



مثال:

نريد عرض جميع الكتب التي سعرها أكبر من سعر كتاب Life Without Fear وحتى نقوم بتنفيذ هذه التعليمات فنحن بحاجة إلى القيام بخطوتين :

الخطوة الأولى هي جلب سعر كتاب Life Without Fear.

الخطوة الثانية هي مقارنة أسعار الكتب مع سعر هذا الكتاب لكي نستطيع معرفة الكتب التي سنحضرها

وهنا تكمن وظيفة الـ Sub Query

```
select *
from titles
where Price > (
    select price
    from titles
    where title = ' Life Without Fear ')
```



و تكون نتيجة هذه التعليمات :

<pre>select * from titles where price &gt; (select price from titles where title = 'Life Without Fear')</pre>									
100 %									
Results Messages									
	title_id	title	type	pub_id	price	advance	royalty	ytd_sales	notes
1	BU1032	The Busy Executive's Database Guide	business	1389	19.99	5000.00	10	4095	An overview of available database systems with emp...
2	BU1111	Cooking with Computers: Sureptitious Balance Sheets	business	1389	11.95	5000.00	10	3876	Helpful hints on how to use your electronic resource...
3	BU7832	Straight Talk About Computers	business	1389	19.99	5000.00	10	4095	Annotated analysis of what computers can do for yo...
4	MC2222	Silicon Valley Gastronomic Treats	mod_cook	0877	19.99	0.00	12	2032	Favorite recipes for quick, easy, and elegant meals.
5	PC1035	But Is It User Friendly?	popular_comp	1389	22.95	7000.00	16	8780	A survey of software for the naive user, focusing on t...
6	PC8888	Secrets of Silicon Valley	popular_comp	1389	20.00	8000.00	10	4095	Muckraking reporting on the world's largest compute...
7	PS1372	Computer Phobic AND Non-Phobic Individuals: Beha...	psychology	0877	21.59	7000.00	10	375	A must for the specialist, this book examines the diffe...
8	PS2091	Is Anger the Enemy?	psychology	0736	10.95	2275.00	12	2045	Carefully researched study of the effects of strong e...

ولاحظ أن الـ Sub Query أو ما يسمى بـ IN Query تعيد في نتيجتها القيمة 7

فأصبحت التعليمات السابقة كأننا نقارن أسعار الكتب مع القيمة 7

ملاحظة:

- المثال السابق هو مثال على استخدام الـ Sub Query ضمن الـ Criteria و نلاحظ في هذه الحالة أن الـ in Query هي التي سيتم تنفيذها قبل الـ Out Query وذلك لأننا لا يمكننا إحضار أي كتاب حتى نقوم بمقارنة سعره أي اعتماداً على الـ Sub Query.

الآن بالنسبة لاستخدام الـ Sub Query ضمن الـ selection فإننا نستخدمه عندما نريد جلب value من table آخر.

### مثال:

نريد جلب أسماء الكتب مع أسماء دور النشر الخاصة بها .

في الـ table title لدينا اسم الكتاب و الـ publisher\_id ولكننا لا نملك اسم دار النشر؟! هنا تكمن وظيفة الـ Sub Query .

```
Select t.title, t.pub_id ,(select p.pub_name
                        From publishers p
                        Where p.pub_id = t.pub_id) as pub_name
from titles t
```

و تكون نتيجة تنفيذ التعليمة السابقة على الشكل التالي:

The screenshot shows a SQL query window with the following query:

```
select t.title ,t.pub_id,(select p.pub_name
from publishers p
where p.pub_id = t.pub_id) as pub_name
from titles t
```

The results are displayed in a table with 7 rows:

	title	pub_id	pub_name
1	The Busy Executive's Database Guide	1389	Algodata Infosystems
2	Cooking with Computers: Surreptitious Balance Sh...	1389	Algodata Infosystems
3	You Can Combat Computer Stress!	0736	New Moon Books
4	Straight Talk About Computers	1389	Algodata Infosystems
5	Silicon Valley Gastronomic Treats	0877	Binnet & Hardley
6	The Gourmet Microwave	0877	Binnet & Hardley
7	The Psychology of Computer Cooking	0877	Binnet & Hardley

