الفصل الثامن
إنشاء الجداول ومعالجتها
Creating &maintaining tables

أخيرا، سوف نقوم بتكوين الجداول التي عملنا معها عدة فصول سابقة، ولكننا الآن سوف نقوم بتكوينها بأنفسنا، أليس ذلك شيقا؟ ولأننا سنقوم بتكوين الجداول أو إنشائها ان أردت، فيجب أولا أن نقوم بإنشاء قاعدة البيانات التي ستحتوي تلك الجداول. وعلى ذلك فإننا فى هذا الفصل سوف نتعلم الآتي:

- *إنشاء أو تكوين قاعدة بيانات*
- *إنشاء جداول وتعديلها والغاؤها.*
- *تكوين جدول جديد من جدول موجود*
- *حذف قاعدة بيانات*

obeikandl.com

طبعا سوف نتعرض لعدد من الأدوات والخصائص والاختيارات المصاحبة لتكوين قواعد البيانات وأيضا الجداول ، وقد حرصت أن يأتي هذا الفصل بعد عدد من الفصول التي تعاملنا فيها مع قواعد البيانات والجداول، لكي يصبح الأمر شيقا من ناحية ، ومن ناحية أخرى فان تعاملك السابق مع قواعد البيانات سوف يعطيك فهما أكبر لما نحن مقدمون عليه في هذا الفصل

إنشاء قاعدة بيانات Create Database

رغم أن هناك العديد من الخصائص والمهام التي يمكن إضافتها أثناء إنشاء أي قاعدة بيانات إلا أنك ستجد في معظم الأحيان أنك لن تحتاج إلا إلى تعليمات إنشاء قاعدة البيانات في أبسط صورة كما يلي :

الشكل العام لجملة إنشاء قاعدة البيانات كما يلي :

```
CREATE DATABASE database_name
```

مثال :

لإنشاء قاعدة بيانات باسم **payroll** استخدم المثال الآتي :

```
Create database payroll;
```

ولن نعلق طويلا على باقي التعليمات التي يمكن أن تضاف الى هذه **query** ففي معظم الأحيان كما قلنا لن نحتاج للمزيد ، وربما نتحدث عن باقي التعليمات في فصول أخرى .

لكي نتعامل مع قاعدة بيانات موجودة بعد إنشائها استخدم أمر **USE** كما يلي:

```
USE payroll
```

إنشاء الجداول Create Tables

ومن الأمور الهامة عند تصميم قواعد البيانات تحديد هيكل الجدول بحيث يحتوي علي مفتاح أساسي (**Primary Key**) ومفتاح خارجي (**Foreign Key**) ويجب أن يحقق المفتاح الأساسي الهدف من عدم تكرار بيانات السجل الواحد أكثر من مرة . أما حقل المفتاح الخارجي فيقوم بربط الجدول مع جدول آخر بناء علي حقل مفتاح أساسي أو

openkandil.com

خارجي . والمثال التالي يوضح ذلك :

مثلا عندما ننشي جدول يحتوي بيانات الفواتير ونسميه **Invoice** فانه سوف يحمل البيانات التالية :

اسم الحقل	طول الحقل	معناه
Invoice_No	8	رقم الفاتورة
Customer_Name	30	اسم العميل
Customer_Id	5	رقم العميل
Amount	6	قيمة الفاتورة
Customer_Address	40	عنوان العميل
City	15	المدينة
State	15	المحافظة

هذا الجدول سوف يعمل بدون أية مشاكل ، ولكن تخيل معي كل مرة سوف نقوم باصدار فاتورة جديدة لنفس العميل فاننا سوف نقوم بادخال بيانات العميل مرة أخرى من اسم ورقم وعنوان والمدينة والمحافظة ، تخيل اصدار ٣٠ فاتورة مثلا لعميل واحد و ١٠٠ فاتورة لعميل آخر وهكذا فكيف ستمتليء قاعدة البيانات بحجم وبيانات مكررة لا داعي لتكرارها أبدا ، يكفي أن يكون لدينا جدول باسماء وعناوين العملاء يكتب فيه بيانات العميل كالتالي :

