخطط قاعدة بيانات لتمثيل العلاقات بشكل صريح. في التصميم الأولي لأنواع الكيانات ، يتم عادةً تسجيل العلاقات في شكل صفات. أثناء تحسين التصميم ، يتم تحويل هذه الصفات إلى علاقات بين أنواع الكيانات.
ويرمز للعلاقة في نموذج (ER).



أصول العلاقة(Cardinality of Relationship)

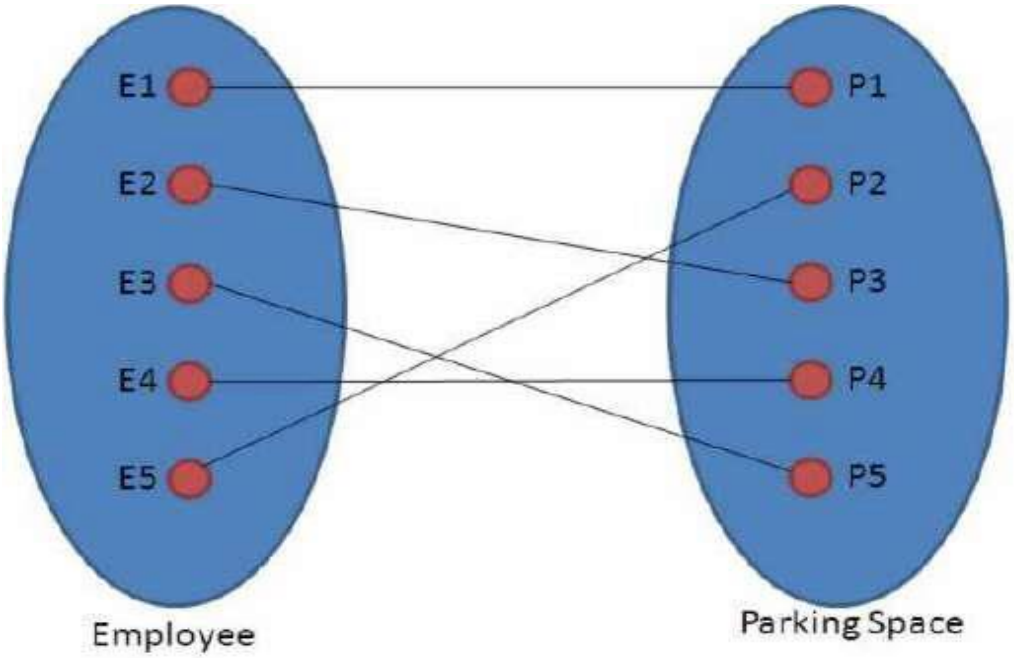
تحدد أصول العلاقة العدد المسموح به من كل نوع كيان. يمكن أن تحتوي العلاقات على أربعة انواع من الروابط على النحو المبين أدناه.

١. علاقة واحد لواحد (١: ١)

٢. علاقة واحد إلى متعدد (١: N)
٣. علاقة كثير لواحد (M: 1)
٤. علاقة كثير إلى كثير (M: N)

أمثلة

- **علاقة (١-١) : وهي علاقة الموظف مع موقف السيارة حيث ان كل موظف لديه سيارة له مكان واحد بالموقف والعكس صحيح**



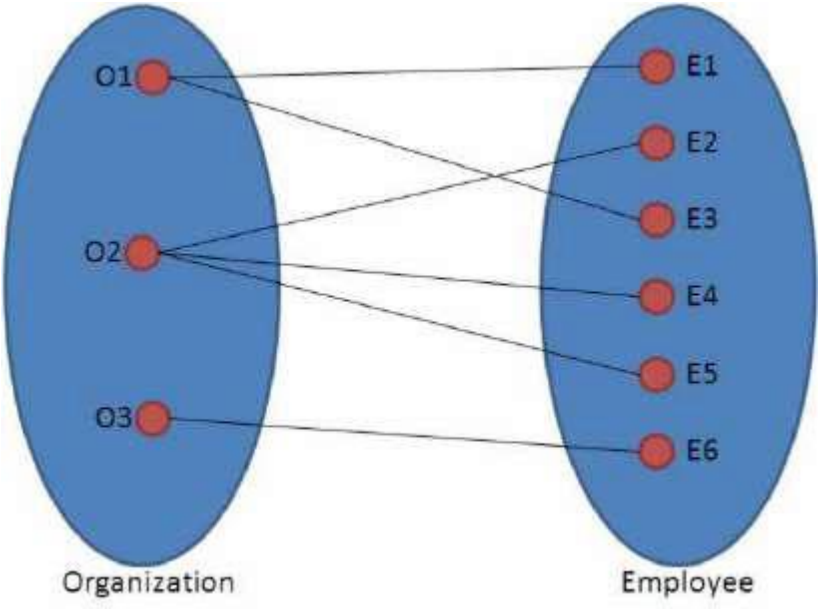
ويتم تمثيلها بنموذج(ER) كالآتي:

