



جامعة طرابلس  
University of Tripoli



# MySQL, The Comprehensive Course الدورة الشاملة

إعداد وتقديم: د. عبدالناصر ضياف

# مؤثرات SQL وأوامر DML

## Operators in SQL & DML Commands

- نتعرف هنا على المؤثرات البرمجية بأنواعها في MySQL ونعرِّج على أساسيات أوامر إدارة البيانات DML،  
INSERT, UPDATE, and DELETE •

المؤثرات الحسابية Arithmetic Operators ▶

المؤثرات العلائقية أو المقارنة Relational or Comparison Operators ▶

المؤثرات المنطقية Logical Operators ▶

مؤثرات الخانة الثنائية Bitwise Operators ▶

الأولويات بين المؤثرات Precedence of Operators ▶

إدخال البيانات Data Insertion ▶

تعديل البيانات Data Updating ▶

حذف البيانات Data Deletion ▶

## مؤثرات المقارنة Comparison Operators

المؤثر	الوصف
=	يساوي
<=>	يساوي آمن من NULL
>	أكبر من
<	أصغر من
>=	أكبر من أو يساوي
<=	أصغر من أو يساوي
<> !=	لا يساوي

## المؤثرات الحسابية Arithmetic Operators

المؤثر	الوصف
+	جمع
-	طرح
*	ضرب
/	قسمة
<b>DIV</b>	قسمة صحيحة
% <b>MOD</b>	باقي القسمة

## Sign Operators مؤثرات الإشارة

المؤثر	الوصف
+	إشارة موجبة
-	إشارة سالبة

## مؤثرات الخانة الثنائية

المؤثر	الوصف
&	AND
	OR
^	XOR
<<	إزاحة يسرى
>>	إزاحة يُمنى
~	متمم الأحاد

## المؤثرات المنطقية Logical Operators

المؤثر	الوصف
ALL	إرجاع TRUE عندما يتم استيفاء كافة الاستعلامات الفرعية
AND &&	إرجاع TRUE إذا تم استيفاء جميع الشروط في AND
ANY SOME	إرجاع TRUE إذا تم استيفاء أي استعلام فرعي
BETWEEN	إرجاع TRUE إذا كان المعامل ضمن النطاق المحدد
EXISTS	يُرجع TRUE في حالة وجود سجل بيانات واحد أو أكثر
IN	إرجاع TRUE إذا كان المعامل يطابق إحدى عناصر القائمة
IS NULL	تستخدم لمقارنة القيم NULL
LIKE	تُرجع TRUE إذا كان المعامل مشابهًا للنمط
NOT !	إرجاع TRUE إذا لم يتم استيفاء الشرط
OR	إرجاع TRUE إذا تم استيفاء أحد الشروط في OR
SOME	إرجاع TRUE إذا تم استيفاء أي من الاستعلامات الفرعية لدينا
XOR	ترجع TRUE إذا كانت نتيجة شروط طرفي XOR مختلفة

# Precedence of Operator أولويات المؤثرات

▶ دخول NULL في أي تعبير يجعل النتيجة NULL ما لم يكن في محل DO-NOT-CARE

▶ أولويات المؤثرات:

```
!  
- (unary minus), ~ (unary bit inversion)  
^  
*, /, DIV, %, MOD  
-, +  
<<, >>  
&  
|  
= (comparison), <=>, >=, >, <=, <, <>, !=, IS, LIKE, IN, BETWEEN  
NOT  
AND, &&  
XOR  
OR, ||  
= (assignment), :=
```

# AND, OR, IN, NOT IN, BETWEEN, NOT BETWEEN

- ▶ استجلاب قائمة الزبائن المقيمين في ولاية كاليفورنيا الأميركية
- ▶ استجلاب قائمة الزبائن المقيمين في الولايات الأميركية أو فرنسا
- ▶ استجلاب قائمة الزبائن المقيمين في أميركا أو فرنسا شرط أن تكون حدود ائتمانهم 100,000 دولار فما فوق
- ▶ `value IN (value1, value2, value3,...)` is equivalent to:
- ▶ `value = value1 OR value = value2 OR value = value3 OR ...`

▶ تُرجع IN اللاشيء NULL في إحدى الحالتين:

- القيمة التي تبحث عنها NULL
- لم تجد القيمة المطابقة في القائمة مع احتواء القائمة على NULL
- ▶ استجلاب قائمة المكاتب الموجودة بأميركا أو فرنسا
- ▶ استجلاب قائمة المكاتب عدا الموجودة بأميركا أو فرنسا
- ▶ قائمة المنتجات التي أسعار شرائها ما بين 90 و100 دولار
- ▶ استجلاب قائمة الطلبات باستثناء ما تم في سنة 2005

