

المحاضرة السابعة

الاستعلامات الفرعية

Sub Queries

أساسيات الاستعلامات الفرعية Subquery Fundamentals
الاستعلام الفرعي هو تعليمة انتقاء SELECT تعيد قيمة وحيدة بحيث تكون مضمنة ضمن تعليمة SELECT, INSERT, UPDATE, or DELETE أو ضمن استعلام فرعي آخر. يمكن استخدام الاستعلامات الفرعية في أي مكان يمكن وضع تعبير فيه.

بمعنى آخر:

الاستعلام الفرعي هو أي استعلام يتم تضمينه في استعلام آخر ويجب على أسئلة مركبة.

يسمى الاستعلام الفرعي بالاستعلام الداخلي Inner Query، بينما يسمى الاستعلام الأب بالاستعلام الخارجي Outer Query. هناك نوعان أساسيان من الاستعلامات الفرعية:

1. الاستعلامات الفرعية المرتبطة باستعلام رئيسي:

هي الاستعلامات الفرعية التي تعتمد في عملها على بيانات من استعلامات رئيسية حاوية لها. في هذا النوع من الاستعلامات الفرعية، يتم تكرار عملية تنفيذ الاستعلام بعدد مرات مساوٍ لعدد السجلات التي يُعيدها الاستعلام الرئيسي. تحتاج الاستعلامات الفرعية المرتبطة باستعلام رئيسي للكثير من الوقت والمعالجة بالمقارنة مع الاستعلامات غير المرتبطة باستعلامها الرئيسي.

مثال:

نفرض أن لدينا الجدول Accounts الحاوي على الأرقام التسلسلية للحسابات المصرفية accountID وقيم أرصدة هذه الحسابات accountBalance. ولنفرض أن لدينا الجدول Clients الحاوي على أسماء أصحاب الحسابات clientName، ورقم الحساب لكل زبون accountID.

لإظهار قائمة بأرقام الحسابات وأرصدها وأسماء أصحابها نستخدم الصيغة:

```
SELECT ac.accountID, ac.accountBalance as dd ,
(SELECT clientName
FROM Clients cl
WHERE cl.accountID = ac.accountID) AS
myClientName
FROM Accounts ac ;
```

Client as cl	
clientName	accountID
Ahmad	121
Yaser	123
Fahed	124
Wesam	122

Accounts as ac	
accountID	accountBalance
121	2755
122	4588
123	5690
124	7800

accountID	dd	myclientName
121	2755	Ahmad
122	4588	Wesam
123	5690	Yaser
124	7800	Fahed

يُعتبر المثال السابق عيّنة من الاستعلامات الفرعية المرتبطة باستعلام رئيسي نظراً لكون الاستعلام الفرعي (الظاهر باللون الأحمر) لن يعمل وحيداً، بل يحتاج إلى معلومات من الاستعلام الرئيسي وهي قيم Accounts.accountID.

2. الاستعلامات الفرعية غير المرتبطة (المستقلة):

وهي الاستعلامات الفرعية التي تكون مستقلة تماماً عن الاستعلامات الرئيسية الحاوية لها، أي أن الاستعلام الفرعي سيُنَفَّذ بشكل كامل ويُمرر القيمة أو مجموعة القيم الناتجة إلى الاستعلام الرئيسي.



أنواع الاستعلامات الفرعية

1. الاستعلامات الفرعية التي يجب أن تعيد قيمة واحدة:

(1) استعمال الاستعلام الفرعي كعمود من أعمدة الاستعلام الرئيسي

يُعتبر استعمال الاستعلام الفرعي كعمود من أعمدة الاستعلام الرئيسي، أحد التركيبات المُستخدمة بكثرة في الاستعلامات الفرعية.

كما في المثال السابق.

يأخذ هذا الاستعلام الصيغة:

```
SELECT columnA, columnD as mdv, (subquery) AS columnB  
FROM Table_Name;
```

في الصيغة السابقة سوف يتم تنفيذ الاستعلام Subquery على كل سجل (سطر) يعيده الاستعلام الرئيسي. يفيد هذا النوع من الاستعلامات الفرعية في توليد علاقة

بين جدول وآخر.

وفقاً لهذه الصيغة يجب ألا يعيد الاستعلام الفرعي أكثر من قيمة واحدة (خلية وحيدة) لكل سجل في سجلات الاستعلام الرئيسي.