Employee موظف Parking space: موقف

Assigned withالسيارات : يخصص له

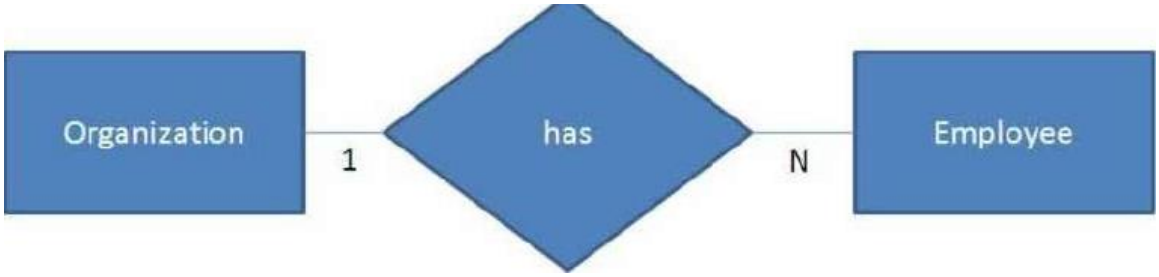
- **علاقة واحد إلى متعدد (١: N)**

والمثال التالي حول علاقة "موظف" و"مؤسسة" يعمل بها



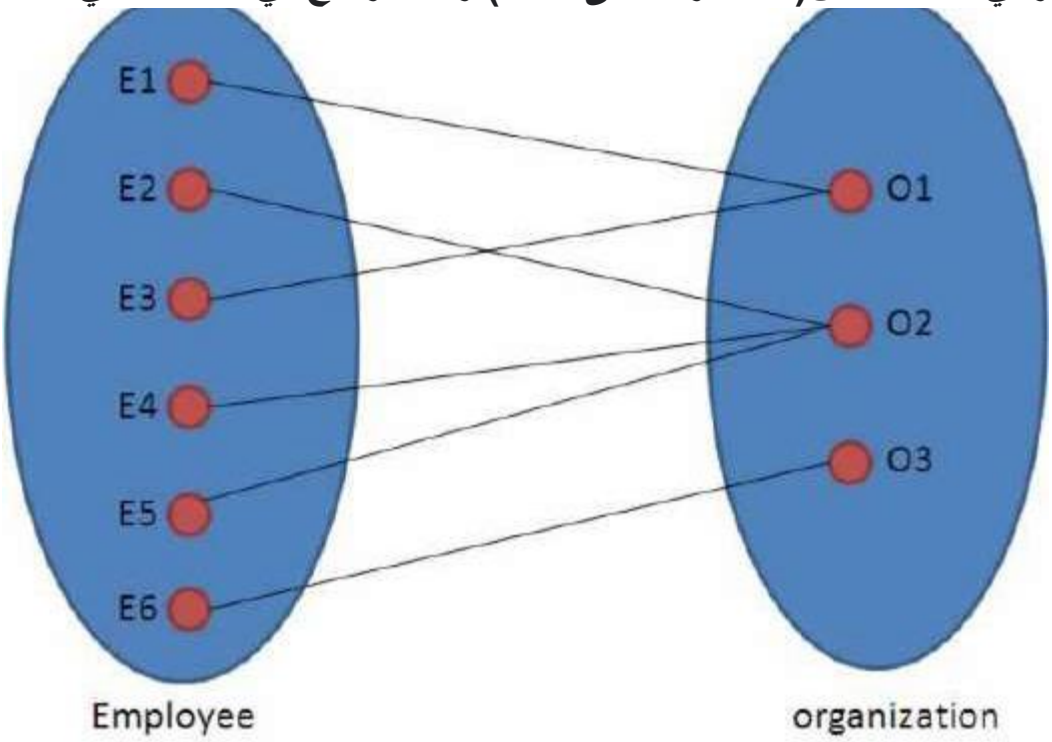
مؤسسة(Organization) **موظف(Employee)**

ويتم تمثيله في نموذج(ER) بالشكل التالي وأسم العلاقة بينهما هي (لها has) وتعني ان المؤسسة لها موظفين وتقرأ هذه العلاقة من اليسار الى اليمين. المؤسسة لها موظفين

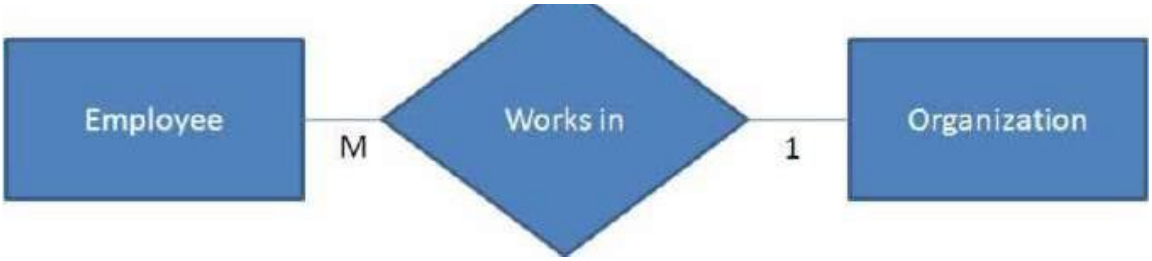


- علاقة كثير لواحد (M: 1)

وهي علاقة عكس(علاقة واحد الى متعدد) وكما موضح في المثال التالي:

