

التحويل من مخطط الكيان والعلاقة الرابطة ERd الى جداول

خطوات بناء قواعد البيانات

رسم نموذج الكيان و العلاقة
الرابطة (ER Diagram)

المرحلة الأولى



تحويل نموذج الكيان والعلاقة الرابطة الى جداول
(Database Schema)

المرحلة الثانية



تخزين الجداول في نظام إدارة
قواعد البيانات DBMS

المرحلة الثالثة

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:

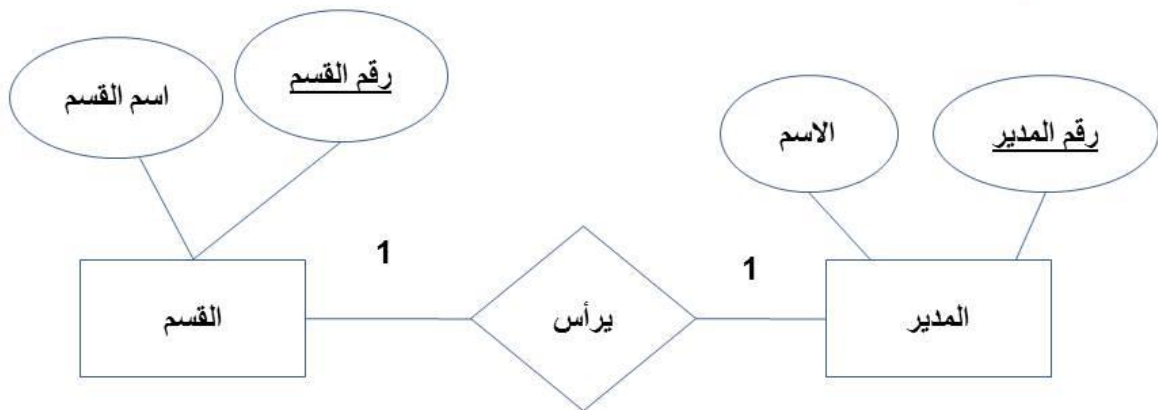
1- كل كيان في نموذج ERD يتحول إلى جدول ، ويكون اسم الجدول بنفس اسم الكيان .

2- خصائص الكيان تصبح هي حقول الجدول أي عناوين أعمدته ، ولا ننسى أن نضع خطأً 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:



أولاً :

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 :

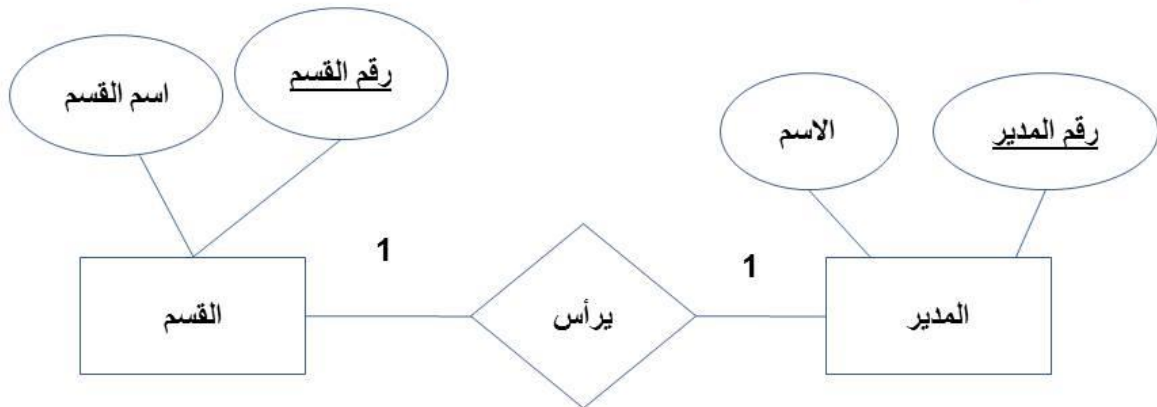
- الحالة الأولى عندما تكون العلاقة بين الكيانيين هي واحد إلى واحد :