2 استعمال الاستعلامات الفرعية ضمن شرط WHERE

تدخل نتيجة الاستعلام الفرعي ضمن الشرط في تعبير WHERE شرط أن يعيد الاستعلام الفرعي **قيمة وحيدة**. (خلية وحيدة) (إذا كان شرط المقارنة هو يساوي) يأخذ هذا الاستعلام الصيغة:

```
SELECT columnA, columnB
FROM Table_Name
WHERE columnB = (Subquery);
```

مثال:

نريد قائمة بأسماء المؤلفين الذين يسكنون في نفس المدينة التي تقع فيها شركة النشر Algodata Infosystems. **علما أن هناك مدينة واحدة لكل ناشر**. (العبارة باللون الأحمر جعلتنا نضمن أن الاستعلام الفرعي سيعيد قيمة وحيدة)

```
SELECT au_fname, au_lname
FROM authors
WHERE city =
```

```
( SELECT city FROM
publishers WHERE
pub_name = 'Algodata Infosystems' )
```

في هذه الصيغة أيضاً يجب أن يعيد Subquery قيمة وحيدة. نلاحظ أن نتيجة الاستعلام الفرعي دخلت كجزء من الشرط في تعبير WHERE.

2. الاستعلامات الفرعية التي يمكن أن تعيد عدة قيم:

1 الاستعلامات الفرعية مع التعبير IN, NOT IN

اشتراطنا حتى الآن في الاستعلامات الفرعية، سواء تلك المستخدمة كأعمدة في الاستعلام الرئيسي أو تلك المستخدمة ضمن شرط التعبير WHERE، أن يعيد الاستعلام الفرعي قيمة وحيدة كي لا يفشل الاستعلام.

لكي نتمكن من استخدام الاستعلامات الفرعية التي تعيد أكثر من قيمة نستخدم التعبير IN وذلك وفق الصيغة التالية:

```
SELECT columnA, columnB
FROM Table_Name
WHERE columnC IN (Subquery);
```

لكن يجب أن نراعي أن يعيد الاستعلام الفرعي المستخدم مع التعبير IN **حقلًا** واحداً فقط (نقصد هنا حقلًا واحدًا وليس قيمة واحدة). (بينما من الممكن له أن لا يعيد أي سطر أو يعيد أي عدد من الأسطر من ضمن ذلك الحقل الواحد) أي كما يلي:

```
SELECT column1 FROM Table1;
```

مثال:

نريد قائمة بأسماء الناشرين الذين ينشرون كتباً في مجال إدارة الأعمال:
ومرة أخرى أسماء الناشرين الذين لا ينشرون كتباً في مجال إدارة الأعمال:

```
SELECT PUB_NAME
FROM Publishers
WHERE PUB_ID IN (SELECT PUB_ID FROM
TITLES
WHERE TYPE = 'BUSINESS')
```

```
SELECT PUB_NAME
FROM Publishers
WHERE PUB_ID NOT IN (SELECT PUB_ID FROM
TITLES
WHERE TYPE = 'BUSINESS')
```

أمثلة أخرى إضافية
مثال:

نريد تقريراً بأسماء المؤلفين الذين يسكنون ولاية كاليفورنيا ويحصلون على أقل من 30% من المبيعات كحق ملكية: حق الملكية هو الحق royaltyper من الجدول titleauthor

```
SELECT
    au_fname, au_lname
FROM
    Authors WHERE state = 'CA'
    AND
    au_id IN (
        SELECT au_id
        FROM titleauthor
        WHERE royaltyper < 30)
```

مثال آخر

```
USE PUBLS
GO
SELECT au_fname, au_lname
FROM Authors
WHERE au_id NOT IN
    (SELECT au_id FROM titleauthor
    WHERE royaltyper < 30 )
```

قواعد في الاستعلامات الفرعية Subquery Rules

يتضمن الاستعلام الفرعي الفقرات التالية:

1. فقرة الانتقاء النظامية Select
2. فقرة جداول الانتقاء From
3. فقرة الفلتر Where وهي فقرة اختيارية
4. فقرة تجميع Group by وهي فقرة اختيارية
5. فقرة التجميع و الفلتر Having وهي فقرة اختيارية

• يجب أن يحاط الاستعلام الفرعي دوماً بأقواس ().