# AND, OR, IN, NOT IN, BETWEEN, NOT BETWEEN

---

```
SELECT * FROM customers WHERE country='USA' AND state='CA' ORDER BY  
contactFirstName;
```

```
SELECT * FROM customers WHERE country='USA' OR country='FRANCE' ORDER BY country,  
contactFirstName;
```

```
SELECT * FROM customers WHERE (country='USA' OR country='FRANCE') AND  
creditLimit>=100000;
```

```
SELECT * FROM offices WHERE country IN ('USA','FRANCE');
```

```
SELECT * FROM offices WHERE country NOT IN ('USA','FRANCE');
```

```
SELECT * FROM products WHERE buyPrice BETWEEN 20 AND 90 ORDER BY buyPrice;
```

```
SELECT * FROM orders WHERE orderDate NOT BETWEEN '2004-12-01' AND  
'2004-12-31' ORDER BY orderDate;
```



# LIKE, Escape code, IS NULL, LIMIT

## ▶ في LIKE

- العلامة المئوية % يطابق أي نص من طول صفر أو أكثر من الأحرف

- علامة الشرطة السفلية \_ تطابق حرفاً واحداً

▶ استجلاب قائمة الزبائن الذين تبدأ أسماء مندوبيهم بحرف a

▶ استجلاب قائمة الزبائن الذين تنتهي ألقابهم مندوبيهم بالنص on

▶ استجلاب قائمة الزبائن الذين تبدأ أسماءهم بحرف T شرط أن يكون الحرف الثالث m

▶ استجلاب المنتجات التي يحوي كودها الترميز 20\_

▶ استجلاب قائمة الزبائن الذين لم يخصص لهم ممثلي مبيعات

▶ LIMIT [offset:0, ] row\_count

▶ استجلاب قائمة بأكبر 5 زبائن لديهم أعلى حدود ائتمان

▶ استجلاب الزبون السادس ترتيباً من حيث كبر حدّه الائتماني

# LIKE, Escape code, IS NULL, LIMIT

---

```
SELECT * FROM customers WHERE contactFirstName LIKE 'a%';
```

```
SELECT * FROM customers WHERE contactLastName LIKE '%on';
```

```
SELECT * FROM customers WHERE customerName LIKE 'T_m%';
```

```
SELECT * FROM products WHERE productCode LIKE '%\_20%';
```

```
SELECT * FROM customers WHERE salesRepEmployeeNumber IS NULL;
```

```
SELECT * FROM customers ORDER BY creditLimit DESC LIMIT 5;
```

```
SELECT * FROM customers ORDER BY creditLimit DESC LIMIT 5, 1;
```

# أوامر معالجة البيانات

## Data Manipulation Language (DML) Commands

أوامر معالجة البيانات Data Manipulation Commands ▶

▶ **CRUD: Create, Read, Update, or Delete** data

◦ INSERT إدخال بيانات جديدة

◦ SELECT استرجاع أو استخلاص المعلومات

◦ UPDATE تعديل وتحديث البيانات

◦ DELETE حذف السجلات أو الصفوف

▶ اعتماد التغييرات في قاعدة البيانات أو التراجع عنها COMMIT or ROLLBACK

# إضافة صفوف للجدول

## إدخال صفوف بكامل السمات

- توفر لنا SQL إمكانية إدخال صفوف جديدة إلى الجدول من خلال تعليمة INSERT:

```
INSERT INTO table_name  
VALUES (value1, value2, .., valuen);
```

## تذكر:

- يتم إعطاء محتويات الصف بين أقواس بعد عبارة VALUES
- يتم إعطاء قيم الحقول أو السمات حسب ترتيبها بالجدول
- يتم الفصل بين القيم المعطاه باستخدام الفاصلة (,)
- القيم النصية والتاريخ والوقت يجب إدخالها بين علامات التنصيص (")
- القيم العددية يتم إدخالها بشكل مجرد
- يجب إدخال قيمًا لجميع حقول أو سمات الجدول

# إضافة صفوف للجدول

## ▶ إدخال صفوف تحوي قيمًا منعدمة NULLs

- يتم ذلك من خلال وضع NULL في محل قيمة الحقل أو الخاصية في تعليمة الإدخال INSERT:

```
INSERT INTO table_name  
VALUES (value1, NULL, ..., valuen);
```