فأننا عند تحويلها إلى جداول فإن المفتاح الأساسي لأحد الجدولين

يظهر كحقل إضافي عند الجدول الآخر – تحديد المفتاح اختياري-

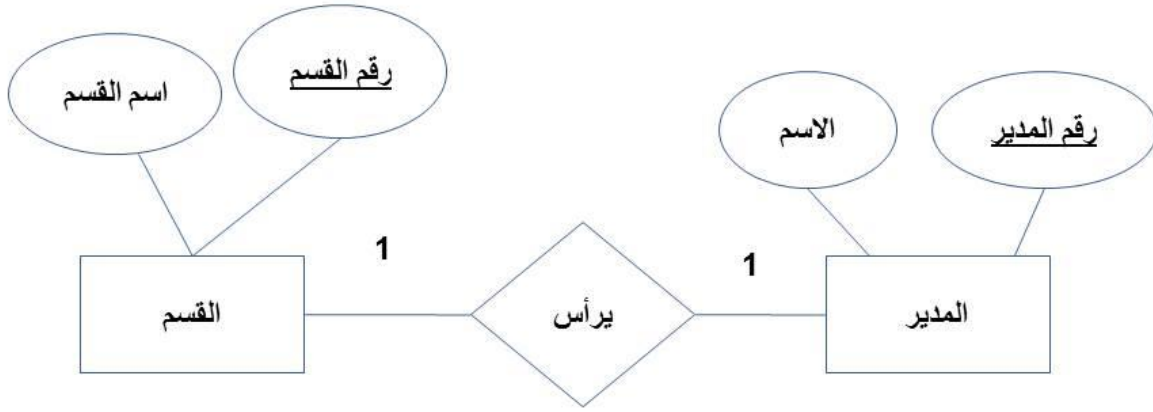
، ويسمى مفتاح أجنبي لأنه مفتاح أساسي لجدول وظهر في جدول آخر ونضع تحته خط متقطع .

مثال : حول نموذج الكيان والعلاقة الرابطة التالي إلى جداول:



أولاً :

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 خيارى :

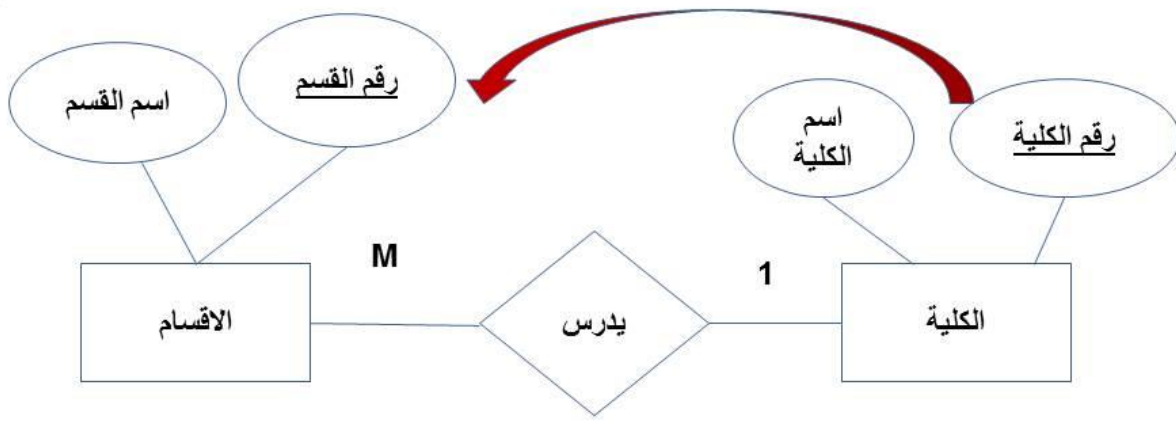
المدير (رقم المدير, الاسم)

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 إجبارى .