The status bar at the bottom indicates: Query executed successfully. DESKTOP-J6IKHT7 (14.0 RTM) | DESKTOP-J6IKHT7\HP (60) pubs 00:00:00 18 rows

**سؤال:** ماذا تختلف الـ Sub Query من النمط السابق (ضمن selection) عن الـ Sub Query التي تستخدم ضمن Criteria؟!

■ ضمن الـ criteria فإن الـ Query in حتماً هي التي سيتم تنفيذها أولاً فلا يمكن أن يتم جلب أي Record حتى يتم التأكد فيما إذا كان يحقق ما ضمن الـ criteria فالـ Query in تنفيذ و من ثم الـ Query out .

■ أما بالنسبة للنوع الثاني (كالمثال السابق) يتم تنفيذ الـ Query out أولاً، نلاحظ في مثالنا - على أساس الـ Query out نقوم بالتحقق من دار النشر المقابلة و نلاحظ أننا أيضاً في الـ Query in قمنا بإعطائه قيمة من الـ Query Out ليتحقق على أساسها .

## ملاحظة:

- لو لم نقم باستخدام الـ where في المثال السابق لظهر لنا error وذلك لأنه لم يعد قادراً على عملية المطابقة بعدد الـ Records بين الـ 2 Queries.
- أي قمنا بوضع where و كأننا نقول نريد دور النشر التي حتماً قامت بنشر كتب موجودة في الـ titles table.
- إن الـ t و الـ m التي رأيناها في المثال السابق تدعى alias وهي حتى يستطيع الـ Engine فهم و معرفة الوجهة التي سيقراً منها و قمنا باستخدامها عوضاً عن أن نقوم بكتابة اسم الـ table كاملاً.
- أي t.pub\_name عوضاً عن titles\_pub\_name.

## مثال آخر على استخدام الـ sub Query ضمن الـ criteria :

نريد جلب جميع دور النشر التي فيها نشر من نوع "Business" أي على الأقل قامت بنشر كتاب من هذا النوع.

```
select pub_name
from publishers
where pub_id in (select pub_id
                  from titles
                  where type = 'BUSINESS')
```

وتكون نتيجة تنفيذ التعليمة السابقة :

The screenshot shows a SQL query execution window with the following query:

```
select pub_name
from publishers
where pub_id in (select pub_id from titles
                  where type='BUSINESS')
```

The results table shows two rows:

pub_name
1 New Moon Books
2 Algodata Infosystems

At the bottom of the window, there are two cartoon characters: a red one playing a guitar and a blue one playing a guitar. The status bar at the bottom indicates "Query executed successfully." and "DESKTOP-J6IKHT7 (14.0 RTM) DESKTOP-J6IKHT7\HP (60) pubs 00:00:00 2 rows".