- يُشترط للحقل الذي سيتم إعطاؤه NULL أن تكون له خاصية قبولها عند بناء الجدول

## ▶ إدخال صفوف باستخدام حقول أو سمات اختيارية

- تذكر أنه عند استخدامنا لتعليمة INSERT البسيطة كان لزامًا علينا إدخال قيمًا لجميع الحقول وحسب ترتيبها بتركيبة الجدول
- SQL تتيح لنا اختيار الحقول وترتيبها عند الإدخال
- فبذلك يمكننا تجاوز الحقول الاختيارية أو التي لا يوجد لدينا قيمًا لها NULLs والاقصصار فقط على القيم الإجبارية

```
INSERT INTO table_name (column_name1, ..., column_namen)  
VALUES (value1, ..., valuen);
```

# إضافة صفوف للجدول

▶ إدخال مجموعة صفوف في وقت واحد بالاستعانة بالاستفسار الفرعي:

▶ **تذكر:**

- في الصور السابقة لتعليمة INSERT يتم إدخال الصفوف واحدًا في كل خطوة one at a time
- تمكنا SQL من إدخال مجموعة من الصفوف في خطوة واحدة باستخدام جدول آخر كمصدر للبيانات من خلال استفسار فرعي sub-query:

```
INSERT INTO dst_table_name [(column_name1, .., column_namen)]
```

```
SELECT column_name1, .., column_namen FROM src_table_name;
```

# تعديل أو تحديث صفوف الجدول

من خلال تعليمة UPDATE، تمكننا SQL من تعديل بيانات الجدول بعد تحديد الصفوف المعنية

```
UPDATE table_name
```

```
SET column_name1=expression1 [, ..., column_namen=expressionn]
```

```
[WHERE condition];
```

**انتبه:**

- تعليمة UPDATE من النوع الموجه نحو فئة من الصفوف set-oriented-command فإن لم يتم تحديد هذه الفئة بشكل دقيق من خلال عبارة WHERE فسيتم تنفيذها على كل صفوف الجدول

# حذف صفوف من الجدول

▶ من خلال تعليمة DELETE، تمكننا SQL من حذف صف أو مجموعة صفوف من الجدول بعد تحديدها

```
DELETE FROM table_name  
  
[WHERE condition];
```

▶ **انتبه:**

◦ تعليمة DELETE من النوع الموجه نحو فئة من الصفوف set-oriented-command فإن لم يتم تحديد هذه الفئة بشكل دقيق من خلال عبارة WHERE فسيتم حذف جميع الصفوف من الجدول

▶ **تذكر:**

◦ تعليمة DELETE تحذف صفوفاً وليس حقولاً



# اعتماد التغييرات في قاعدة البيانات أو التراجع عنها

- ▶ تتيح معظم أنظمة قواعد البيانات خط دفاع ضد الأخطاء العفوية في إدارة البيانات بها من خلال تأجيل حفظ واعتماد عمليات الإدخال والتعديل والحذف التي يقوم بها المستخدم بقاعدة البيانات إلى حين طلب اعتمادها بشكل مستقل بعد التأكد من سلامتها أو التراجع عنها
- ▶ وحتى وقت طلب الاعتماد أو التراجع
  - يظهر للمشغل المعني وكأن عملياته تتم أولاً بأول داخل قاعدة البيانات
  - تكون العمليات التي يقوم بها المستخدم عرضة للضياع والتلاشي في حالة توقف عمل النظام لأي سبب، كانقطاع الكهرباء مثلاً
- ▶ تقوم تعليمة COMMIT باعتماد حفظ العمليات بقاعدة البيانات

```
COMMIT [WORK] ;
```

- ▶ تقوم التعليمة ROLLBACK بإلغاء تأثير أي عملية على البيانات منذ آخر اعتماد COMMIT واسترجاع المحتويات السابقة:

```
ROLLBACK ;
```

# اعتماد التغييرات في قاعدة البيانات أو التراجع عنها

انتبه: ▶

◦ تعتمد بعض نظم قواعد البيانات على إعدادات قد تجعل اعتماد وحفظ التغييرات يتم تلقائياً بعد كل عملية حيث لا يكون هناك مجال للتراجع

□ نظام إدارة قواعد البيانات MySQL هو أحد هذه الأنظمة حيث يملك إعداداً يسمى

AUTOCOMMIT وهو مبدئياً فعال TRUE

□ يمكنك تعطيله من خلال الأمر:

□ `SET AUTOCOMMIT=0;`