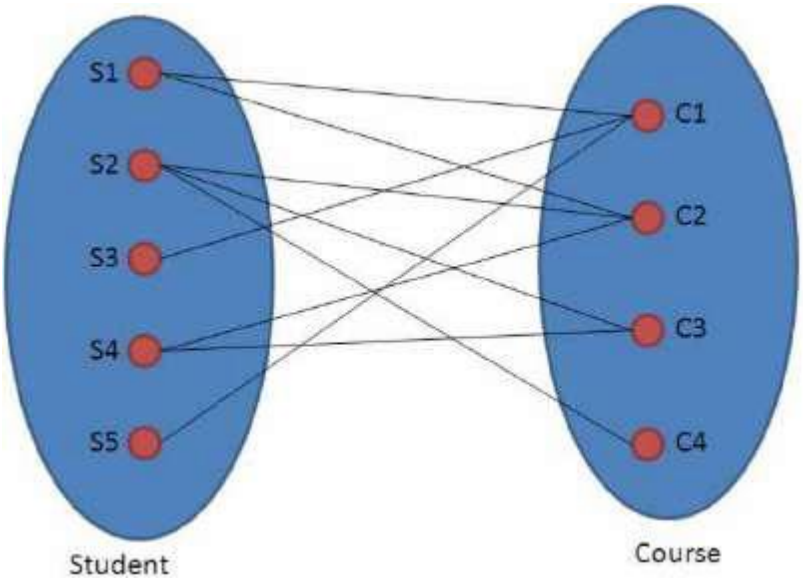


وتقرأ كالآتي: يوجد موظفين كثيرون يعملون في مؤسسة واحدة ويعبر عنها في علاقة (يعمل في work in) نموذج (ER) بالشكل التالي:



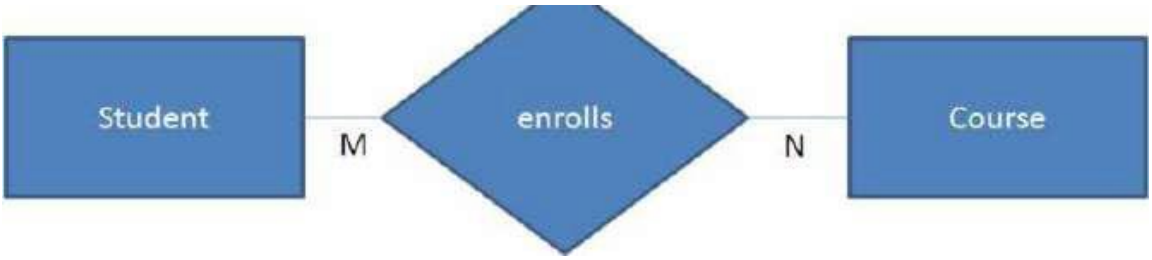
- **علاقة كثير إلى كثير (M: N)**

والمثال التالي حول علاقة طالب بالكورس فمن الممكن يدرس الطالب الواحد أكثر من كورس وكذلك ممكن الكورس الواحد يدرس فيه اكثر من طالب



الطالب Student **الكورس Course**

ويتم تمثيله في نموذج(ER) بالشكل التالي :



enroll يسجل او يلتحق

مزايا وعيوب نمذجة ER

المزايا

١. نمذجة ER بسيطة وسهلة الفهم.و يتم تمثيلها بلغة مستخدمي الأعمال ويمكن أن تكون كذلك لغير المتخصص ان يفهموها.

٢. حدسي اي انه تصوري ويساعد في إنشاء قاعدة البيانات المادية.

٣. يمكن أن تكون معممة ومتخصصة على أساس الاحتياجات.

٤. يمكن أن تساعد في تصميم قاعدة البيانات.

نماذج البيانات Data Model

في الماضي كانت قواعد البيانات المتعارف عليها هي : قواعد البيانات الشبكية و قواعد البيانات الهرمية . وظلت هذه الأنواع هي المستخدمة حتى ظهرت قواعد البيانات العلائقية ونظرا لقوة نظم إدارة قواعد البيانات العلائقية ، و لسهولة تصميمها و برمجتها و تعامل المستخدمين معها فقد طغت على الأنواع الأخرى وأصبحت هي النوع الوحيد المستخدم

النموذج العلائقيةRelational Model

يعد النموذج العلائقي هو نموذج البيانات الأساسي لتطبيقات معالجة البيانات التجارية. وحقق هذه المكانة بسبب بساطته ، مما يسهل عمل المبرمج ، مقارنة بنماذج البيانات الأخرى مثل نموذج الشبكة أو النموذج الهرمي.

ويعتمد النموذج العلائقي على تنظيم البيانات في جداول ، ويعرف الجدول بأنه يتكون من الصف والعمود ، حيث أن الصفوف تشكل السجلات ، أما الاعمدة (صفات الجدول) فتشكل الحقول ، وتتميز هذه الجداول بوجود علاقة فيما بينهما تمكن المستخدم من الوصول الى بيانات من مختلف أجزاء قاعدة البيانات. وهنالك بعض المفاهيم الخاصة بقواعد البيانات العلائقية ومنها مايلي

1 -العلاقة-: وهي ماتعرف بالجدول ، حيث يشكل الوحدة الاساسية في قواعد البيانات العلائقية ، وله عدة أنواع في قواعد البيانات العلائقية

2- .الصفة-: وهي عبارة عن أعمدة الجدول ، فأذا كان لدينا علاقة طلاب بها رقم الطالب ، اسم الطالب ، معدل الطالب ، فأننا نعتبر أن هذه البيانات الخاصة بالطلاب هي صفات العلاقة (جدول الطلاب)