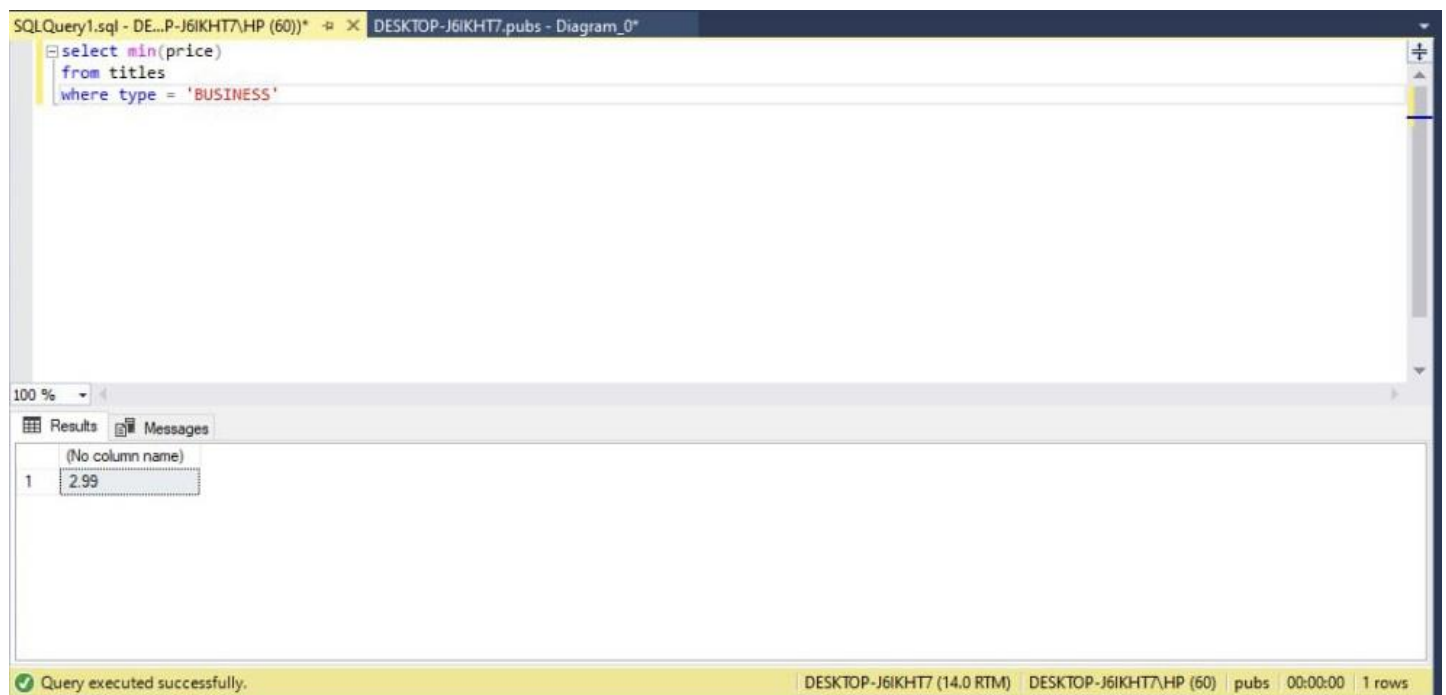
## ملاحظة:

- من الخطأ استخدام = بدلاً من in عندما تكون النتيجة هي set of values.

- نستطيع أيضاً على سبيل المثال استخدام aggregate Function ضمن الـ Sub Query. فمثلاً لو أردنا إحضار جميع الكتب التي سعرها مساوٍ لأقل سعر لكتاب من نوع 'BUSINESS'

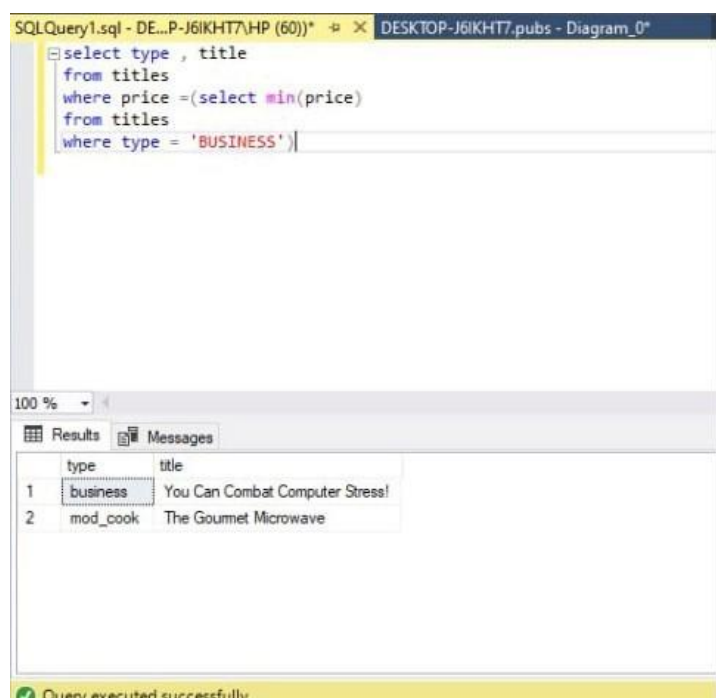
أولاً دعونا نتحقق من سعر أقل كتاب من نوع BUSINESS:

```
Select min(price)
from titles
where type = 'BUSINESS'
```