اسم الحقل	معناه
Customer_Name	اسم العميل
Customer_Id	رقم العميل
Customer_Address	عنوان العميل
City	المدينة
State	المحافظة

وبالتالي يكون جدول الفواتير **Invoices** في الصورة التالية :

اسم الحقل	معناه
Invoice_No	رقم الفاتورة
Customer_Name	اسم العميل
Customer_Id	رقم العميل

Orelkandi.com

وبهذا فان بيانات العميل سوف نقوم بادخالها مرة واحدة وعند اصدار الفاتورة يتم سحب بيانات العميل من الجدول الخاص بالعملاء **Customers** أليس ذلك أفضل تنظيم ؟

ولكي نقوم ببناء الجداول في قاعدة البيانات فاننا نحتاج للالمام ببعض القواعد التي تتبع مع مكونات أي جدول وهي :

- اسم الجدول **Table Name** وشروط اختيار الاسم
- اسماء الحقول داخل الجدول **Fields Name** وشروطها
- أنواع البيانات لكل حقل **Field's Data Type**

اسم الجدول **Table Name**

عند إنشاء جدول يجب الالتزام بعدة قواعد لتسمية الجدول هي :

- يتكون اسم الجدول من ١ الى ٣٠ من الحروف الانجليزية كلها فيما عـــدا "/" أو "."
- أول حرف في اسم الجدول يجب أن يكون من **A** الى **Z** والباقي ممكن أن يشتمل على حروف أو أرقام أو بعض الحروف الخاصة مثل " _ "," @ "," # "
- يجب عدم تكرار اسم الجدول في قاعدة بيانات واحدة
- في بعض قواعد البيانات مثل **Oracle 9i** يمكن استخدام الحروف الكبيرة أو الصغيرة على السواء **Upper or Lower Case** في أسماء الجداول
- يمنع استخدام الكلمات المحجوزة في أسماء الجداول مثل **SELECT**
- في قواعد البيانات **MySQL** يمكن أن يصل الاسم الى ٦٤ حرف

أسماء الحقول **Field Names**

لا يمكن تكرار اسم الحقل داخل جدول واحد ، بالطبع يمكن تكرار اسم الحقـــل داخل قاعدة البيانات ولكن في جداول أخرى ، مثلا يمكن اســـتخدام **Customer_Id** كاسم لحقل ولكن في جدول **Invoices** مرة وفي جدول **Customers** مرة أخرى .

أنواع البيانات **Data Types**

المقصود هو نوع البيانات التي سيتم تخزينها في كل حقل من الحقول ، وهذا يعني أن

obeikandl.com

كل حقل نقوم بإنشائه داخل الجدول يجب أن يكون له نوع ومساحة فمثلا حقل سيستقبل اسم فيه يختلف عن حقل سيستقبل رقم وغير ذلك .

الجدول التالي ١-٩ يشتمل على معظم أنواع البيانات التي يمكن تخصيص الحقل بها وشرح لهذه الأنواع :

نوع البيانات	معناها أو استخدامها
VARCHAR(size)	هذا النوع يستخدم مع الحقول التي سوف نستخدمها في تخزين الحروف ، ومساحة البيانات التي نكتبها تحدد طول الحقل ، فمثلا لو كتبنا CHARS(10) معناها أننا حجزنا حقل طوله ١٠ حروف ، فلو كتبنا فيه ٤ حروف فسوف يتم اضافة ٦ مسافات
DATE	هذا الحقل يتم حجزه للتاريخ حسب طريقة قواعد البيانات في كتابة التاريخ فبعضها يتكون من القرن والسنة والشهر واليوم والساعة والدقيقة والثانية
LONG	هذا الحقل يتم حجزه لاستخدامه في الحروف والكلمات وقد يصل طوله الى ٢ جيجابايت ، ويستخدم في كتابة ملاحظات أو بيانات كبيرة وسوف نوضحها فيما بعد
LONG RAW	نوع هذا الحقل BINARY ويصل طوله الى ٢ جيجابايت
NUMBER	يستخدم هذا النوع في حجز حقل لاستخدامه في الأرقام سواء الموجبة أو السالبة.
Integer	نفس استخدام number ولا يسمح بالعلامة العشرية
Integer(n)	يتم تحديد عدد الأرقام التي يمكن تخزينها في الحقل