3- . (درجة العلاقة-: وهي عدد الصفات (الحقول) داخل العلاقة ، فمثلاً اذا كان لدينا علاقة (جدول) به ثلاث صفات فيكون هذا الجدول من الدرجة الثالثة ، واذا كان به صفتين يكون من الدرجة الثانية وهكذا

4- . مجال القيم-: وهو المعيار الذي يتم فيه تحديد مجال القيم المسموح بأدخالها الى الجدول

مثال: (قاعدة بيانات مستشفى)

جدول المرضى

اسم المريض	رقم المريض	رقم الغرفة	الجنس	الطبيب
محمد	313	100	1	سيف
حنان	345	300	2	محمد
خالد	988	100	1	دعاء
منى	456	200	2	عزة

جدول الغرف

عدد الأسرة	رقم التحويلة	رقم الغرفة
3	435	100
2	342	200
1	676	300

جدول الأدوية

المصنع	اسم الدواء	رقم الدواء
HG	FDG	s123
AB	PANADOL	s153
AB	FIFA	s173

جدول يعالج بواسطة

الكمية	رقم الدواء	رقم المريض
3	s123	313
2	s153	345
1	s173	988

نلاحظ أن هذه الجداول يوجد بينهم علاقات ، لو أردنا أن نستعلم عن شيء معين داخل قاعدة البيانات ، فسيقوم الحاسب باسترجاعه عن طريق العلاقات التي بين تلك

الجداول . مثال : لو أردنا اسم المريض الذي يحمل رقم ٣١٣ ورقم الغرفة التي يرقد بها وتحويلة هذه الغرفة واسم الدواء الذي يتناوله ؟ أولا يستخرج الحاسب اسم المريض والغرفة التي يرقد بها من : جدول المرضى (اسم المريض: محمد ، الغرفة: ١٠٠) عن طريق رقم لطالب(وهو يمثل مايسمى بالمفتاح الرئيسي)

ثم ينتقل إلى جدول الغرف ليأخذ رقم التحويلة للغرفة ١٠٠ (التحويلة: ٤٣٥ ،)ثم ينتقل إلى جدول يعالج بواسطة ليأخذ رقم الدواء ،ومن ثم يتجه إلى جدول الأدوية ليأخذ اسم هذا الدواء.FDG

المفاتيح Keys

المفتاح: هو عبارة عن صفة أو أكثر مجتمعة معا الغرض منها تمييز البيانات عن بعضها مثلما راينا في مثالنا السابق حيث ميزنا البيانات الخاصة بكل مريض عن طريق رقم المريض .

انواع المفاتيح

-:**أولاً : المفتاح الرئيسي-: (PRIMARY KEY)** وهو المفتاح الذي يحدد بشكل وحيد ومنفرد بحيث يتميز عن غيرة ، فلا تتكرر قيمت في أكثر من سجل واحد ، ولا يقبل قيمة) NULL أي لايمكننا أن نترك الحقل فارغاً بدون قيمة

. ثانياً : المفتاح المركب أوالمجمع-: (COMPOSITE KEY) وهو المفتاح الذي يستخدم لتعريف السجل بشكل وحيد ومنفرد ، ولكنة يختلف عن المفتاح الرئيسي بأنه يشمل على أكثر من صفة

.**ثالثاً : المفتاح المرشح-: (CANDIDATE KEY)** عند البدء بتصميم الجدول يتم ترشيح عدد من الحقول (الصفات) كي تصبح مفاتيح رئيسية ، وعند ادخال البيانات ، قد يتبين أن هذه المفاتيح يمكن أن تأخذ قيمة NULL ، فالمفتاح الذي يأخذ قيمة NULL يستثنى ، والمفاتيح التي لاتأخذ قيمة NULL ولاتكرر تبقى وتصبح مفاتيح أساسية ، بمعنى أخر : فأن المفتاح المرشح هو الصفة أو مجموعة الصفات التي يتم اختيارها وفحصها حتى يتقرر فيما بعد أنها ستبقى مفاتيح مرشحة أويتم اعتمادها كمفتاح رئيسي.

رابعاً : المفتاح الاجنبي-: **(FOREIGN KEY)** وهو عبارة عن حقل (صفة) أو اكثر يستخدم للربط بين جدولين ، وسمي المفتاح الاجنبي بهذا الاسم لانه ليس من الحقول الموجودة أصلاً في الجدول ، أي انه عبارة عن حقل أو اكثر تضاف الى جدول لربطه مع جدول اخر. وكمثال على استخدام المفتاح الأجنبي نرى في مثالنا السابق حول قاعدة بيانات مستشفى ان صفة "رقم الغرفة" في جدول المريض هي مفتاح اجنبي لانها مفتاح رئيس في جدول اخر هو ".جدول الغرف".

مخطط العلاقة Relational Schema

يتوافق مفهوم مخطط العلاقة مع مفهوم لغة البرمجة لتعريف النوع. شكل عام ، يتكون مخطط العلاقة من قائمة الصفات والمجالات المقابلة لها. ومفهوم مثيل العلاقة (relational instance) يتوافق مع مفهوم لغة البرمجة لقيمة متغير. قيمة معطى قد يتغير المتغير مع مرور الوقت ؛

وبالمثل ، قد تتغير محتويات مثيل العلاقة مع مرور الوقت حيث يتم تحديث العلاقة. في المقابل ، لا يتغير مخطط العلاقة بشكل عام. على الرغم من أنه من المهم معرفة الفرق بين مخطط العلاقة ومثيل العلاقة ، فإننا غالبًا ما نستخدم نفس الاسم .

