



جامعة المنارة

كلية: الهندسة

قسم: المعلوماتية

اسم المقرر: قواعد بيانات ٢

رقم الجلسة (٣)

عنوان الجلسة

لغة التحكم بالتدفق والمؤشرات



العام الدراسي

٢٠٢٣_٢٠٢٤

الفصل الدراسي

الأول

جدول المحتويات

Contents

رقم الصفحة	العنوان
٣	الكلمات المفتاحية
٤	الحلقة الشرطية IF...ELSE
٥	التعليمة CASE
٧	حلقة التكرار WHILE

الغاية من الجلسة:

تعريف الطلاب على التحكم بتدفق و حركة البيانات باستخدام لغة T-SQL و ذلك بوضع شروط أكثر تعقيداً على عمليات مثل الاستعلام و ادخال البيانات أو حذفها و ذلك تمهيداً لبناء الاجراءات و التوابع و القوادح

لغة التحكم بالتدفق Control of Flow Language

الكلمات المفتاحية

الشرح	الكلمة المفتاحية
تحصر بينها مجموعة من التعليمات	BEGIN ... END
لتعريف التنفيذ الشرطي (أي تنفيذ كتلة تعليمات عند تحقق شرط منطقي ما) والتنفيذ الاختياري لكتلة أخرى عند عدم تحقق الشرط	IF ... ELSE
تنفيذ مجموعة من التعليمات طالما أن شرطاً محدداً ما زال محققاً	WHILE
متابعة التنفيذ عند التعليمة التالية للعلامة التي اسمها <i>label</i>	GOTO <i>label</i>
إعادة تنفيذ حلقة التكرار While	CONTINUE
الخروج غير المشروط	RETURN
الخروج من حلقة التكرار While الداخلية	BREAK
تعيد تعبيراً واحداً من مجموعة من التعابير بناء على شرط مقارنة	CASE

مقدمة:

في الكثير من الأحيان نجد أنه من الضروري ألا يتم تنفيذ استعلام ما ضمن إجرائية مخزنة إلا عند تحقق شرط معين كإدخال معامل بقيمة صحيحة مثلاً. وقد يكون أحد السيناريوهات المحتملة أيضاً الرغبة في عدم تنفيذ استعلام ما لم يعد استعلام سابق نتيجة متوقعة. أو أن نحتاج إلى تنفيذ استعلامات مختلفة بحسب قيمة معامل محدد مثلاً .

تحتوي أغلب أنظمة قواعد البيانات العلائقية على نوعين أساسيين من التعبيرات الشرطية:

• التعبير IF.....ELSE

• التعبير CASE.....WHEN

كما نحتاج في الكثير من الأحيان إلى تكرار مجموعة من التعليمات بحسب شرط أو شروط معينة. ويتم ذلك باستخدام التعبير WHILE.

سنقوم فيما يلي باستعراض عمل هذه البنى واختلافات تطبيقها بين قواعد البيانات المختلفة.

الحلقة الشرطية IF...ELSE

وتستخدم لتنفيذ كتلة تعليمات بناءً على تحقق شرط منطقي ما.

```
IF Boolean_expression
  { sql_statement | statement_block }
[ ELSE
  { sql_statement | statement_block } ]
```

ينفذ الجزء ELSE في حال كان الشرط المنطقي Boolean_expression غير محقق (أي يعطي قيمة منطقية FALSE). وهذا الجزء اختياري وليس بالضرورة موجود.

مثال ١:

يقوم المثال التالي بتحديد كون وسطي أسعار الكتب أكبر أو أصغر أو يساوي \$15 مع طباعة العبارة المناسبة

```
IF (SELECT AVG(price) FROM titles ) < $15
  PRINT 'Average title price is less than $15.'
ELSE
  IF (SELECT AVG(price) FROM titles ) > $15
    PRINT 'Average title price is more than $15.'
  ELSE
    PRINT 'Average title price is equal to $15.'
```

مثال ٢:

نريد طباعة بيانات عن الكتب بحسب مجالات السعر: أصغر من ١٠، بين ١٠ و ٢٠، وأكبر من \$20 مع رسالة توضيحية تبين مجال السعر.

```
Use Pubs
Go
Declare @msg VarChar(50)
If (Select Count(*) From Titles Where Price<10) >0
Begin
  Set @msg = 'There are several books with prices less than $10. These
Books are:'
  Print @msg
  Select title From titles where price < 10
End
If (Select Count(*) From Titles Where Price Between 10 and 20) >0
Begin
```

```

Set @msg = 'There are several books with bearable prices. These Books
are:'
Print @msg
Select title From titles Where price between 10 and 20
End
If (Select Count(*) From Titles Where Price > 20) >0
Begin
Set @msg = 'Many books are very expensive. These Books are:'
Print @msg
Select title From titles Where price > 20
End
    
```

التعليمة CASE

تعيد إحدى القيم الممكنة من مجموعة قيم وذلك بناء على شرط ما.

