

2- SQL - Subqueries

أكاديمية طويق
Tuwaiq Academy



By: Nourah Almutlag

مفهوم SQL subqueries

هي أدوات أساسية إذا كنت تريد التواصل بشكل فعال مع قواعد البيانات (Relational DB - RDMS). في هذا الدرس ، نقدم خمسة أمثلة على subquery توضح كيفية استخدام الاستعلامات الفرعية العددية والمتعددة والمتراصة في عبارات WHERE و FROM / JOIN و SELECT.

مفهوم A subquery, or nested query

هو استعلام يتم وضعه داخل استعلام SQL آخر. عند طلب معلومات من قاعدة بيانات ، قد تجد أنه من الضروري تضمين استعلام فرعي (subquery) في عبارة SELECT أو FROM أو JOIN أو WHERE. ومع ذلك ، يمكنك أيضًا استخدام الاستعلامات الفرعية (subquery) عند تحديث قاعدة البيانات (على سبيل المثال في عبارات INSERT و UPDATE و DELETE).

هناك عدة أنواع من استعلامات SQL subqueries

أولاً: Scalar subqueries

تقوم بإرجاع قيمة واحدة، أو صف واحد بالضبط وعمود واحد بالضبط.

ثانياً: Multi-row subqueries

تقوم بإرجاع إما:

- عمود واحد به عدة صفوف (أي قائمة بالقيم) ، أو أعمدة متعددة مع صفوف متعددة (مثل الجداول).

ثالثًا: Correlated subqueries

حيث يعتمد الاستعلام الداخلي على المعلومات التي تم الحصول عليها من الاستعلام الخارجي.

لنفترض أننا ندير معرضًا فنيًا. لدينا قاعدة بيانات بها أربعة جداول:

- اللوحات
- الفنانين
- المبيعات
- الهواة

يمكنك الاطلاع على البيانات المخزنة في كل جدول أدناه.

paintings			
id	name	artist_id	listed_price
11	Miracle	1	300.00
12	Sunshine	1	700.00
13	Pretty woman	2	2800.00
14	Handsome man	2	2300.00
15	Barbie	3	250.00
16	Cool painting	3	5000.00
17	Black square #1000	3	50.00
18	Mountains	4	1300.00

artists		
id	first_name	last_name
1	Thomas	Black
2	Kate	Smith
3	Natali	Wein
4	Francesco	Benelli

collectors		
id	first_name	last_name
101	Brandon	Cooper
102	Laura	Fisher
103	Christina	Buffet
104	Steve	Stevenson

sales					
id	date	painting_id	artist_id	collector_id	sales_price
1001	2021-11-01	13	2	104	2500.00
1002	2021-11-10	14	2	102	2300.00
1003	2021-11-10	11	1	102	300.00
1004	2021-11-15	16	3	103	4000.00
1005	2021-11-22	15	3	103	200.00
1006	2021-11-22	17	3	103	50.00

المثال الأول:

سنبدأ بمثال بسيط: نريد سرد اللوحات ذات الأسعار الأعلى من المتوسط. في الأساس ، نريد الحصول على أسماء اللوحات جنبًا إلى جنب مع الأسعار المدرجة ، ولكن فقط لتلك التي تكلف أكثر من المتوسط. هذا يعني أننا نحتاج أولاً إلى إيجاد متوسط السعر هذا ؛ وهنا يأتي دور الاستعلام الفرعي القياسي:

```
SELECT name, listed_price
FROM paintings
WHERE listed_price > (
  SELECT AVG(listed_price)
  FROM paintings
);
```

name	listed_price
Pretty woman	2800
Handsome man	2300
Cool painting	5000

استعلامنا الفرعي (subquery) موجود في جملة WHERE ، حيث يقوم بتصفية مجموعة النتائج بناءً على السعر المدرج. يعرض هذا الاستعلام الفرعي (subquery) قيمة واحدة: متوسط السعر لكل لوحة لمعرضنا. تتم مقارنة كل سعر مدرج بهذه القيمة ، واللوحات التي تم تسعيرها أعلى من المتوسط فقط هي التي تصل إلى الناتج النهائي.

المثال الثاني:

دعنا الآن نلقي نظرة على الاستعلامات الفرعية التي ترجع عمومًا واحدًا يحتوي على صفوف متعددة. غالبًا ما يتم تضمين هذه الاستعلامات الفرعية في جملة WHERE لتصفية نتائج الاستعلام الرئيسي. لنفترض أننا نريد سرد جميع هواة الجمع الذين اشترى لوحات من معرضنا. يمكننا الحصول على المخرجات الضرورية باستخدام استعلام فرعي متعدد المهام. على وجه التحديد، يمكننا استخدام استعلام داخلي لسرد معرفات جميع الهواة الموجودة في جدول المبيعات - ستكون هذه معرفات مطابقة لهواة الجمع الذين أجروا عملية شراء واحدة على الأقل باستخدام معرضنا. بعد ذلك ، في الاستعلام الخارجي ، نطلب الاسم الأول والاسم الأخير لجميع المُجمِّعين الذين يكون معرفهم في ناتج الاستعلام الداخلي: (Suppose we want to list all collectors who purchased paintings from our gallery).

```
SELECT first_name, last_name
FROM collectors
WHERE id IN (
    SELECT collector_id
    FROM sales
);
```

first_name	last_name
Laura	Fisher
Christina	Buffet
Steve	Stevenson

ومن المثير للاهتمام ، أنه يمكننا الحصول على نفس النتيجة بدون طلب بحث فرعي باستخدام INNER JOIN (أو JOIN فقط). يقوم نوع الصلة هذا بإرجاع السجلات التي يمكن العثور عليها في كلا الجدولين فقط. لذلك ، إذا انضمامنا إلى الهواة وجدول المبيعات ، فسنحصل على قائمة بالهواة مع السجلات المقابلة في جدول المبيعات. ملاحظة: لقد استخدمت أيضًا الكلمة الأساسية DISTINCT هنا لإزالة التكرارات من الإخراج.

```
SELECT DISTINCT collectors.first_name, collectors.last_name
FROM collectors
JOIN sales
```