إذاً فإن الـ Query السابقة بحد ذاتها ستكون عبارة عن sub Query في الـ Query التي سنقوم بتنفيذها .

```
Select type, title
from titles
where price = (
Select min(price)
from titles
where type = 'BUSINESS' )
```





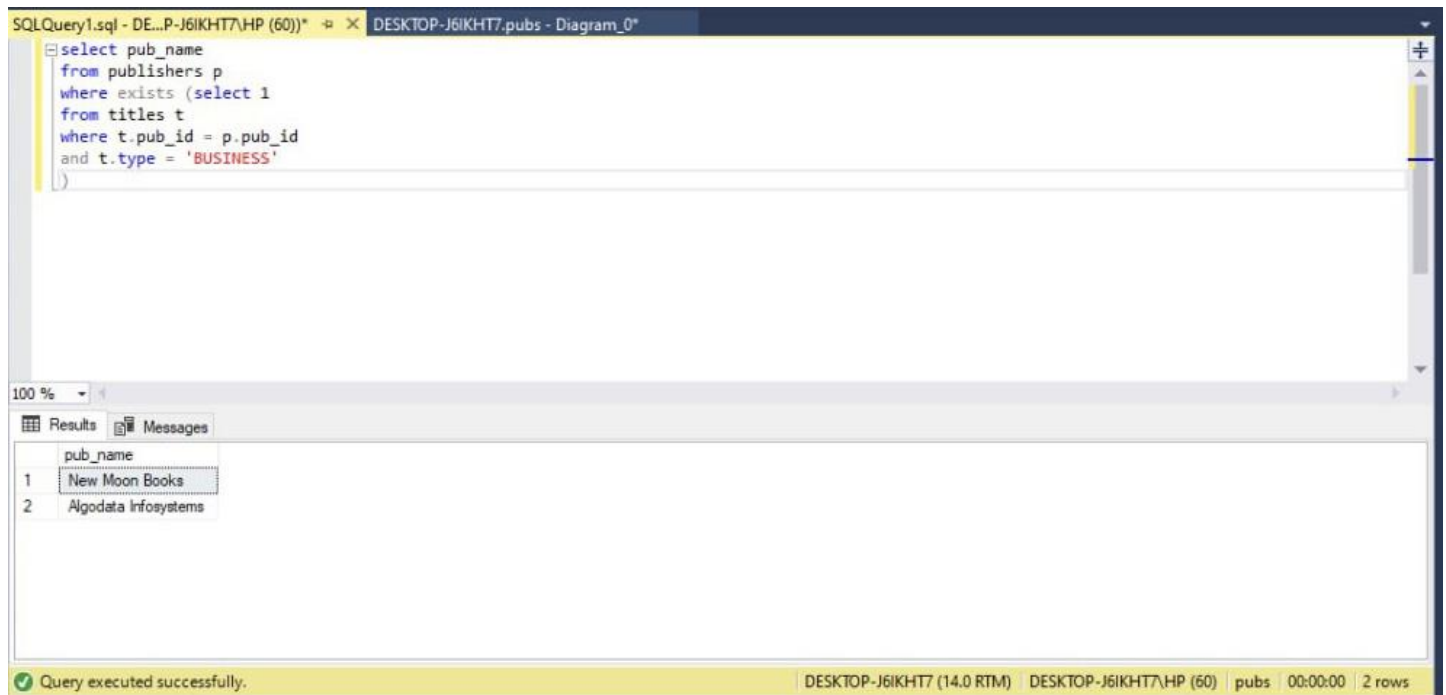
## سننتقل الآن إلى فكرة أخرى وهي Exist:

سنقوم بتطبيقها على نفس المثال الذي قمنا فيه بإحضار جميع دور النشر التي يوجد فيها نشر من نوع BUSINESS و يكون استخدام الـ Exist كالتالي :

```
select pub_name
from publishers p
where exists (select 1
from titles t
where t.pub_id = p.pub_id
and t.type = 'BUSINESS' )
```



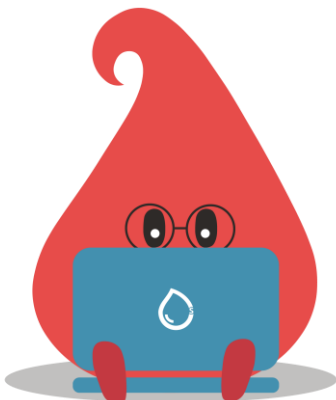
و تكون نتيجة تنفيذ التعليمة السابقة :



أي في المثال السابق مع كل دار نشر سيقوم بإحضارها سيقوم بالتحقق فيما إذا كانت دار النشر هذه لديها كتاب من ضمن الـ title من نوع 'BUSINESS' فهي بشكل عام تعيد قيمة true او false على مستوى الـ record.

### ملاحظة:

- هنا مع الـ Select في الـ Sub Query يمكننا وضع أي رمز مع الـ select لأننا في هذه الـ select لا نريد إحضار أي Result و إنما نريد من خلالها التحقق فقط .

