

الكلمات المفتاحية:

تعليلة ، صيغة، جدول ، حق ، سجل ، عمود، استعلام، استعلام فرعي.

ملخص:

تم تصنيف تعليلات لغة SQL إلى ثلاثة أصناف رئيسية ضمن ثلاث لغات فرعية: لغة معالجة المعطيات، ولغة تعريف المعطيات، ولغة التحكم بالمعطيات. و من بين هذه التعليلات تعتبر تعليلات معالجة المعطيات الأكثر استخداماً. سنستعرض تعليلات المعالجة والتي تشمل SELECT,INSERT,DELETE,UPDATE و بعض الكلمات المفتاحية المستخدمة معها.

أهداف تعليمية:

يتعرف الطالب في هذا الفصل على:

- استخدام تعليلة SELECT لاستعادة البيانات من قاعدة البيانات.
- استخدام تعليلة INSERT لإدراج سجلات ضمن جدول في قاعدة البيانات.
- استخدام تعليلة DELETE لحذف سجل أو مجموعة من السجلات.
- استخدام تعليلة UPDATE لتعديل سجل أو مجموعة من السجلات.
- التعرف على الإطار التطبيقي لاستخدام هذه التعليلات مع الكلمات المفتاحية المرافقة لها.

١. تعليلة SELECT

تعتبر تعليلة SELECT من أشهر تعليلات اللغة وأكثرها استخداماً. تُستخدم هذه التعليلة لاستعادة و انتقاء مجموعة من البيانات من قاعدة البيانات و ذلك بإعادة جدول يحتوي مجموعة البيانات المطلوبة.

تأخذ تعليلة SELECT الصيغة:

```
SELECT [field_1,field_2,..]  
FROM [table_name];
```

- تُستخدم إشارة * كبديل لأسماء الحقول (عادة لا ننصح باستخدامها في الحالات التطبيقية لأنها تُحمّل برنامج إدارة قاعدة البيانات عبء تحديد الحقول وتحديد عددها وأسماءها).
- يُستخدم تعبير DISTINCT لاستعادة جميع السجلات مع إلغاء التكرار في السجلات المعادة.
- يُستخدم التعبير ORDER BY لترتيب السجلات المعادة ترتيباً تصاعدياً أو تنازلياً حسب التعبير المرافق المستخدم: ASC للترتيب التصاعدي أو DESC للترتيب التنازلي.
- في حال الرغبة باستخدام أسماء بديلة لحقول جدول القيم المعادة نستخدم التعبير AS.

تمرين ١:

ما هي عبارة SQL التي تساعدنا على استعادة بيانات حقلي StudentName و StudentAge من جميع سجلات الجدول Students ؟

```
SELECT StudentName , StudentAge  
FROM Students ;
```

ما هي عبارة SQL التي تساعدنا على استعادة جميع السجلات من الجدول ؟

```
SELECT *  
FROM Students;
```

تمرين ٣:

ما هي عبارة SQL التي تساعدنا على استعادة جميع بيانات الحقل StudentName مع إلغاء التكرار في السجلات المُعادة ؟

```
SELECT DISTINCT StudentName  
FROM Students;
```

تمرين ٤:

ما هي عبارة SQL التي تساعدنا على استعادة جميع بيانات الحقول StudentName و StudentAge مرتبة ترتيباً تصاعدياً وفق الحقل StudentName؟

```
SELECT StudentName, StudentAge  
FROM Students  
ORDER BY StudentName ASC;
```

نستخدم الكلمة المفتاحية WHERE مع تعليمة SELECT لاستعادة مجموعة من السجلات التي تحقق شرط أو مجموعة من الشروط التي نعبر عنها بعبارة شرطية.

- تُعيد العبارة الشرطية قيمة منطقية (صح أو خطأ)
- يمكن للعبارة الشرطية أن تتضمن عمليات مقارنة مثل (=, <, >, <=, >=, <>) ويتم ضم السجل الذي يحققها إلى جدول النتائج.

الكلمات المفتاحية LIKE و BETWEEN

- تُستخدم الكلمة المفتاحية LIKE ضمن العبارة الشرطية، كشرط لوجود مثيل. غالباً ما تُستخدم هذه الكلمة مع إشارة (%)، التي تضاف إلى القيمة التي نبحث عن مثيلاتها، كبديل عن أي عدد من المحارف (أحرف أو أرقام أو إشارات). وتستخدم مع الإشارة (_) كبديل عن محرف واحد فقط.
- تُستخدم الكلمة المفتاحية BETWEEN ضمن العبارة الشرطية، كشرط لوجود قيمة محصورة بين قيمتين محددتين.
- تقبل الكلمة المفتاحية WHERE أكثر من شرط يفصل بينها عمليات منطقية مثل AND أو OR ويمكن أن يسبق الشرط العملية NOT لنفيه.