- يمكن للاستعلام الفرعي أن يظهر في مكان أي تعبير شريطة أن يعيد قيمة واحدة.
 - يمكن وضع الاستعلام الفرعي في فقرات WHERE, HAVING من تعليمات SELECT, UPDATE, DELETE. كما يمكن تضمين استعلام فرعي ضمن استعلام فرعي آخر حتى 32 مستوى.
- غالباً ما يتم استخدام الاستعلامات الفرعية في فقرة الفلترة WHERE للعمليات SELECT, UPDATE, DELETE.

➤ لا يمكن استخدام أعمدة الجداول الفرعية في عملية الاختيار الخارجية إذا كانت الجداول المستخدمة في الاستعلام الفرعي غير مستخدمة في الاستعلام الخارجي.

- يمكن استخدام فقرات Group By و Having في الاستعلامات الفرعية المضمنة في فقرة مقارنة متبوعة حكماً بإحدى الكلمات ALL أو ANY (أو أن تكون واثقاً أن الاستعلام الفرعي سيعيد قيمة واحدة)
- لا يمكن استخدام كلمة DISTINCT في الفقرات الفرعية المتضمنة على Group By
- يمكن استخدام فقرة Order By مع كلمة TOP فقط في الاستعلامات الفرعية

استعمال الأسماء المستعارة مع الاستعلام الفرعي Subqueries with aliases
تتبع الأعمدة المذكورة في فقرة الاختيار والفلترة إلى الجداول المذكورة في فقرة FROM ما لم يذكر خلاف ذلك صراحة.

إن استعمال الأسماء المستعارة في الاستعلامات الفرعية مفيد لتوضيح تابعة الأعمدة المختارة في كل من الاستعلامين الخارجي والداخلي، وتظهر أهمية الأسماء المستعارة في حال استخدم الاستعلامين الخارجي والداخلي على بعض الجداول المشتركة مع انتقاء أعمدة بنفس الاسم

مثال:

المطلوب تقرير بأرقام الطلبات وتواريخها مع أعلى سعر لمفردات كل طلبية.

SELECT

ord.OrderId,

ord.OrderDate,

(SELECT

Max(od.UnitPrice)

FROM [order details] od

WHERE od.OrderId = ord.OrderId

) AS MaxItemPrice

FROM

orders ord

[order details] **od**

UnitPrice	oderID
105	1
53.5	1
22	1
16	1
14	2
12.5	2
76	2
38	3
19	3

orders **ord**

orderID	OrderDate
1	1-5-2014
2	2-4-2013
3	6-3-2011

MaxItemPrice	orderId	orderDate
105	1	1-5-2014
76	2	2-4-2013
38	3	6-3-2011

مثال:

نريد تقريراً باسم ونسبة ومدينة كل مؤلف اسمه الأول Livia ونسبته Karsen

SELECT a1.au_fname, a1.au_lname, city

FROM Authors a1

WHERE a1.au_id IN

(SELECT a2.au_id FROM Authors a2

WHERE a2.au_fname = 'Livia' AND

a2.au_lname = 'Karsen')

الاستعلامات الفرعية في تعليمات الإضافة، الحذف والتعديل

يمكن استخدام الاستعلامات الفرعية مع جميع تعليمات DML الأربعة

مثال:

نريد مضاعفة أسعار جميع الكتب للناشرين في مجال إدارة الأعمال (يعني نشروا كتابا على الأقل في مجال الإدارة):

```
UPDATE
  titles
SET price = price * 2
WHERE
  pub_id IN
    (SELECT      pub_id FROM titles
     WHERE  type='Business' )
```

ملاحظة : ما هو الفرق بين التعليمة السابقة وهذه التعليمة:

```
UPDATE titles
SET price = price * 2
WHERE type='Business'
```

الاستعلامات الفرعية مع عمليات المقارنة

يمكن استخدام عمليات المقارنة التالية (>, >=, <, <=, !=, <!, >!, <=, >=) في الاستعلامات الفرعية.

عمليات المقارنة غير المتغيرة (أي التي لا تحوي على ANY, ALL) المستخدمة في الاستعلامات الفرعية يجب أن تعيد قيمة واحدة.