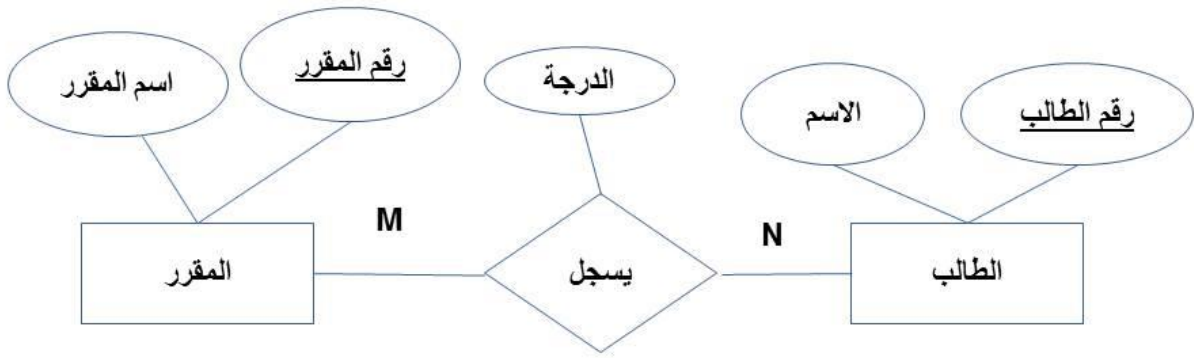
أولاً : نحول الكيانات إلى جداول :
 الكلية (رقم الكلية, اسم الكلية)
 الأقسام (رقم القسم, اسم القسم)

ثانياً : 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 الكيانيين هي متعدد إلى متعدد :

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



أولاً : نحول الكيانات إلى جداول :
 الطالب (رقم الطالب , الاسم)
 المقرر (رقم المقرر , اسم المقرر)

ثانياً : نربط بين الجدولين :
 الطالب (رقم الطالب , الاسم)
 المقرر (رقم المقرر , اسم المقرر)
 التسجيل (رقم الطالب , رقم المقرر , الدرجة)

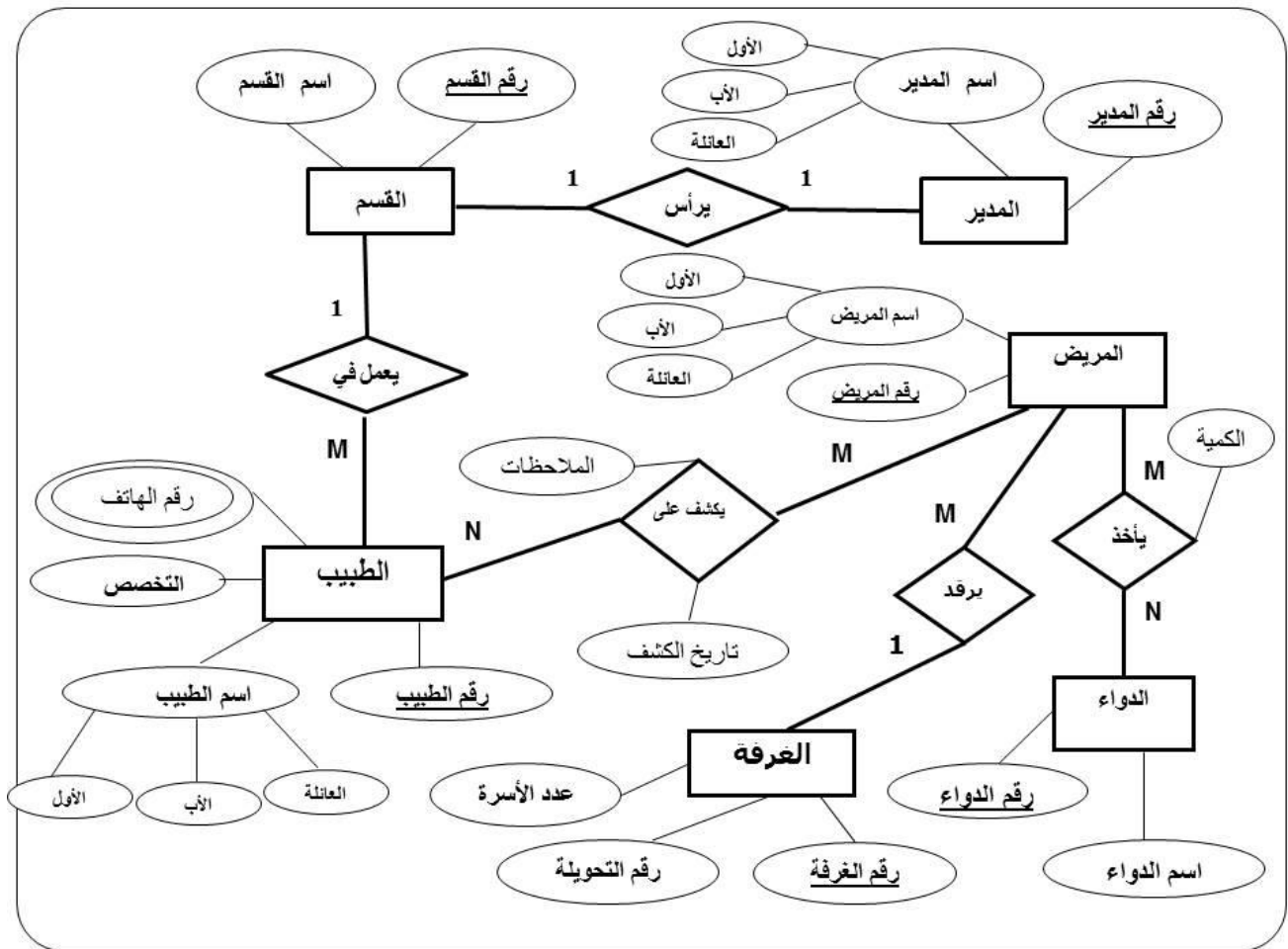
التحويل من مخطط الكيان و العلاقة الرابطة ERD إلى جداول :