مثال لمخطط العلاقة:

Student(id,name,age)

وهي تعني هنالك علاقة(جدول) يحتوي على الصفات (id) ، (name) ، (age)

مثال لمثيل العلاقة:

Student(id=7631,name=Ali ,age=19)

وهي تعني ان هذا هو مثيل للعلاقة ويحتوي على بيانات طالب معين(Ali)

الجبر العلائقي Relational Algebra

الجبر العلائقي هو إحدى لغتي الاستعلام الرسميتين المرتبطين بالنموذج العلائقي. و هو لغة استعلام إجرائية ، تأخذ أمثلة(instances) من العلاقات كمدخلات وتنتج حالات من العلاقات كناتج. ويستخدم ادوات خاصة لاجراء عملياته تسمى بعوامل التشغيل(Operators) لإجراء الاستعلامات.

تتكون الاستعلامات في الجبر باستخدام مجموعة من العوامل. ومن العوامل الأساسية للجبر العلائقي هي:-

١-الأختيار Selection
٢-الأسقاط projection
٣- الأتحاد union
٤-المنتج التبادلي cross-product
٥- الفرق difference
٦- الأنضمام الطبيعي Natural Join
٧- اعادة التسمية Rename

Selection and Projection

يتضمن الجبر العلائقي عوامل التشغيل لتحديد(أختيار Selection) الصفوف من العلاقة ويستخدم لها رمز(σ)

ويكون عامل التشغيل (الأسقاطProjection بتحديد الأعمدة ويستخدم لها العامل رمز(π). وتسمح لنا هذه العمليات بمعالجة البيانات في علاقة واحدة.

لو أخذنا قاعدة البيانات التالية التي تتكون من جدولين هما (A,B)

```
A                                   B
 (Name   Age  Sex )                  (Id   Course)
 ------------------                  -------------
 Ram    14   M                        1     DS
 Sona   15   F                        2     DBMS
 kim    20   M
```

مثال (١)

$$\sigma_{sex="M"}(A)$$

نتيجة هذا التغبير تكون:
نفس الأعمدة الموجودة في جدول() ولكن الصفوف تختلف

Name	Age	sex
Ram	14	M
Kim	20	M

مثال(٢)

ماهي نتيجة التعبير التالي ؟

$$\sigma_{age<20}(A)$$

مثال(٣) : ماهي نتيجة التعبير التالي؟

$$\pi_{name,age}(A)$$

تكون نتيجة هذا التعبير هو: اختيار اعمدة(name, age) من الجدول (A) وكما يلي:

Name	Age
Ram	14
Sona	15
Kim	20

مثال(٤): أوجد نتيجة التعبير التالي:

$$\pi_{Id}(B)$$

النتيجة لهذا التعبير هي اختيار عمود (Id) من الجدول (B)

Id
1
2

واجب : أوجد نتيجة التعبير التالي:

$$\pi_{name,age}(\boldsymbol{\sigma_{sex="M"}(A)})$$

عمليات المجاميع (الأتحاد، التقاطع، الفرق ، الضرب)

الأتحاد

ويستخدم لها رمز الاتحاد (U) ، يُرجع علاقة جديدة تحتوي على جميع المثيلات(الصفوف في كلا الجدولين) المتحدات بشرط ان يكون كلا الجدولين متوافقين من ناحية الأعمدة ولاتوجد فيها تكرار

ليكن لدينا قاعدة البيانات متكونة من جدولين(علاقتين) A,B

A

S#	SNAME	SATUS	CITY
S1	Smith	20	London
S4	Clark	20	London

B

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	20	Paris

مثال(٥)

أوجد نتيجة التعبير التالي:

$$A \cup B$$

النتيجة سوف تكون:

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S4	Clark	20	London
S2	Jones	20	Paris

التقاطع

ويستخدم لها رمز الاتحاد (∩) ، يُرجع علاقة جديدة تحتوي على جميع المثيلات(الصفوف في كلا الجدولين) المشتركة بشرط ان يكون كلا الجدولين متوافقين من ناحية الأعمدة .

مثال(٦)

$$A \cap B$$

وتكون نتيجة التعبير هي :

S#	SNAME	STATUS	CITY
S1	Smith	20	London

لان هذا الصف موجود في كلا الجدولين

الفرق

ويستخدم لها رمز الاتحاد (−) ، يُرجع علاقة جديدة تحتوي على جميع المثيلات(الصفوف في الجدول الأول والغير موجودة في الجدول الثاني) بشرط ان يكون كلا الجدولين متوافقين من ناحية الأعمدة .