نسمى عملية مقارنة ما بأنها متغيرة إذا سبقت بإحدى الكلمات ANY, SOME, ALL.

كما يمكن استخدام التوابع التجميعية في الاستعلامات الفرعية المستخدمة مع عمليات المقارنة غير المتغيرة، ذلك أن التوابع التجميعية تعيد قيمة وحيدة.

مثال:

نريد قائمة بأسماء الكتب التي سعرها أدنى ما يمكن. (مع تابع تجميعي، وهو يشبه مثال الطلبات السابق)

```
SELECT title FROM
  Titles WHERE
```

```
price = ( SELECT MIN(price)
          FROM titles )
```

بما أن الاستعلامات الفرعية المستخدمة في عمليات المقارنة غير المتغيرة يجب أن تعيد قيمة وحيدة فلا يمكن استخدام فقرات Group By و Having فيها ما لم نكن واثقين أن هذه الاستعلامات سوف تعيد قيمة وحيدة.

مثال:

أوجد قائمة بنمط واسم جميع الكتب التي يساوي سعرها سعر أرخص كتاب من كتب إدارة الأعمال

```
SELECT type, title
FROM TITLES
WHERE price = ( SELECT MIN(price)
                FROM TITLES
                GROUP BY type
```

```
Having
type = 'business' )
```

بطريقة أخرى:

```
SELECT type, title
FROM TITLES
WHERE price = ( SELECT MIN(price)
                FROM TITLES
                where
                type = 'business' )
```

عمليات المقارنة مع ANY, SOME, ALL

التعبير ALL:

يُستخدم التعبير ALL للتحقق من كون جميع القيم المعادة من استعلام فرعي، تحقق شرطاً ما في تعبير WHERE التابع للاستعلام الرئيسي.

التعبير ANY:

يُستخدم التعبير ANY للتحقق من كون قيمة أو أكثر من القيم المعادة من استعلام فرعي، تحقق شرطاً واحداً على الأقل من شروط تعبير WHERE الخاص بالاستعلام الرئيسي.

يمكن للاستعلامات الفرعية المستخدمة في مقارنة متغيرة أن تعيد أكثر من سطر، كما يمكنها أن تستخدم Group By و Having.

لنأخذ على سبيل المثال المقارنة المتغيرة أكبر من الكل ALL >: لكي يتم اختيار سطر من الاستعلام الخارجي فيجب أن تكون قيمة حقل المقارنة فيه أكبر من جميع القيم العائدة من الاستعلام الداخلي.

أما المقارنة المتغيرة أكبر من أحد ANY > فإنه لكي يتم اختيار سطر من الاستعلام الخارجي فيجب أن يكون حقل المقارنة أكبر من إحدى قيم الاستعلام الداخلي على الأقل (SOME مكافئة لـ ANY التي تكافئ بدورها IN في حالة المساواة)

مثال: AA

نريد قائمة بالمؤلفين الذين يسكنون في مدن يوجد فيها ناشرين
(أي مدينة تعتبر مقبولة طالما أنها هي إحدى مدن الناشرين)

```
USE PUBLS
GO
SELECT au_fname, au_lname
FROM authors
WHERE city = ANY ( SELECT city
FROM Publishers )
```

نلاحظ أن ANY = مكافئ تماماً لـ IN غير أن هذا الكلام غير صحيح من أجل ANY <> و NOT IN. حيث أن ANY <> تعني أن عنصر المقارنة مختلف عن أحد عناصر الاستعلام الداخلي على الأقل، في حين أن NOT IN تعني أن عنصر المقارنة مختلف عن جميع عناصر الاستعلام الداخلي.

الاستعلامات الفرعية مع EXISTS

التعبير EXISTS:

يُستخدم التعبير EXISTS للتحقق من إعادة الاستعلام الفرعي الذي يليه لأي سجل.

ويأخذ التعبير كاملاً القيمة TRUE في حال أرجع الاستعلام الفرعي سجلاً أو أكثر، والقيمة FALSE إذا لم يُرجع الاستعلام الفرعي أي سجل.

