



جامعة المنارة
كلية الهندسة
قسم الهندسة المعلوماتية

مقرر قواعد بيانات 1 لغة التعامل مع البيانات DML

د.م. كندة أبو قاسم
م.جمال محمود – م.جهاد عيسى

جلسة الأسبوع الخامس
(الفصل الثاني 2023 | 2024)

الغاية من الجلسة

- ✓ التعريف بلغة التعامل مع البيانات DML.
- ✓ إجراء عمليات إدخال بيانات باستخدام الأمر Insert.
- ✓ إجراء عمليات تعديل بيانات باستخدام الأمر Update.
- ✓ إجراء عمليات حذف بيانات باستخدام الأمر Delete.

لغة التعامل مع البيانات DML

DML أو Data Manipulation Language هي مجموعة جزئية من أوامر لغة SQL تستخدم للتعامل مع البيانات المخزنة في جداول قاعدة البيانات ومعالجتها، حيث تتضمن هذه المجموعة أوامر لإضافة وتعديل وحذف البيانات المخزنة في قاعدة البيانات.

تتضمن هذه المجموعة، التعليمات التالية:

INSERT: وهي مخصصة لإضافة سجلات جديدة إلى قاعدة البيانات.

UPDATE: وهي مخصصة لتعديل سجل أو مجموعة من السجلات في قاعدة البيانات.

DELETE: وهي مخصصة لحذف سجل أو مجموعة سجلات من قاعدة البيانات.

تعليمية إدخال البيانات INSERT

تملك هذه التعليمية الصيغة العامة التالية:

```
INSERT INTO table [(column [, column...])]  
VALUES (value [, value...]);
```

حيث :

- Table هو اسم الجدول المراد إدخال البيانات فيه.

- Column اسم العمود.
- Value القيمة المراد إدخالها.

إرشادات لاستخدام التعليمة:

1. من الممكن حذف الجزء المتعلق بذكر أسماء الأعمدة بحيث تصبح الصيغة:

```
INSERT INTO table  
VALUES (value [, value...]);
```

- في هذه الحالة يجب إدخال قيم في جميع أعمدة الجدول وتكرر القيم بعد الكلمة المفتاحية VALUES بنفس ترتيب وجودها في الجدول.
2. من الممكن إدخال قيم لبعض أعمدة الجدول فقط، في هذه الحالة تذكر الأعمدة المراد إدخال قيم لها فقط وتكرر القيم بعد الكلمة المفتاحية VALUES بنفس ترتيب ورودها بعد اسم الجدول، مع مراعاة الحقول المطلوبة (التي تملك قيد NOT NULL في بنية الجدول).
3. توضع قيم الحقول النصية وحقول التاريخ (ضمن قائمة القيمة VALUES) بين علامات اقتباس "أما القيم الرقمية فتكتب مباشرة بدون علامات اقتباس.
4. من الممكن إدخال أكثر من سجل بنفس التعليمة، في هذه الحالة توضع قيم كل سجل ضمن قوسين، ويفصل بين السجلات بفاصلة.

ملاحظة هامة جداً:

- من الضروري لسلامة تمرير القيم معرفة ببنية الجدول بحيث يعرف ترتيب الحقول، نوع بيانات الحقول، القيود المفروضة على كل حقل.
- يمكن معرفة بنية الجدول بعدة طرق:
- من مستعرض الكائنات object explorer في برنامج الإدارة SSMS، انقر على تبويب columns.
 - explorer في برنامج الإدارة SSMS، باختيار اسم الجدول والنقر بالزر اليميني واختيار design.
 - باستخدام أمر النظام:

```
EXEC sp_help table_name;
```



إدخال بيانات إلى جميع أعمدة الجدول:

تمرين 1 : أكتب استعلاماً لإدخال السجل التالي إلى الجدول Shippers في قاعدة البيانات Northwind

ShipperID	CompanyName	Phone
101	Xpress	(011)333-1874

لنجرّب كتابة التعليميّة كما يلي:

```
INSERT INTO Shippers  
VALUES (101, 'Xpress', '(011)333-1874');
```

ملاحظ أنّ هذه التعليميّة لن تعمل لأنّ الحقل ShipperID يملك صفة auto-increment (تزايد تلقائي) وبالتالي، فإنّ قيمه تتولد بشكل تلقائي من قبل محرك قواعد البيانات، لذلك لانقوم بإدخال قيم له يدوياً، لتصحيح الخطأ نكتب التعليميّة على النحو:

```
INSERT INTO Shippers  
VALUES ('Xpress', '(011)333-1874');
```

ملاحظة:

كان بالإمكان كتابة هذه التعليميّة على النحو:

```
INSERT INTO Shippers (CompanyName, Phone)  
VALUES ('Xpress', '(011)333-1874');
```

كما كان بالإمكان كتابة هذه التعليميّة على النحو:

```
INSERT INTO Shippers (Phone ,CompanyName)  
VALUES ('(011)333-1874', 'Xpress');
```



أخطاء شائعة:

- إذا كتبنا التعليمة على النحو التالي،

```
INSERT INTO Shippers (CompanyName, Phone)
```

```
VALUES ( Xpress, (011)333-1874);
```

فإنها لن تعمل، لأن الحقلين المذكورين من النوع النصي وبالتالي يجب أن توضع قيمهما ضمن علامات اقتباس.