ON collectors.id = sales.collector_id;

المثال الثالث:

عندما يُرجع استعلام فرعي جدولاً يحتوي على عدة صفوف وأعمدة متعددة ، يتم عادةً العثور على هذا الاستعلام الفرعي في عبارة FROM أو JOIN. يتيح لك ذلك الحصول على جدول ببيانات لم تكن متاحة بسهولة في قاعدة البيانات (مثل البيانات المجمعة grouped data) ثم ضم هذا الجدول مع جدول آخر من قاعدة البيانات الخاصة بك ، إذا لزم الأمر. لنفترض أننا نريد أن نرى المبلغ الإجمالي للمبيعات لكل فنان باع لوحة واحدة على الأقل في معرضنا. قد نبدأ باستعلام فرعي يعتمد على جدول المبيعات ويحسب المبلغ الإجمالي للمبيعات لكل معرف فنان. بعد ذلك ، في الاستعلام الخارجي ، نقوم بدمج هذه المعلومات مع الأسماء الأولى للفنانين وأسماء العائلة للحصول على المخرجات المطلوبة:

```
SELECT
  artists.first_name,
  artists.last_name,
  artist_sales.sales
FROM artists
JOIN (
  SELECT artist_id, SUM(sales_price) AS sales
  FROM sales
  GROUP BY artist_id
) AS artist_sales
ON artists.id = artist_sales.artist_id;
```

first_name	last_name	sales
Thomas	Black	300
Kate	Smith	4800
Natali	Wein	4250

نقوم بتعيين اسم مستعار ذي مغزى لإخراج طلب البحث الفرعي الخاص بنا (artist_sales). بهذه الطريقة ، يمكننا الرجوع إليها بسهولة في الاستعلام الخارجي ، عند تحديد العمود من هذا الجدول ، وعند تحديد شرط الصلة في جملة ON. ملاحظة: ستعرض قواعد البيانات خطأ إذا لم تقدم اسمًا مستعارًا لإخراج الاستعلام الفرعي. لذلك ، ضمن استعلام SQL قصير واحد ، تمكنا من حساب إجمالي المبيعات لكل فنان بناءً على البيانات الأولية من جدول واحد (المبيعات) ، ثم ضم هذا الإخراج مع البيانات من جدول آخر (الفنانين). يمكن أن تكون الاستعلامات الفرعية قوية جدًا عندما نحتاج إلى دمج المعلومات من جداول متعددة. دعونا نرى ما يمكننا فعله مع الاستعلامات الفرعية

المثال الرابع:

سنوضح في المثال التالي كيفية الاستعلامات الفرعية:

- يمكن استخدامها في عبارة SELECT و
 - يمكن أن تكون مرتبط **correlated** (أي أن الاستعلام الرئيسي أو الخارجي يعتمد على المعلومات التي تم الحصول عليها من الاستعلام الداخلي)
- لكل هواة ، نريد حساب عدد اللوحات المشتراة من خلال معرضنا. للإجابة على هذا السؤال ، يمكننا استخدام استعلام فرعي يحسب عدد اللوحات التي اشتراها كل هواة.

```
SELECT
first_name,
last_name,
(
SELECT count(*)
FROM sales
WHERE collectors.id = sales.collector_id
) AS paintings
FROM collectors;
```

first_name	last_name	paintings
Brandon	Cooper	0
Laura	Fisher	2
Christina	Buffet	3
Steve	Stevenson	1

لاحظ كيف يتم تشغيل الاستعلام الداخلي في هذا المثال بالفعل لكل صف في جدول الهواة:

يتم وضع الاستعلام الفرعي في عبارة SELECT لأننا نريد أن يكون لدينا عمود إضافي مع عدد اللوحات التي تم شراؤها بواسطة الهواة.

لكل سجل في جدول الهواة ، يقوم الاستعلام الفرعي الداخلي بحساب إجمالي عدد اللوحات المشتراة بواسطة الهواة بالمعرف **id**. كما ترى، يختلف ناتج الاستعلام الفرعي (أي عدد اللوحات) لكل سجل ويعتمد على ناتج الاستعلام الخارجي (the corresponding collector). وبالتالي، فإننا نتعامل هنا مع استعلام فرعي مرتبط **correlated**.

المثال الخامس:

هذه المرة ، نريد أن نظهر الأسماء الأولى وأسماء العائلة للفنانين الذين لم يتم بيعهم في معرضنا. دعنا نحاول إنجاز هذه المهمة باستخدام استعلام فرعي مرتبط في جملة WHERE:

```
SELECT first_name, last_name
```

```
FROM artists
WHERE NOT EXISTS (
  SELECT *
  FROM sales
  WHERE sales.artist_id = artists.id
);
```

first_name	last_name
Francesco	Benelli

إليك ما يحدث في هذا الاستعلام:

- يسرد الاستعلام الخارجي المعلومات الأساسية عن الفنانين ، ويتحقق أولاً مما إذا كانت هناك سجلات مقابلة في المبيعات
- يبحث الاستعلام الداخلي عن السجلات التي تتوافق مع معرف الفنان الذي يتم التحقق منه حالياً بواسطة الاستعلام الخارجي.
- في حالة عدم وجود سجلات مقابلة ، تتم إضافة الاسم الأول والاسم الأخير للفنان.