مثال:

أوجد قائمة بالناشرين الذين ينشرون كتباً في مجال إدارة الأعمال.

```
SELECT pub_name
FROM publishers p
WHERE EXISTS (SELECT *
               FROM titles t
               WHERE t.pub_id = p.pub_id
               AND
               t.type = 'Business' )
```

إن كلمة EXISTS مهمة جداً حيث تكون أحياناً هي الطريقة الوحيدة للحصول على المعلومات المطلوبة. جميع الاستعلامات التي تستخدم IN, SOME, ANY يمكن إعادة صياغتها باستخدام EXISTS.

مثال:

أوجد قائمة باسم وكنية كل مؤلف يسكن في مدينة يوجد فيها دار نشر.

```
USE pubs

SELECT au_lname,      au_fname
FROM   authors
WHERE  city = ANY ( SELECT city FROM publishers )
```

-- يمكن كتابتها باستخدام exists:

```
USE pubs
SELECT au_lname,      au_fname
FROM   authors
WHERE  exists (SELECT *
               FROM publishers
               WHERE publishers.city = authors.city
               )
```

الاستعلامات الفرعية مع NOT EXISTS

وهي مماثلة لكلمة EXISTS غير أنها تعطي نتيجة معاكسة لـ EXISTS

إيجاد التقاطعات والفروق باستخدام EXISTS, NOT EXISTS

دون استخدام التعليمات **Intersect** و **except** التي تعلمناها في الفصل الثاني

أحد أهم استخدامات EXISTS و NOT EXISTS هو تنفيذ عمليات الجبر العلاقتي.

تقاطع مجموعتين من البيانات يعيد الأسطر الموجودة في كليهما، أما الفرق فيعيد الأسطر الموجودة في إحداهما وغير موجودة في الأخرى.

مثال:

أوجد المدن التي تحوي ناشرين ومؤلفين (أي تقاطع مدن الناشرين مع مدن المؤلفين)

```
SELECT DISTINCT city
FROM authors
WHERE EXISTS
  (SELECT *
   FROM publishers
   WHERE publishers.city = authors.city )
```

أما المدن التي يوجد فيها مؤلفين ولا يوجد فيها ناشرين (أي الفرق بين مدن المؤلفين والناشرين) فنحصل عليها باستخدام NOT EXISTS.

مثال:

```
SELECT DISTINCT city
FROM authors
WHERE NOT EXISTS
  (SELECT *
   FROM publishers
   WHERE publishers.city = authors.city )
```

الاستعلامات الفرعية المتعددة المستويات

يمكن نسج الاستعلامات الفرعية إلى أي مستوى (حتى 32).

مثال:

أوجد قائمة المؤلفين الذين نشروا على الأقل كتابا في مجال Popular-Comp

```
SELECT au_lname, au_fname
```

```
FROM authors
```

```
WHERE au_id IN
```

```
(SELECT au_id
```

```
FROM titleauthor
```

هذا الاستعلام يعيد أرقام المؤلفين الذين ألفوا
في مجال الـ popular_comp

```
WHERE title_id IN
```

```
( SELECT title_id
```

```
FROM titles
```

هذا الاستعلام يعيد أرقام الكتب التي هي في
مجال popular_comp

```
WHERE type = 'popular_comp'
```

```
)
```

```
)
```

مثال:

نريد قائمة بنمط واسم الكتب التي يفوق سعرها السعر الوسطي لهذا النمط.

```
SELECT
```

```
    t1.type,
```

```
    t1.title
```

```
FROM
```

```
    titles t1
```

```
WHERE t1.price >
```

```
(
```

```
    SELECT
```

```
        AVG(t2.price)
```

```
    FROM
```

```
        titles t2
```

WHERE
t2.type = t1.type)

HAVING الاستعلامات الفرعية المترابطة باستخدام فقرة
يمكن استخدام الاستعلامات الفرعية المترابطة في فقرة **HAVING**

مثال:

أوجد قائمة بأنماط الكتب التي فيها أعلى دفعة مقدمة أكبر من وسطي الدفعات لهذا

النمط. الدفعة المقدمة هي الحقل الذي اسمه advance من جدول titles

الفكرة من هذا المثال أنه يمكن استخدام الاستعلام الفرعي ضمن الـ having

```
SELECT  
    t1.type  
FROM  
    titles t1  
GROUP BY  
    t1.type  
HAVING  
    MAX(t1.advance) >=  
    (  
        SELECT  AVG(t2.advance)  
        From  titles t2  
        WHERE    t2.type = t1.type  
    )
```

- نهاية المحاضرة -