مثال(٧)
ماهي نتيجة التعبير التالي:
A-B
والنتيجة هي علاقة جديدة تحتوي على الصفوف الموجودة في الجدول الأول(A) والغير موجودة في الجدول الثاني(B)

S#	SNAME	STATUS	CITY
S4	Clark	20	London

المنتج التبادلي

ويستخدم لها رمز الضرب(X) وناتج عامل الضرب هو توافق لصفوف الجدول الاول وربطها مع صفوف الجدول الثاني

مثال(٨)

أذا كانت لديك قاعدة البيانات التالية :

```
A                                   B
 (Name    Age   Sex )                (Id    Course)
 ------------------                  -------------
 Ram     14    M                      1      DS
 Sona    15    F                      2      DBMS
 kim     20    M
```

أوجد نتيجة التعبير AXB

النتيجة ستكون :

Name	Age	Sex	Id	Course
Ram	14	M	1	DS
Sona	15	F	1	DS
Kim	20	M	1	DS
Ram	14	M	2	DBMS
Sona	15	F	2	DBMS
Kim	20	M	2	BMS

الأنضمام الطبيعي

هذه العملية تمثل بالرمز (⋈) وتعني: إنشاء علاقة جديدة معتمدة على وجود صفات مشتركة في العلاقات(الجداول) المنضمة مع بعضها، والعلاقة الجديدة تحتوي فقط على تلك الصفوف التي تتساوى فيها القيم الموجودة في الصفتين المشتركتين في كلا الجدولين.

مثال (١٠)
لوكانت لدينا قاعدة البيانات التي تحتوي على علاقتين (جدولين) هما (Employee) و (Parking)

Employee		
Emp_id	Emp_name	Emp_office
1001	Bob	10
1002	Alice	11
1003	Sandy	10
1004	Larry	11
1005	Susan	11

Parking		
Emp_id	Parking_lot	Parking_space
1001	A	6
1002	A	14
1003	B	17
1004	B	6
1005	A	12

المطلوب : أوجد نتيجة التعبير التالي: $Emploee \bowtie Parking$

النتيجة هي:

Emp_id	Emp_name	Emp_office	Parking_lot	Parking_space
1001	Bob	10	A	6
1002	Alice	11	A	14
1003	Sandy	10	B	17
1004	Larry	11	B	6
1005	Susan	11	A	12

اعادة التسمية

تعتبر نتائج الجبر العلائقي هي أيضًا علاقات(جداول) ولكن بدون اسم. ولذل نستخدم عامل(التسمية) بإعادة تسمية علاقة الإخراج. ويستخدم لهذه العملية الرمز (ρ)

مثال(١١) : بالعودة الى المثال السابق(١٠) نجد ان نتاج العملية هو جدول بون أسم لذلك سنعيد التعبير السابق بعد ادخال عامل اعادة التسمية باستخدام التعبير التالي:

$$\rho_X(Emploee \bowtie Parking)$$

هذا الأيعاز سوف يعطي تسمية لناتج التعبير والذي هو الجدول التالي أسم (X)

Emp_id	Emp_name	Emp_office	Parking_lot	Parking_space
1001	Bob	10	A	6
1002	Alice	11	A	14
1003	Sandy	10	B	17
1004	Larry	11	B	6
1005	Susan	11	A	12

لغة SQL

هي اختصارٌ لـ Structured Query language وترجمتها هي "لغة الاستعلام البنائية. ان لغة SQL هي لغة قاعدة بيانات شاملة: تحتوي على عبارات لتعريفات البيانات ، الاستفسارات والتحديثات. وهي لغة برمجة غير إجرائية، وهي بذلك تختلف عن لغات البرمجة المعتادة مثل سي أو جافا ، حيث أن اللغات غير الإجرائية هي لغات متخصصة ولذلك فإن لغة الاستعلامات البنائية هي لغة للتعامل والتحكم مع قواعد البيانات المترابطة من خلال التعامل مع تراكيب البيانات وإجراء عمليات إدخال البيانات والحذف والفرز والبحث والتصفية والتعديل وخلافه.

ويمكن باستخدام لغة SQL أنجاز الفعاليات التالية:

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

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

- **لغة التعامل مع البيانات :Data Manipulation Language** 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** 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** تساعد هذه المجموعة من الأوامر في تحديد الصلاحيات التي يمكن منحها أو سلبها من المستخدمين الموجودين في قاعدة البيانات.

<u>ملاحظة:</u> تستخدم SQL مصطلحات (جدول Table) و(الصف row)و (العمودcolumn).

مثال على الجدول:

جدول اسمه(employee) يتكون من خمسة أعمدة(سمات)

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

مثال على الصف (السجل)

1	Ramesh	32	Ahmedabad	2000.00

مثال على العمود(السمة)

```
+-----------+
| ADDRESS   |
+-----------+
| Ahmedabad |
| Delhi     |
| Kota      |
| Mumbai    |
| Bhopal    |
| MP        |
| Indore    |
+----+------+
```

جملة إنشاء قاعدة البيانات

الصيغة العامة لأمر إنشاء قاعدة البيانات هي كالتالي:

```
CREATE DATABASE أسم قاعدة البيانات;
```

مثال:

```
CREATE DATABASE company;
```

وهذا الأمر يقوم بأنشاء قاعدة بيانات جديدة أسمها " company"

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 .
ويجب تحديد قاعدة البيانات التي سيتم انشاء الجداول فيها من خلال امر "USE‘" وكالأتي:

```
USE أسم قاعدة البيانات;
```

.ثم بعد ذلك ينفذ أمر أنشاء جدول البيانات حسب الصيغة التالية :

```
CREATE TABLE أسم الجدول الأول (
   اسم العمود ١ datatype [constraint],
   اسم العمود ٢  datatype [constraint],
   اسم العمود ٣ datatype [constraint],
  ....
);

CREATE TABLE أسم الجدول الثاني (.....
)
وهكذا لبقية الجداول التي يراد انشاءها داخل قاعدة البيانات
```

توضيح :

```
CREATE TABLE أسم الجدول (
```

- CREATEتعني إنشاء العنصر (الجدول هنا) وهي بداية الأمر.
- TABLEلتحديد أن هذه الجملة لإنشاء جدول.
- أسم الجدول: وهو 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.

```
أسم العمود ١ datatype [constraint],
```

- أسم العمود ١ :هو الاسم الذي سوف نعطيه للعمود الأول.
- :Datatypeيعني 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 بقاصلة منقوطة.