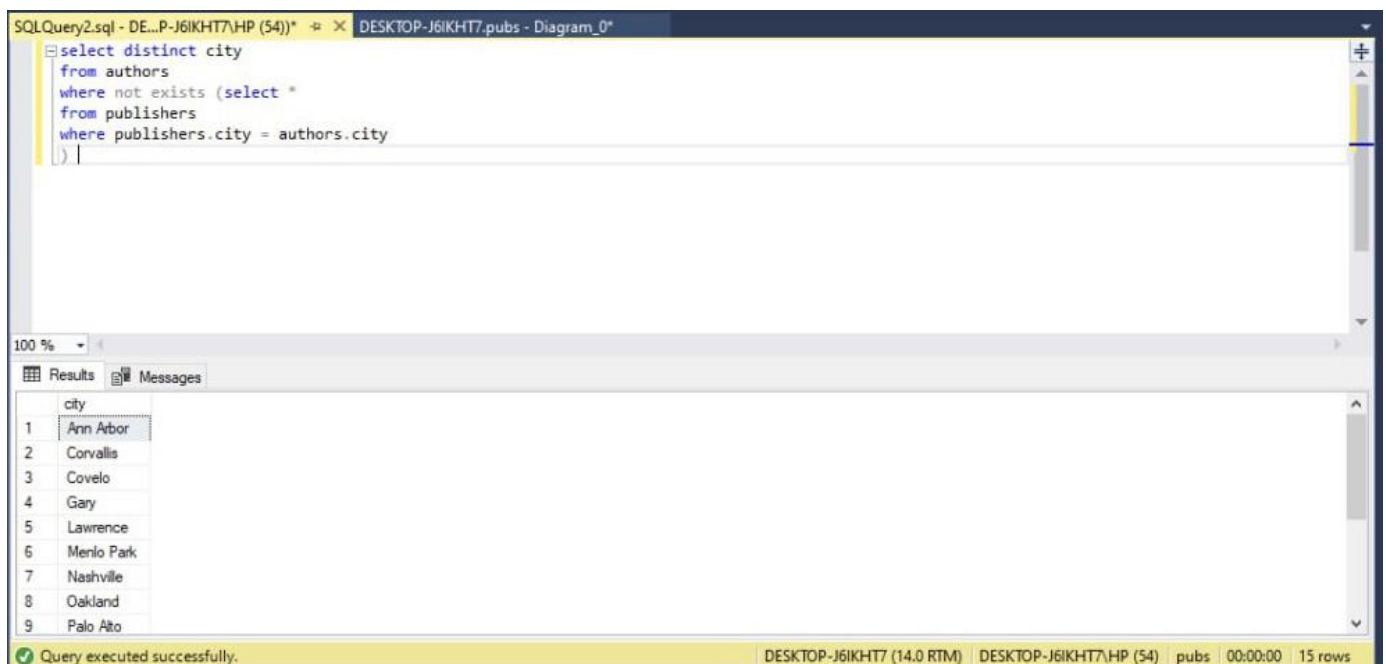


مثال آخر:

نريد جلب أسماء المدن المتواجد فيها مؤلفين وليس موجود فيها دور نشر :

```
select distinct city
from authors
where not exists (
    select *
    from publishers
    where publisher.city = authors.city )
```

وتكون نتيجة تنفيذ التعليمة السابقة :

ملاحظة:

■ قمنا في المثال السابق باستخدام not exists و التي تعاكس في إلية عملها ال exists

■ ومن الأنماط الأخرى لاستخدام ال Sub Query

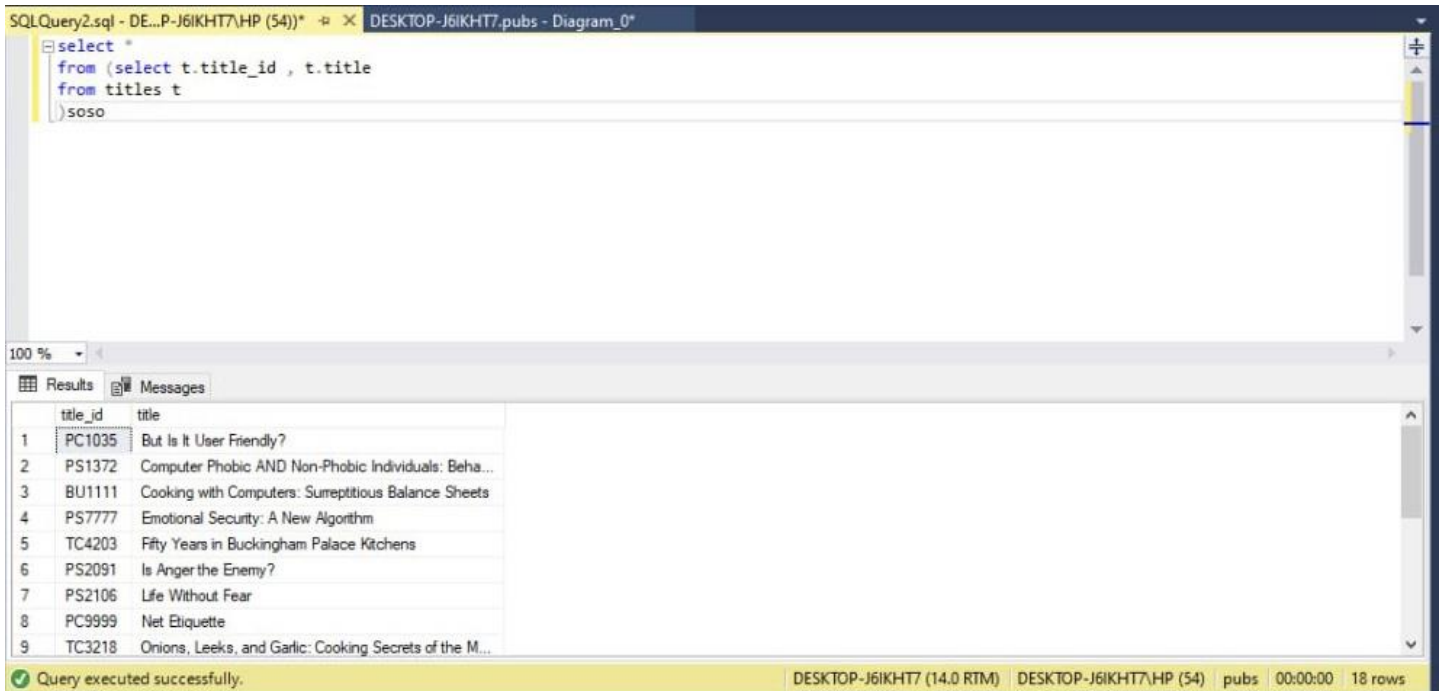
هي أن نقوم مباشرة بعمل Select من ال sub Query وفي هذا النمط تكون النتيجة dynamic View

```
select*
from(select t.title_id,t.title
from titles t
) soso
```



