الفصل الحادى عشر
معالجة البيانات

سوف نتحدث في هذا الفصل عن معالجة البيانـــات **Data Manipulation** وفي نهاية هذا الفصل فانك سوف تكـــون قادرا على فهم

- أوامر معالجة البيانات.
- استخدامات الشرط **Where** عند معالجة البيانات.

أوامر معالجة البيانات

استخدمنا في الفصول السابقة جملة SELECT وعرفنا العديد من أشكال استخداماتها ولعلنا أيضا استخدمنا كل من INSERT و DELETE ولكن دون شرح يذكر ، ولعلنا نتساءل كيف أدخلنا البيانات داخل قاعدة البيانات أو الجداول ؟

لو رجعنا بالذاكرة الى بعض أنظمة قواعد البيانات البسيطة مثل DB –IV أو Fox Pro ثم بعد ذلك Microsoft Access فسوف نجد أنها مصممة بحيث تسمح لك بادخال البيانات من داخل أدوات تأتي ضمن قاعدة البيانات .

ولكن قواعد البيانات الكبيرة مثل Oracle, SQL Server, Sybase وحتى MySQL وغيرها تم تصميمها للتعامل مع برامج التطبيقات مثل Visual Basic و C# وغيرها ، ذلك لأن قواعد البيانات هذه يتم استخدامها على مدى أوسع ولن يكون من المناسب أن يتم ادخال آلاف المعلومات مباشرة من خلال قاعدة البيانات ، فقد يعمل العشرات من مدخلي البيانات في ادخال المعلومات في وقت واحد أو متزامن ، لذا فقد وجد من الأنسب أن يتم ذلك من خلال التطبيقات التي يمكن أن تعمل من خلال عدد من محطات العمل Workstations الموجودة في عدد من الأماكن وكلها مرتبطة بخادم الملفات الموجود عليه قاعدة البيانات .

هذا لا يمنع وجود امكانية محدودة لادخال عدد من السجلات داخل قاعدة البيانات بواسطة مدير قاعدة البيانات (Data Base Administrator (DBA .

كما يمكن استخدام جملة Insert لإدخال عدد من السجلات أو المعلومات داخل الجداول وهذا ما سوف نتعرف عليه .

يتم أيضا إدخال البيانات واستخراجها إلى ومن قاعدة البيانات باستخدام بعض الأدوات التي تأتي مع نظام قواعد البيانات مثل Import/Export أو Transform وتلك تتيح ادخال بيانات من ملفات مثل word, excel إلى داخل الجداول ، كما تتيح نسخ بيانات قاعدة بيانات الى قاعدة بيانات أخرى ، وهذا ما سوف نتعرف عليه لاحقا .

ولأن نظام قواعد البيانات MySQL مصمم أساسا للعمل من خلال شاشة مثل شاشات DOS أو UNIX تلك الشاشة السوداء التي تقوم من خلالها بادخال الأوامر والتعليمات بنفسك، ولان كل شاشات MySQL تعمل بنفس الطريقة ، فقد قام بعض مطوري البرامج بتطوير نظام يتيح التعامل مع قواعد البيانات MySQL بطريقة النوافذ Windows وطريقة الأدوات من هذا النظام يسمى Mysql front وسوف نتعرف عليه في آخر الكتاب، كما أن رخص هذا التطبيق مجانية مثل نظام MySQL تماما وهو يدخل ضمن دائرة التطبيقات الحرة مثل نظام التشغيل Linux و تطبيقات Open Office وغيرها .

إدخال بيانات باستخدام جملة Insert

هناك عدد من الاستخدامات المفيدة لهذا الأمر ، فتستطيع ادخال عدد من السجلات باستخدام هذا الأمر أو ادخال عدد من الحقول أو سجل واحد أو حقل واحد أو ادخال بيانات جدول موجود في جدول آخر وهذا ما سنتعلمه في هذا البند ، ولكن دعنا نتعرف على بعض القواعد قبل البدء في استخدام هذا الآمر :

- القيم المدخلة يجب أن تكون من نفس نوع الحقل ، حقل الأرقام Number, Numeric, Decimal مثلا لا يصح ادخال حروف فيه
- طول البيانات لا يجب أن يزيد عن طول الحقل ، فمثلا حقل طوله ٢٠ لا يجب ادخل بيانات طولها ٢٥
- ترتيب البيانات المدخلة يجب أن يكون نفس ترتيب الحقول
- إذا لم تضع اسماء الحقول فان ترتيب ادخال البيانات يتم وفقا لترتيب الحقول داخل الجدول ، القيمة الأولى تدخل الى الحقل الأول وهكذا
- يتم وضع القيم داخل "فاصلة" إذا كانت حروف وبدون فاصلة إذا كانت أرقام

الشكل العام لجملة Insert

كما سبق القول فان جملة Insert تستخدم لإدخال بيانات مباشرة أو من خلال جدول

آخر وفيما يلي الشكل الأول (إدخال بيانات مباشرة)

```
Insert INTO table_name
(Field1,field2 , .....)
VALUES (value1, value2, . .. . . )
```

الشكل الثاني

```
Insert INTO table_name
(Field1,field2 , .....)
SELECT Filed , field 2 , field 3
From other_table_name
```

سنستخدم هذا الشكل العام لإدخال بيانات في الأمثلة التالية :

إدخال بيانات سجل واحد الى جدول

في هذا المثال سوف نقوم بإنشاء جدول داخل قاعدة البيانات **football** ثم نقوم بإدخال بيانات سجل الى هذا الجداول.