القيودConstraints

هي القواعد التي يتم فرضها على أعمدة البيانات في الجدول. تُستخدم لتحديد نوع البيانات التي يمكن أن تتواجد في الجدولمن أجل ضمان دقة وموثوقية البيانات في قاعدة البيانات.

- يمكن أن تكون القيود على مستوى العمود أو مستوى الجدول. يتم تطبيق قيود مستوى العمود على عمود واحد فقط ، بينمايتم تطبيق قيود مستوى الجدول على الجدول بأكمله. وفيما يلي القيود الشائعة الاستخدام المتوفرة في SQL:
- NOT NULL: يضمن أن العمود لا يمكن أن يحتوي على قيمة NULL.
- القيد "الافتراضي": يوفر قيمة افتراضية للعمود عندما لا يتم تحديد أي منها.
- قيد "القيمة الفريدة" يضمن أن جميع القيم في العمود مختلفة.
- قيد المفتاح الأساسي: يتم تحديد كل صف / سجلات في جدول قاعدة البيانات بشكل فريد.
- مفتاح خارجي: تحديد صفوف / سجلات بشكل فريد في أي جدول قاعدة بيانات آخر.
- قيد التحقق: يضمن قيد التحقق أن جميع القيم الموجودة في عمود تفي بشروط معينة.

جملة أدخال البيانات الى الجدول

يمكن ملأ الجدول بالبيانات باستخدام جملة الإضافة (INSERT) وصيغتها كالأتي:

```
INSERT TABLE أسم الجدول (اسم العمود١، اسم العمود ٢، .....أسم العمود )
VALUES(القيمة ١، القيمة ٢، .....القيمة الأخيرة) (الأخير);
```

حيث يتم أضافة صف من البيانات الى الجدول

مثال

```
INSERT TABLE employee(ID, NAME, AGE, ADRESS,
SALARY) VALUES(6, Komal, 22, MP, 4500.00 );
```

جملة تعديل الجدول

جملة تعديل الجدول يمكن استخدامها 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.

إضافة عمود

```
ALTER TABLE أسم الجدول ADD أسم العمود datatype;
```

حذف عمود

```
ALTER TABLE أسم الجدول DROP COLUMN أسم العمود;
```

تعديل عمود

```
ALTER TABLE أسم الجدول MODIFY | ALTER COLUMN أسم العمود datatype;
```

جملة الحذف DELETE:

يتم استخدام أمر DELETE لحذف الصفوف(السجلات) الموجودة في الجدول. ويمكن استخدام جملة WHERE مع DELETE لحذف صفوف محددة ، وإلا ستكون جميع السجلات تم الحذف. وصيغة هذا الأمر هي:

DELETE FROM أسم الجدول WHERE [condition];

مثال

DELETE FROM employee WHERE [age<10];

حسب هذا الأمر سوف تحذف جميع الصفوف التي يكون فيها قيمة العمود(age) أقل من ١٠

مثال:

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   |
+-----------+--------------+--------------+------------+--------+
```

ويمكن ملء هذا الجدول الفارغ بأستخدام أمر"INSERT"

ولإنشاء 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**

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

```
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);
```

عبارة SELECT

جملة او عبارة SELECT من الجمل الأساسية المستخدمة في لغة SQLوهي ذات اهمية كبيرة في استخراج المعلومات(Queryالأستعلام) من قاعدة البيانات ، وتتكون من الأجزاء الثلاثة: SELECT و FROM و WHERE ولها الشكل التالي:

SELECT <attribute list>
FROM <table list>
WHERE <condition>;

١- attribute list: وتعني قائمة بأسماء الصفات (الأعمدة) المطلوب استخراجها
٢- table list: قائمة بأسماء الجداول التي ستستخرج منها الصفات اعلاه
٣- Condition: وهو الشرط او القيد الموضوع على البيانات التي ستستخرج

في SQL ، عوامل المقارنة المنطقية الأساسية لمقارنة قيم الصفات مع بعضها البعض ومع الثوابت الحرفية هي = ، <، <= ،=> ،> = ، و <>. هذه تتوافق إلى عوامل الجبر العلائقية = و <و و> و على التوالي وإلى عوامل تشغيل لغة البرمجة C / C ++ = ، <، <= ،=> ،> = ، و! =. النحوي الرئيسي الاختلاف هو عامل التشغيل لا يساوي. SQL لديها عوامل مقارنة إضافية سوف نقدم تدريجيا. هذه الأدوات يمكن استخدامها في صياغة الشرط(Condition).

استعلام Query(١)

استرجع تاريخ الميلاد وعنوان الموظف من جدول(employee) الذي يحمل اسمه "جون ب. سميث"

الحل:

SELECT Bdate, Address
FROM EMPLOYEE

WHERE Fname = ‘John’ **AND** Minit = ‘B’ **AND** Lname = ‘Smith’;

حيث : تاريخ الميلاد اسمها (Bdate) و العنوان (Address) والأسم الأول(Fname) والأسم الوسط (Minit) والأسم الأخير (Lname).

استعلام Query ٢.

استرجع اسم name وعنوان address جميع الموظفينemployees الذين يعملون لدى قسم "البحث" ‘Research’

الحل

SELECT Fname, Lname, Address
FROM EMPLOYEE, DEPARTMENT
WHERE Dname = ‘Research’ **AND** Dnumber = Dno;

حيث: الأسم الأول(Fname) والأسم الأخير(Lname) والعنوان (Address) واسم القسم (Dname) ورقم الموظف في جدول الموظفين (Dnumber) ورقم الموظف في جدول الأقسام هو (Dno).

ولومثلنا الجداول التي عمل عليها هذا الأستعلام ستكون بالشكل التالي

EMPLOYEE

Dnumber	Fname	Lname	Address

DEPARTMENT

Dname	Dno

استعلام ٣

أستخرج الرواتب(SALARY) مرتبة من الجدول التالي

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

الحل