تمرين ٥:

نريد الحصول على قائمة بجميع السجلات التي تحتوي على السلسلة 'SVU' في حقل اسم الجامعة (universityName) ضمن جدول الجامعات (Universities).

```
SELECT *  
FROM Universities  
WHERE UniversityName LIKE '%SVU%';
```

تمرين ٦:

نريد الحصول على قائمة بجميع الطلاب ذوي الإسم Sami في حقل الاسم الأول للطلاب (StudentFirstName) ضمن جدول طلاب الجامعة الافتراضية (SVUStudents) والذين تنحصر أعمارهم (StudentAge) بين ٢٠ و ٣٠ عام.

```
SELECT *  
FROM SVUStudents  
WHERE StudentFirstName = 'Sami' AND  
StudentAge BETWEEN 20 AND 30;
```

٢. تعليمة DELETE

تقوم تعليمة DELETE بحذف سجل أو مجموعة من السجلات من جدول ما.

تأخذ تعليمة DELETE الصيغة :

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

تمرين:

نريد حذف السجل الخاص باسم المستخدم 'Sami' (الحقل StudentName) من جدول طلاب الجامعة الافتراضية SVUStudents:

```
DELETE FROM SVUStudents  
WHERE StudentName='Sami' ;
```

٣. تعليمة INSERT

تُستخدم تعليمة INSERT لإدراج سجل في جدول محدد.

تأخذ تعليمة INSERT الصيغة:

```
INSERT INTO table_name  
VALUES (value_1,value_2,value_3,...,value_n);
```

حيث تمثل value_1,...,value_n القيم التي سوف يتم تخزينها في السجل تبعاً لترتيب حقوله. كما يمكننا استخدام صيغة بديلة تمكننا من تحديد حقول السجل التي نود إدراج البيانات فيها فقط:

```
INSERT INTO table_name (field_1,field_2,...,field_n)
VALUES (value_1,value_2,...,value_n);
```

يمكن لتعليمة INSERT إدراج أكثر من سجل بأمر واحد ولكن ستحتاج إلى استخدام ما ندعوه الاستعلامات الفرعية (Sub Queries) التي سنأتي على ذكرها لاحقاً.

تمرين:

استخدم التعليمة اللازمة لإدراج سجل الطالب سامي في الجدول SVUStudents علماً أن البيانات التي تملكها عنه تقتصر على اسمه الأول (StudentFirstName) وهو Sami وعلى عمره (StudentAge) وهو ٣٠.

```
INSERT INTO SVUStudents (StudentFirstName, StudentAge)
VALUES ('Sami', 30);
```

٤. تعليمة UPDATE

تُستخدم تعليمة UPDATE لتعديل البيانات في سجل أو في مجموعة من السجلات.

ويمكن استخدام الكلمة المفتاحية WHERE مع تعليمة UPDATE لتحديد شروط التعديل. تأخذ تعليمة UPDATE الصيغة:

```
UPDATE table_name
SET Field1 = new_field_value1,
    Field2 = new_field_value2;
```

يمكن استخدام الكلمة المفتاحية WHERE مع تعليمة UPDATE لتصبح الصيغة:

```
UPDATE table_name
SET Field1 = new_field_value1,
    Field2 = new_field_value2
WHERE condition;
```

تمرين:

اكتب تعليمة تعديل سجل الطلاب الذين تزيد معدلاتهم (Average) عن ٧٠ بحيث يوضعون جميعاً في الصف (Class) رقم ٢ ضمن جدول طلاب الجامعة الافتراضية (SVUStudents).

```
UPDATE SVUStudents  
SET Class=2  
WHERE Average > 70;
```

٥. بعض الملاحظات العملية

تتطلب بعض برامج إدارة قواعد البيانات إنهاء كل تعليمة SQL بـ (;) بينما يضيفها البعض الآخر تلقائياً. ومن المهم استخدام تعليقات SQL وخصوصاً عند كتابة نصوص SQL تحتوي على عدد كبير من الأسطر والتعليمات.

كذلك، من المهم تلافي استخدام أسماء أعمدة (حقول) حاوية على فراغات. أما في الحالات الاضطرارية، فيمكن استخدام إشارات تنصيب أو أقواس مربعة لإحاطة اسم الحقل (إشارة تنصيب في Oracle و أقواس مربعة في Access و SQL Server []).

تمرين:

اكتب عبارة SQL التي تمكننا من استعادة جميع بيانات الحقول StudentName و StudentAge (من الجدول Students) الذين يدرسون في الصف (Class) رقم ٣ والذين تبدأ أسمائهم بالحرف 'M' أو الحرف 'S' وأن لا تقل أعمارهم عن ٢١ سنة. بحيث تكون النتيجة مرتبة تصاعدياً بالنسبة للاسم StudentName وتنازلياً بالنسبة للعمر. ماذا لو أردنا إلغاء التكرار بالبيانات (في حال وجوده)؟

```
SELECT StudentName, StudentAge  
FROM Students  
WHERE Class = 3 AND  
      (StudentName Like 'M%' OR StudentName Like 'S%') AND  
      StudentAge >= 21  
ORDER BY StudentName, StudentAge DESC;
```