تستعمل جملة CREATE TABLE لإنشاء الجدول بالشكل العام الآتى

```
CREATE TABLE table_name
Fieldname 1 Fieldtype [Not Null]
Fieldname 2 FieldType [Not Null]
Fieldname 3 FieldType [Not Null]
```

وعن هذا الشكل نوضح الآتى

Table _name: اسم الجدول المطلوب.

Field _name: اسم الحقل.

Field _type: نوع بيانات الحقل.

Not Null: معامل اختيارى إذا أضيف فعناه لابد من إدخال بيانات فى هذا الحقل.

تعال معي ننشئ جدول الفواتير **invoices** وجدول العملاء **Customers** على النحو التالي :

أولا سننشئ قاعدة البيانات :

```
create database invoicing;
use invoicing;
```

ثم ننشيء الجدول الأول :

```
create table invoices (
invoice_no varchar(8),
customer_id  varchar(6),
amount numeric);
```

يشتمل هذا الجدول على ثلاثة حقول هى

Invoice_no "رقم الفاتورة": يجب أن يحتوى على بيانات حرفية بطول ٨.

Customer_ID "رقم العميل" : يجب أن يحتوى على بيانات حرفية بطول ٦.

Amount "المبلغ" : يجب أن يحتوى على بيانات رقمية.

ثم الجدول الثاني :

```
create table customers (
customer_id varchar(8),
address  varchar(40),
city  varchar(15),
state varchar(15));
```

***القيمة* NULL *و القيمة* NOT NULL**

هل هناك فرق ؟ طبعا يعرف كل منا وخاصة مطوري التطبيقات أن هناك فرق بين **NULL** و **NOT NULL** فهناك فرق بين خانة بها مسافة أو خانة بها صفر فالمسافة **Blank**

والصفر هما قيم هامة ومؤثرة وتختلف تماما عن خانة ليست بها أي قيمة فيجب الاحتياط عند تعريف الحقول والتفريق بين لا قيمة **NULL** وعدم ترك الحقل بدون قيمة **NOT NULL** وهذا يعتمد على طريقة تصميمك للجداول واستخدامك للحقول .

عند إنشاء الجداول تسمح معظم نظم قواعد البيانات بتعريف الحقل باستخدام **NOT NULL** . ومعناها أن العمود لا يجب أن يحتوي علي قيمة **NULL** أي لابد أن يحتوي علي قيمة في جميع سجلات الجدول والمثال التالي يوضح استخدام **NOT NULL** :

```
CREATE TABLE INVOICES(
 NAME CHAR (15) NOT NULL,
 AMOUNT int,
 ACCOUNT_ID NOT NULL);
```

تعديل مواصفات الجدول بعد إنشائه :

كثيرا ما تحتاج لتعديل محتويات الجداول بعد إنشائها . يستخدم أمر **ALTER TABLE** بواسطة مسئول قواعد البيانات أو المصمم لتغيير مواصفات الجدول بعد إنشائه .ويوفر هذا الأمر حذف الجدول بعد إنشائه عند الرغبة في تعديله وعموما يستخدم أمر **ALTER TABLE** في :

- إضافة عمود (حقل)لجدول موجود .
- *تعديل مواصفات جدول موجود .*
- *حذف حقل موجود .*

والشكل العام لأمر **ALTER TABLE** كما يلي :

```
ALTER TABLE table_name
ADD column_name data_type |
MODIFY column_name data_type
```

حيث **ADD** لإضافة عمود جديد متبوعة باسم العمود ونوع البيانات التي ستخزن به و **MODIFY** لتعديل مواصفات حقل موجود متبوعة باسم الحقل ونوع البيانات الجديدة .