لهذه التعليمة نمطين:

- النمط البسيط Simple CASE: وفيه تتم مقارنة تعبير ما مع مجموعة من التعابير المحتملة لتحديد النتيجة.
- نمط الاكتشاف Searched CASE: وفيه يتم اختبار مجموعة من الشروط المنطقية لتحديد النتيجة.

يمكن تضمين جزء ELSE في كلا النمطين.

النمط البسيط:

```

CASE input_expression
  WHEN when_expression THEN result_expression
  [ ...n ]
  [
    ELSE else_result_expression
  ]
END
    
```

نمط الاكتشاف:

```

CASE
  WHEN Boolean_expression THEN result_expression
  [ ...n ]
  [
    ELSE else_result_expression
  ]
END
    
```

: input_expression

وهو التعبير المراد بمقارنته مع مجموعة القيم الموجودة في النمط البسيط (تعبير المقارنة)

: WHEN when_expression

وهو تعبير بسيط لمقارنة قيمته مع قيمة تعبير المقارنة. نمط هذا التعبير ونمط تعبير المقارنة يجب أن يتطابقا أو أن يكون هناك تحويل ممكن مضمن بينهما.

: THEN result_expression

وهو ناتج التعليم في حال طابق تعبير المقارنة التعبير البسيط المحدد في فقرة WHEN الموافقة.

n : للدلالة على أن الجزء WHEN...THEN يمكن أن تتكرر أكثر من مرة.

: ELSE else_result_expression

وهو نتيجة التعليم في حال كانت قيمة المقارنة مختلفة عن جميع التعابير البسيطة المحددة في الفقرات WHEN السابقة له. إذا لم يتم تحديد هذا الجزء وكانت قيمة المقارنة مغايرة لجميع الأنماط البسيطة المحددة فإن ناتج التعليم هو NULL.

مثال للنمط البسيط

يظهر المثال التالي كيفية استخدام تعليمة case من أجل تعديل القيمة الراجعة من حقل. تظهر التعليمة صنف، عنوان، سعر الكتاب، مع وسطي السعر للكتاب حسب النوع ومجموع سعر كل نوع من أنواع الكتب.

```
USE pubs
GO
SELECT
    CASE type
        WHEN 'popular_comp' THEN 'Popular Computing'
        WHEN 'mod_cook' THEN 'Modern Cooking'
        WHEN 'business' THEN 'Business'
        WHEN 'psychology' THEN 'Psychology'
        WHEN 'trad_cook' THEN 'Traditional Cooking'
        ELSE 'Not yet categorized'
    END AS category,type,
    price AS Price
FROM titles
WHERE price IS NOT NULL
ORDER BY type
```

مثال لنمط البحث

اكتب تعليمة استعلام عنوان الكتاب، تعيد من أجل كل كتاب "very Reasonable" إذا كان سعر الكتاب أقل من عشرة، "Coffee Table Title" إذا كان سعر الكتاب بين 10 و 20 (أصغر تماماً من 20) ، "not yet priced" إذا كان السعر null. و إلا تعيد "Expensive Book".

```
USE pubs
GO
SELECT
  CASE
    WHEN price IS NULL THEN 'Not yet priced'
    WHEN price < 10 THEN 'Very Reasonable Title'
    WHEN price >= 10 and price < 20 THEN 'Coffee Table Title'
    ELSE 'Expensive book!'
  END As cat_price ,
price
FROM titles
ORDER BY price
```

حلقة التكرار WHILE

وتستخدم لتنفيذ تعليمة SQL أو كتلة تعليمات طالما أن شرطاً منطقياً مازال محققاً.
ففي SQL Server يستخدم التعبير WHILE مع BEGIN وEND.

```
WHILE Boolean_expression
  { sql_statement | statement_block }
  [ BREAK ]
  { sql_statement | statement_block }
  [ CONTINUE ]
{statements}
```

:Boolean_expression

وهو تعبير منطقي (أي بعيد إما صح أو خطأ). يمكن للتعبير المنطقي أن يحوي تعليمة Select وفي هذه الحالة يجب وضع تعليمة الانتقاء بين أقواس.

```
{ sql_statement | statement_block }
```

ويمكن أن تتضمن أي تعليمة SQL أو كتلة تعليمات SQL. كتلة تعليمات هي مجموعة من التعليمات الموضوعية بين BEGIN وEND.

:BREAK

وتؤدي إلى الخروج من تعليمة WHILE الداخلية (في حالة وجود أكثر من حلقة WHILE منسوجة ضمن بعضها البعض). الكلمة المفتاحية END تحدد نهاية الحلقة.

:CONTINUE

وتؤدي إلى العودة إلى بداية الحلقة مجدداً وإهمال كافة التعليمات اللاحقة لها.

مثال لحلقة while

نريد أن نضاعف أسعار جميع الكتب طالما أن وسطي السعر أقل من \$٣٠ وأعلى سعر لا يتجاوز \$١٠٠.

```
USE pubs
GO
Begin tran t ;
WHILE (SELECT AVG(price) FROM titles) < $30
BEGIN
    IF (SELECT MAX(price) FROM titles) > $40
        BREAK

    UPDATE titles
    SET price = price * 2

    Print ' Hel'
END
Rollback tran t ;
```