```
SELECT SALARY FROM CUSTOMERS ORDER BY SALARY;
```

ملاحظة: وجود (ORDER BY) في هذا الأستعلام تعني رتب البيانات المستخرجة حسب ماورد بعدها وهي (SALARY).

استعلام ٤

استخرج كل محتويات جدول(EMPLOYEE)

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

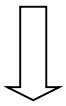
الحل

```
SELECT *  FROM EMPLOYEE
```

ان (*) هي تشير الى الكل أي انها افتراضية عندما تكون موجودة فهذا يعني انه لايوجد تحديد على استخراج والحصول على جميع المعلومات الموجودة في الجدول

أستعلام ٥

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi



```
SELECT *
FROM   Product
WHERE  Category = 'Gadgets'
```

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks

استعلام ٦

استخلص المعلومات للاعمدة (actor, birth, movie) من الجدولين المرتبطين (Role, Person) حسب قاعدة البيانات التالية:

Role

actor	movie	persona
Ben Affleck	Argo	Tony Mendez
Alan Arkin	Argo	Lester Siegel
Ben Affleck	The Company Men	Bobby Walker
Tommy Lee Jones	The Company Men	Gene McClary

Person

name	birth	city
Ben Affleck	1972	Berkeley
Alan Arkin	1934	New York
Tommy Lee Jones	1946	San Saba

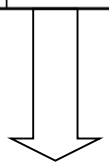
الحل

SELECT actor, birth, movie FROM Role, Person WHERE actor = name and birth > 1940;

استعلام ٧

في هذا الأستعلام استخلص بيانات الأعمدة الثلاثة (Pname, Price, Manufacturer) الذين ينطبق عليهم شرط ان تكون قيمة العمود(Category) هي (Gadgets)

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi



```
SELECT Pname, Price, Manufacturer
FROM   Product
WHERE  Category = 'Gadgets'
```

PName	Price	Manufacturer
Gizmo	$19.99	GizmoWorks
Powergizmo	$29.99	GizmoWorks

سؤال: حول هذه الأستعلام الى صيغة الجبر العلائقي؟؟

جملتي (GRANT) و (REVOKE)

تستخدم هذه الجمل من قبل لغة (SQL) لمنح او ابطال صلاحيات الوصول الى قاعدة البيانات وأجراء فعاليات مختلفة عليها والغرض منها هو الحفاظ على قاعدة البيانات من الأستخدام الغير مرغوب فيه او خرق السرية والآمنية لها :

- **الأمر GRANT**

يستخدم هذا الأمر لمنح صلاحيات معينة ويأخذ الصيغة التالية:

GRANT <Rights> ON <Object> TO <Users>;

وتمثل (Rights) الصلاحيات المعطاة مثل(قراءة واسترجاع بيانات، تحديث، حذف، الى اخره من هذه الفعاليات)

وتمثل (Object) هو مصدر البيانات التي يمنح عليها الصلاحية

وتمثل (Users) هم المستخدمون الذين تمنح لهم الصلاحيات

مثال ١

اعط صلاحيات القراءة والأدخال على الجدول(COURSES) للمستخدمين (BRASS , SPRING).

```
GRANT SELECT, INSERT ON COURSES TO BRASS,
SPRING;
```

مثال ٢

أعط صلاحية قراءة واسترجاع بيانات الجدول(COURSES) للجميع

```
GRANT SELECT ON COURSES TO PUBLIC;
```

- **الأمر REVOKE**

يستخدم هذا الأمر لأبطال صلاحيات معينة قد تم منحها سابقا ويأخذ الصيغة التالية:

REVOKE <Rights> ON <Object> FROM <Users>;

وتمثل (Rights) الصلاحيات الي سيتم ابطالها مثل(قراءة واسترجاع بيانات، تحديث، حذف، الى اخره من هذه الفعاليات)

وتمثل (Object) هو مصدر البيانات التي سوف يتم ابطال الصلاحية عليها

وتمثل (Users) هم المستخدمون الذين تبطل الصلاحيات الممنوحة لهم مسبقا

مثال

في المثال التالي سوف يتم ابطال صلاحية ادخال البيانات (INSERT) على الجدول(COURSES) من قبل المستخدم (BRASS)

```
REVOKE INSERT ON COURSES FROM BRASS;
```