❖ 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
 لاحقاً .



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 اختصاراً 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 نقوم بالآتي:

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 أوامر SQL :

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 سجلات، حذفها أو تعديلها.
- **لغة تعريف البيانات: Data Definition Language (DDL)** تُقدم الأوامر التي تدرج 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 تُمكن المبرمج من إنشاء قاعدة البيانات الخاصة به وتنفيذ الجمل

الخاصة بإدارتها وإدارة بياناتها.

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

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 المُمرَّر (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 جداول Tables ، مشاهد Views ، دوال وحزم، وإضافة سجلات وبيانات إلى الجداول المنشأة.
- أثناء تنفيذ أمر إنشاء قاعدة البيانات وبعده، تُنشأ ملفات خاصة بقاعدة البيانات الجديدة حسب النظام المستخدم، وتُدار هذه الملفات وتُسمَّى إما تلقائياً 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 ضرورياً في أغلب برامج إدارة قواعد البيانات، إلا أنها عادة في التوثيق Documentations لتمييز الكلمات المفتاحية التي تعد جزءاً 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 تتمكن من البدء في تنفيذ أوامر على القاعدة إلى تحديد قاعدة البيانات التي تريد العمل عليها، أي تلك التي ستُنشئ الجداول فيها. تختلف طريقة تحديد قاعدة البيانات حسب طريقة الاتصال ببرنامج إدارة قواعد البيانات:

- في سطر أوامر MySQL يُنفذ 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 (
```

- CREATE تعني إنشاء العنصر (الجدول هنا) وهي بداية الأمر.
- TABLE لتحديد أن هذه الجملة لإنشاء جدول.
- table_name 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.

column1 datatype [constraint],

- column1 هو الاسم الذي سوف نعطيه للعمود الأول.
- Datatype يعني نوع العمود (نصي, رقم, تاريخ, الخ).
- [Constraint] تعني – اختياريًا – 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 عموماً بحرف وليس برقم ، ومن الممكن أن تُتبع بعد ذلك بالأرقام.
- يُفضّل ألا يتجاوز طول اسم الجدول أو العمود ٣٠ حرفاً Characters ، حيث إن بعض النظم تمنع أن تتجاوز ذلك مثل نظام إدارة قواعد البيانات أوراكل.
- يجب ألا تُستخدم 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 مهماً للمبرمج ومسؤول قواعد البيانات إذ يُنفَّذ باستمرار أثناء وفي بداية بناء النظم البرمجية والبرامج نظراً لتغير المتطلبات وعدم اكتمالها.

تُستخدم جملة تعديل الجدول Alter Table عموماً 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، ننفذ 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 النوع int ، أي أن قيم هذا العمود يجب أن تكون أرقامًا؛
- الأعمدة City ، Last_Name ، First_Name و Address التي هي من النوع varchar ، أي سلسلة محارف، بطول 255 حرفًا.

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، قيوداً على مستوى أعمدة الجدول، بحيث يُعرّف العمود PersonID بأنه المفتاح الرئيسي للجدول، والأعمدة Last_Name و Address بأنها لا تستقبل القيم الفارغة.

في حال أردنا أن نضيف عموداً جديداً 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) في SQL :

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 لذلك UPDATE، INSERT وDELETE.

تغير جمل التعامل مع البيانات على البيانات في الجدول، لذلك فإن مسؤول قواعد البيانات لا بد أن يدير الصلاحيات اللازمة للتغيير على البيانات بطريقة مثلى. لا بد التنويه إلى أن جمل التعامل مع البيانات (UPDATE، INSERT وDELETE) لا تُرجع بيانات أو مخرجات 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، لا بد 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.

نُقدم 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: تُكْتَب النصوص في بعض أنظمة قواعد البيانات بين علامتي التنصيص المزدوجة. فمثلاً لكتابة اسم Ibrahim في جملة الإضافة نكتبها هكذا "Ibrahim" ، بينما توجد أنظمة أخرى تعتمد التعامل مع النصوص بين علامتي تنصيص مفردة، 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 غالباً 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 الجملتين السابقتين 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 Age مع وجود قيد المفتاح الرئيسي على العمود Person_Id وخاصية Auto Increment عليه، فإن أي جملة إضافة لا تشتمل على قيم ل First_Name أو Age سوف تُظهر خطأ عند تنفيذها، ولكن لأننا لم نضف قيد العمود غير الفارغ على عمود Last_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 NULL أو 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. تُمثل column1 و column2 أسماء 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 الشديدتين عند تنفيذ جملة التعديل، حيث إن جملة الشرط (WHERE condition) تحدد السجلات التي سيعُدّل 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 يُستخدم عمود قيد المفتاح الرئيسي في الجدول. فمثلا، إذا أردنا أن نُعدل قيمة العمر Age من 31 إلى 21 للشخص صاحب الرقم 101 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 رقم Person_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 105 في 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:

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

الصيغة العامة لجملة الحذف:

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

لا بد من الانتباه عند تنفيذ جملة الحذف، حيث إنه في حالة عدم تحديد جملة الشرط، فإن كل البيانات في الجدول سٌحذَف.

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 الجملتين التاليتين:

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

ملاحظة: العلامة * تعني جميع السجلات .

الاستعلام عن البيانات في SQL

تعرفنا 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 تأتي 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]  
[ORDER BY order_by_expression];
```

في بداية كل جملة استعلام نكتب كلمة (SELECT) جملة SQL غير حساسة لحالة الأحرف) ومن ثم نكتبها 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 ترشح 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 تصاعدياً أو تنازلياً 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 ستظهر 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، فمثلاً، 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 السجلات 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 السجلات 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 جدول Persons :

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 جملة Where :

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

العملية الوصف مثال:

العملية	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 <u>نمط</u>	First_Name LIKE "%lbr%"
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 NOT للنفى (،) AND (وجوب تحقق جميع الشروط) أو OR يكفي تحقق شرط واحد من الشروط. (

ترتيب السجلات:

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

ترتّب الجملة السجلات تصاعدياً وهو الخيار المبدئي، ولترتيبها تنازلياً 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 عمود First_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