- إذا كتبنا التعليمة على النحو التالي،

```
INSERT INTO Shippers (CompanyName, Phone)
```

```
VALUES ('Xpress', 0113331874);
```

فإنها لن تعمل، لأن الحقل Phone هو من النوع النصي وبالتالي لا يمكن تمرير قيمة رقمية له.

إدخال بيانات إلى بعض أعمدة الجدول:

أشرنا سابقاً إلى أنه، من الممكن إدخال قيم لبعض أعمدة الجدول فقط، في هذه الحالة تذكر الأعمدة المراد إدخال قيم لها فقط وتكرر القيم بعد الكلمة المفتاحية VALUES بنفس ترتيب ورودها بعد إسم الجدول، مع مراعاة الحقول المطلوبة (التي تملك قيد NOT NULL في بنية الجدول).

تمرين 2 : أكتب استعلاماً لإدخال السجل التالي إلى الجدول Shippers في قاعدة البيانات Northwind

CompanyName
ManaraCo

```
INSERT INTO Shippers (CompanyName)
```

```
VALUES ( ' ManaraCo ');
```

سيعمل هذا الأمر بشكل سليم، وسيتم إدخال القيمة NULL للحقل Phone.



تمرين 3 : أكتب استعلاماً لإدخال السجل التالي إلى الجدول Shippers في قاعدة البيانات Northwind

Phone
(011)777-88888

```
INSERT INTO Shippers (Phone)
VALUES ( ' (011)777-88888 ');
```

لن يعمل هذا الأمر بشكل سليم، لأن الحقل CompanyName يملك الصفة غير صفري أو مطلوب NOT NULL وبالتالي لا بد من إدخال قيمة له.

إدخال البيانات مع مراعاة القيم الصفرية:

يمكن للحقول التي لا تملك القيد NOT NULL أن يتم إدخال قيمة لها أو أن لا يتم إدخال قيمة لها.

في حال رغبتنا بإدخال قيم إلى بعض حقول الجدول، هنا يمكن أن لا يتم ذكر الحقول الصفرية ضمن قائمة الحقول إذا لم نكن نرغب بإعطائها قيمة (وفي هذه الحالة يسجل في الحقل الصفرية القيمة NULL تلقائياً).

أما إذا أردنا استخدام الصيغة التي تدخل البيانات إلى جميع حقول الجدول، ولم نكن نرغب بإدخال قيمة إلى حقل ما، عندها يجب أن يتم ذكر ذلك صراحة.

تمرين 4 : أعد كتابة الاستعلام المذكور في التمرين رقم 2 باستخدام الصيغة التي يتم استخدامها لإدخال قيم لجميع أعمدة الجدول.

```
INSERT INTO Shippers
VALUES ( ' ManaraCo ',NULL);
```



ويمكن استخدام الصيغة التالية:

```
INSERT INTO Shippers (CompanyName, Phone)
VALUES ('ManaraCo',NULL);
```

إدخال بعض القيم الخاصة:

يمكن استخدام التابع CURRENT_TIMESTAMP في حال رغبتنا بأن يتم إدخال التاريخ والوقت الحالي إلى حقل من النوع datetime. (يمكن استخدام توابع أخرى نتعرف عليها لاحقاً).

تمرين 5 : أكتب استعلاماً لإضافة سجل من اختيارك إلى الجدول sales في قاعدة البيانات Pubs بحيث تكون قيمة الحقل ord_date هي التاريخ الحالي.

```
INSERT INTO sales
VALUES ('7131','3000',CURRENT_TIMESTAMP,10,'NET30','BU1032');
```

إدخال أكثر من سجل في أمر واحد:

يمكن إدخال أكثر من سجل إلى جدول ما باستخدام أمر إدخال واحد وفي هذه الحالة يتم وضع السجلات بعد الكلمة المفتاحية VALUES بحيث يكون كل سجل محصوراً ضمن قوسين ويفصل بين السجلات بفواصل.

تمرين 6 : نفذ الاستعلام التالي وعين النتيجة.

```
INSERT INTO shippers
VALUES ('DHL4', '(011)987-8764'), ('DHL5', '(012)344-1722'), ('DHL6', '(033)444-2564')
```



إدخال بيانات إلى جدول بطريقة النسخ من جدول آخر:

تمرين 7 : نفذ الاستعلام التالي وعين النتيجة.

```
INSERT INTO Shippers
SELECT Companyname, Phone
FROM Customers
WHERE Country = 'USA'
```

تعليمة تعديل البيانات UPDATE

تملك هذه التعليمة الصيغة العامة التالية:

```
UPDATE table
SET column = value [, column = value, ...]
[WHERE condition];
```

حيث :

- Table هو اسم الجدول المراد إدخال البيانات فيه.
- Column اسم العمود.
- Value القيمة الجديدة.