لتنشيط قاعدة البيانات

```
use football;
```

لإنشاء الجدول

```
create table new_team
(name varchar(30) not null,
club varchar(10) not null,
age integer(2) not null,
rem varchar(20) null );
```

ولإضافة سجل إلي الجدول يشتمل علي أربعة حقول وإدخال بيانات الحقول الأربعة

```
INSERT INTO new_team
(name, club, Age, rem)
VALUES ('ahmed sayed', 'suez', 25, 'forward');
```

في هذا المثال تم اضافة بيانات سجل الى جدول **new_team**

لكي نستعرض بيانات هذا السجل ، سوف نقوم بتنفيذ **Query** التالية :

```
select * from new_team;
        name  club age      rem
ahmed sayed  suez  25  forward
```

لاحظ هنا أن ترتيب القيم المدخلة تم بنفس ترتيب الحقول فالقيمة "ahmed sayed" في أول حقل وهو name والقيمة "suez" في ثاني حقل وهو club ... وهكذا .

إدخال عدد من السجلات في جدول

في المثال السابق أدخلنا سجل واحد في جدول new_team وفي هذا المثال سوف نقوم بإضافة عدد من السجلات :

```
INSERT INTO new_team
VALUES ('aly Hassan', 'ahly', 22, null);
INSERT INTO new_team
VALUES ('wael riyad', 'ahly', 23, 'center');
INSERT INTO new_team
VALUES ('hussam Hassan', 'masry', 37, 'wing');
INSERT INTO new_team
VALUES ('hazem emam', 'zamalek', 29, 'hunter');
INSERT INTO new_team
VALUES ('sayed zaian', 'canal', 38, 'star');
INSERT INTO new_team
VALUES ('ahmed medo', 'hendi', 25, null);
```

في هذا المثال تم اضافة عدد ٦ سجلات لجدول new_team

لاحظ أننا استخدمنا NULL أي لا قيمة في السجل الأول داخل هذه المجموعة من السجلات .

يمكن إدخال هذا السجل بطريقة أخري هكذا :

```
INSERT INTO new_team
VALUES ('aly Hassan','ahly',22,' ');
```

وعند استعراض بيانات هذا السجل ستظهر فراغات في حقل rem .

نستعرض الآن ما حدث :

```
Select * from new_team;
```

name	club	age	rem
ahmed sayed	suez	25	forward
aly Hassan	ahly	22	NULL
aly Hassan	ahly	22	NULL
wael riyad	ahly	23	center

hussam Hassan	masry	37	wing
hazem emam	zamalek	29	hunter
sayed zaian	canal	38	star
ahmed medo	hendi	25	NULL

تعديل بيانات موجودة :

عندما تريد تعديل القيمة في أحد الحقول أو أحد الأعمدة داخل جدول فانك تريد استخدام أمر أو جملة أخرى غير جملة Insert ، لذلك فسوف نستخدم جملة update وفيما يلي سوف نعرض الشكل العام للجملة (Syntax) ثم بعد ذلك نأخذ أمثلة عن استخداماتها :

```
UPDATE table_name
SET (Field1=value, Field2=value, . . . )
WHERE condition;
```

في هذا المثال التالي سوف نستخدم update لتعديل بيانات حقل في كل الجدول وذلك باضافة ١ الى قيمة العمر ، ولكن دعنا نستعرض بيانات الجدول كما هي ثم ندخل الأمر update ثم نعرض الجدول بعد تنفيذ الأمر :

لاستعراض بيانات الجدول قبل التعديل أنظر المثال التالي

```
Select * from stars_team;
```

F_Name	L_Name	Club	Age
Atef	Obaid	Tarsana	33
Wael	Riyad	Ahly	22
Nader	Sayed	Masry	35
Essam	Hadary	Ahly	30
Ibraheem	Hassan	Masry	34
Hazem	Emam	Zamalek	29
Khaled	Ghandoor	Zamalek	30
Ahmed	Yakoub	Ahly	27
Mohmmad	Hamed	Ahly	21

ادخل الأمر التالي :

```
USE FOOTBALL;
UPDATE STARS_TEAM
SET AGE = AGE + 1;
```

ومعناه نشط قاعدة البيانات **football** وقم بتعديل جدول **stars_team** وذلك بإضافة ١ إلى حقل **AGE** في الجدول كله