مثال :

لتعديل طول **invoice_no** الموجود بجدول **INVOICES** الذي أنشأناه قبل قليل ليصبح

طوله ١٥ بدلا من ٨ استخدم الأمر التالي :

```
ALTER TABLE INVOICES
MODIFY INVOICE_NO VARCHAR(15);
```

لاحظ أنه لا يمكن تغيير طول الحقل إلى أقل من طول أطول قيمة أو معلومة موجودة في الجدول حتى لا تفقد البيانات الموجودة .

ولإضافة حقل التاريخ إلى الجدول السابق استخدم المثال التالي :

```
ALTER TABLE INVOICES
ADD INV_DATE date;
```

ولحذف حقل استخدم **Drop Field** كما يلي :

```
Alter table orders
Drop order_date;
```

حذف جدول

قد تحتاج لحذف جدول موجود في قاعدة البيانات . ومن الأمثلة التي تحتاج فيها لحذف الجدول عندما تنشئ جدولا بغرض الاختبار فعندما تنتهي من الجدول واختباره يمكنك حذف هذا الجدول .لاحظ أن حذف الجدول يحذف كل ما يتعلق به مثل الفهارس والبيانات فكن حذرا عند استخدام هذا الأمر .

يستخدم أمر **DROP TABLE** لحذف جدول موجود والشكل العام لهذا الأمر كما يلي :

```
DROP TABLE table_name
```

ولحذف جدول **orders** استخدم الأمر التالي :

```
Drop table orders;
```

حذف قاعدة البيانات

يمكن حذف قاعدة البيانات باستخدام أمر **DROP DATABASE** بنفس طريقة حذف جدول البيانات والشكل العام لهذا الأمر هو :

```
DROP DATABASE database_name
```

لحذف قاعدة البيانات **stockcontrol** استخدم الأمر التالي :

obeikandi.com

```
Drop database stockcontrol;
```

تكوين جدول جديد من جدول موجود

أحيانا يكون من المفيد تكوين جدول مشابه للجدول الموجود . في هذه الحالة استخدم أمر **CREATE TABLE** كما يلي :

```
CREATE TABLE new_table (field1,field2,field3)
AS ( SELECT field1,field2,field3
FROM old_table < WHERE >
```

والمثال التالي يوضح ذلك :

نحتاج أحيانا لانشاء جدول مماثل لجدول موجود من حيث الحقول والخـــصائص ، هب أن لديك جدول للاعبي كرة القدم وتريد انشاء جدول آخر للاعبين أكثر مـــن ٢١ سنة ، الواقع أنك تريد انشاء جدول متطابق مع الجدول الأساسي فيما عدا أن البيانات في الجدول الجديد لن تشتمل الا على بعض اللاعبين ممن هم تحت سن ٢١ سنة .

الجدول الأساسي بياناته كالتالي :

```
Select * from first_team;
```

F_Name	L_Name	Club	Age
Atef	Obaid	Tarsana	33
Wael	Riyad	Ahly	22
Ahmed	Hassan	Ahly	19
Nader	Sayed	Masry	35
Essam	Hadary	Ahly	30
Ibraheem	Hassan	Masry	34
Hazem	Emam	Zamalek	29
Khaled	Ghandoor	Zamalek	30
Mohamed	Sabry	Zamalek	18
Alaa	Hassan	Ismaely	17
Hany	Gamal	Mansoura	20
Ahmed	Yakoub	Ahly	27
Mohmmad	Hamed	Ahly	21
Farook	Hosny	Assuit	17

obeikandl.com

سنقوم الآن بإنشاء جدول للناشئين أقل من ٢١ سنة مـــن الجـــدول الرئيـــسي وذلـــك باستخدام MySQL Query التالية :

```
create table youth_team
as ( select * from first_team
where age < 21 );
```

ولإظهار محتويات الجدول الجديد

```
select * from youth_team;
```

F_Name	L_Name	Club	Age
Ahmed	Hassan	Ahly	19
Mohamed	Sabry	Zamalek	18
Alaa	Hassan	Ismaely	17
Hany	Gamal	Mansoura	20
Farook	Hosny	Assuit	17

openkandil.com