ملاحظة هامة جداً:

يتم تحديد السجل (أو السجلات) المراد تعديله من خلال الشرط WHERE ويجب الانتباه بشكل دقيق أثناء وضع الشرط لأن أي خلل قد يؤدي إلى إجراء تعديلات غير مرغوبة، فمثلاً في حال وضع شرط محقق في أكثر من سجل من الجدول سيتم تعديل جميع هذه السجلات، وفي حال وضع شرط محقق دائماً (أو في حال غياب الشرط) فإنه سيتم تعديل جميع سجلات الجدول.



تمرين 8 : أكتب استعلاماً لتعديل السجل الذي قمت بإضافته مؤخراً إلى الجدول Shippers في قاعدة البيانات Northwind بحيث يتم تعديل اسم الشركة DHL4 إلى Delivery Speed.

UPDATE Shippers

SET companyname = 'Delivery Speed', Phone = '(000)333-2222'

WHERE companyName = 'DHL5'

تمرين 9 : نرغب بتعديل أسعار الكتب في الجدول titles في قاعدة البيانات Pubs بحيث يتم رفع أسعار جميع الكتب بنسبة 20%، أكتب استعلاماً يحقق ذلك.

UPDATE titles

SET price = price + (0.20 * price)

تعليمة حذف البيانات DELETE

تملك هذه التعليمة الصيغة العامة التالية:

DELETE FROM table

[**WHERE** condition];

ملاحظة هامة جداً:

يتم تحديد السجل (أو السجلات) المراد حذفه من خلال الشرط WHERE ويجب الانتباه بشكل دقيق أثناء وضع الشرط لأن أي خلل قد يؤدي إلى إجراء عمليات حذف غير مرغوبة، فمثلاً في حال وضع شرط محقق في أكثر من سجل من الجدول سيتم حذف جميع هذه السجلات، وفي حال وضع شرط محقق دائماً (أو في حال غياب الشرط) فإنه سيتم حذف جميع سجلات الجدول.

تمرين 10 : أكتب استعلاماً لحذف السجل الذي قمت بتعديله مؤخراً إلى الجدول Shippers في قاعدة البيانات Northwind للشركة Delivery Speed.

```
DELETE FROM Shippers
WHERE companyname = ' Delivery Speed '
```

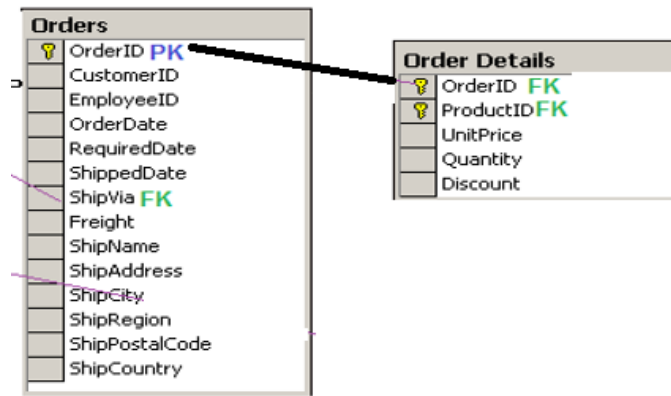
أوامر DML ومفهوم تكامل البيانات Data integrity

لعل أحد أهم الأمور التي يجب الانتباه إليها عند تنفيذ أوامر وعمليات معالجة بيانات DML هو مراعاة قواعد تكامل البيانات data integrity .

يقصد بمفهوم تكامل البيانات: دقة البيانات واكتمالها وجودتها.

تضمن محركات قواعد البيانات تحقيق هذا الأمر عبر إتاحة فرض قيود تكامل integrity constraints على الجداول من خلال الروابط أو العلاقات relations.

لتوضيح الأمر بشكل عملي، لنعد إلى تصميم قاعدة البيانات NorthWind، نجد أن بنية هذه القاعدة تتضمن (على سبيل المثال) جدولين هما: جدول الطلبات Orders وجدول تفاصيل الطلبات Order Details وهذان الجدول مرتبطان ببعضهما بعلاقة واحد لعدة one to many عبر الحقل OrderID بحيث يكون لكل طلب في جدول الطلبات العديد من السجلات المرتبطة في جدول تفاصيل الطلبات.





لنحاول تنفيذ الاستعلامات التالية ومعاينة النتائج:

```
DELETE FROM Orders  
WHERE OrderID = 10248;
```

```
UPDATE Orders  
SET OrderID=8888888  
WHERE OrderID = 10248;
```

```
INSERT INTO [Order Details](OrderID,ProductID,UnitPrice,Quantity,Discount)  
VALUES (767676, 11,100.2,8,0)
```

ستفشل جميع هذه الاستعلامات في التنفيذ بسبب انتهاك قواعد تكامل البيانات.

بالخلاصة:

- لا يمكن إضافة سجلات في جدول ابن ما لم يكن السجل الأب موجوداً.
- لا يمكن تعديل قيمة الحقل المفتاحي لسجل في جدول أب إذا كان يمتلك أبناء مرتبطين به في الجدول الابن.
- لا يمكن حذف سجل في جدول أب إذا كان يمتلك أبناء مرتبطين به في الجدول الابن.

نهاية الجلسة