عند عرض بيانات الجدول ستتم إضافة ١ إلى خانة العمر لكل سجلات الجدول . وذلـــك لأنك لم تشترط أن يتم التعديل على سجل معين .

```
Select * from starts_team;
```

F_Name	L_Name	Club	Age
Atef	Obaid	Tarsana	34
Wael	Riyad	Ahly	23
Nader	Sayed	Masry	36
Essam	Hadary	Ahly	31
Ibraheem	Hassan	Masry	35
Hazem	Emam	Zamalek	30
Khaled	Ghandoor	Zamalek	31
Ahmed	Yakoub	Ahly	28
Mohmmad	Hamed	Ahly	22

استخدام شرط مع update

في المثال التالي سوف (نبحبحها حبتين) ونعطي الموظفين علاوات ١٠% لمن تقل مرتباتهم عن ٤٠٠٠ جنيه و ٥% لباقي الموظفين ، كيف نفعل ذلك ؟

انظر بيانات الجدول قبل التعديل

```
Select * from salary;
```

emp_id	title	salary
1	manager	15000
2	assistantant	10000
3	tech support	5000
4	Instructor	3000
5	SW eng	3000
6	developer	2000

```
7       developer    2500
8       designer     3000
```

أول تعديل

```
update salary
set salary = salary * 1.10
where salary < 4000;
```

ثاني تعديل

```
update salary
set salary = salary * 1.05
where salary > 4000;
```

الطباعة بعد التعديلات

```
select * from salary;
emp_id title          salary
1       manager       15750
2       assistantant  10500
3       tech support  5250
4       Instructor    3300
5       SW eng        3300
6       developer     2200
7       developer     2750
8       designer      3300
```

فارن النتيجة قبل التعديل بتلك الموجودة بعد التعديل

حذف بيانات موجودة

عندما تقرر استخدام جملة **DELETE** يجب أن تتأكد من التعليمات التي تصدرها لنظام قواعد البيانات لحذف مجموعة من السجلات أو الصفوف من داخل جدول ، لأن نظام قواعد البيانات لن يصدر لك رسالة تحذير أو تأكيد حذف بل سيقوم بتنفيذ ما يطلب منه بعكس ما سوف نتعلمه في فصول تالية من كيفية التعامل في هذا الأمر من خلال تطبيقات أو أدوات أخرى .

وعلى ذلك أود التنويه على الأمور التالية ، لتضعها في الحسبان وأنت تستخدم جملة

DELETE:

- إن حذف سجل أو سجلات من جدول لا يعني مسح القيم الموجودة في السجل أو الصف والإبقاء على الصف فارغا، بل ان أمر **Delete** يحذف الصف بما فيه من بيانات ولا يقوم بتفريغ محتواه مثلما يحدث مع **Excel** مثلا باستخدام الآمر **clear content** .
- يجب أن يتم الحذف في وقت تتأكد من أنه لا يوجد مستخدمين آخرين يقومون بالعمل مع نظام قواعد البيانات ، فقد تقوم بحذف سجل وهناك مستخدمين آخرين يستخدمون نفس السجل في نفس الوقت ، فقد يؤدي ذلك لبعض المشاكل .
- لا يقوم أمر **Delete** بحذف حقل من الحقول ، يمكنك استخدام **UPDATE** للتعامل مع حقل أو حقول
- لا يقوم **DELETE** بحذف جدول أو قاعدة بيانات ، يمكنك استخدام **DROP TABLE** أو **DROP DATABASE** التي تعلمناها في فصل سابق
- عندما تقوم بحذف صف أو سجل من قاعدة البيانات يجب التأكد من أن هذا السجل غير مشار اليه في جدول آخر باستخدام **FOREIGN KEY** مثلا ، فسوف يؤدي ذلك الى عدة مشاكل .

الشكل العام لجملة **DELETE**

```
DELETE FROM TABLENAME
WHERE CONDITION;
```

لحذف السجل أو الصف رقم ١٣ من جدول **Sectors** استخدم الأمر هكذا

```
Delete from sectors
Where emp_id = 13;
```

في هذا المثال سنقوم بحذف الأفراد الذين تقل مرتباتهم عن ٢٠٠٠ جنيه

```
Delete from sectors
Where salary < 2000;
```

عندما تقوم بكتابة الأمر delete from table_name; دون استخدام أي شرط فسوف يقوم نظام قواعد البيانات بحذف كل الصفوف أو السجلات الموجودة في الجدول.

! انتبه !

عندما تحذف صف أو عدد من الصفوف يجب مراعاة ارتباط هذه السجلات بجداول أخرى من خلال Foreign Key ولاحظ علاقة الوالد والابن Parent/Child كما ذكرنا سابقا.