*Make the light inside you shine to the outside*

وتكون نتيجة تنفيذ التعليمة السابقة :



The screenshot shows a SQL query window with the following query:

```
select *
from (select t.title_id , t.title
from titles t
) soso
```

The results pane shows the following data:

title_id	title
PC1035	But Is It User Friendly?
PS1372	Computer Phobic AND Non-Phobic Individuals: Beha...
BU1111	Cooking with Computers: Surreptitious Balance Sheets
PS7777	Emotional Security: A New Algorithm
TC4203	Fifty Years in Buckingham Palace Kitchens
PS2091	Is Anger the Enemy?
PS2106	Life Without Fear
PC9999	Net Etiquette
TC3218	Onions, Leeks, and Garlic: Cooking Secrets of the M...

The status bar at the bottom indicates: Query executed successfully. DESKTOP-J6IKHT7 (14.0 RTM) DESKTOP-J6IKHT7\HP (54) pubs 00:00:00 18 rows

و سنرى الفائدة من استخدام هذا النمط لاحقاً

حيث يمكننا على سبيل المثال أن نقوم ببناء Dynamic Query و إعطائها اسم معين ثم سنقوم لاحقاً باستخدامها مع العديد من الـ Queries الأخرى.



### ■ فكرة أخرى:

لو لاحظنا أن في الـ publishers table هناك دور نشر لم تقم بنشر أي كتاب.  
و أيضاً بالمقابل هناك كتب في الـ titles table لم يتم نشرها من قبل أي دار نشر .

وهنا جاءت SQL بفكرة الـ Join

والتي تستخدم بشكل رئيسي لجمع الـ Rows من 2 table أو أكثر اعتماداً على columns من هذه الـ Tables و هنالك عدة أنواع للـ Join وهي :

Cross Join , inner Join , left Join , right Join

### مثال:

نريد إحصار كل الكتب مع دار النشر الخاصة بها

```
select t.title , p.pub_name
from titles t
inner join publishers p
on t.pub_id = p.pub_id
```

*The best time to plant a tree  
was 20 years ago. The  
second best time is now*



وتكون النتيجة كالتالي:

```

select t.title , p.pub_name
from titles t
inner join publishers p
on t.pub_id = p.pub_id

```

	title	pub_name
1	The Busy Executive's Database Guide	Algodata Infosystems
2	Cooking with Computers: Sureptitious Balance Sh...	Algodata Infosystems
3	You Can Combat Computer Stress!	New Moon Books
4	Straight Talk About Computers	Algodata Infosystems
5	Silicon Valley Gastronomic Treats	Binnet & Hardley
6	The Gourmet Microwave	Binnet & Hardley
7	The Psychology of Computer Cooking	Binnet & Hardley
8	But Is It User Friendly?	Algodata Infosystems
9	Secrets of Silicon Valley	Algodata Infosystems

Query executed successfully. DESKTOP-J6IKHT7 (14.0 RTM) DESKTOP-J6IKHT7\HP (54) pubs 00:00:00 18 rows

ملاحظة:

- في المثال السابق لو لم نقم بوضع شرط للربط بين الـ 2 table في الـ inner join باستخدام on, فيكون قد أصبح الـ join في هذه الحالة هو Cross join والذي هو أشبه بالجداء الديكارتي حيث كان سي جلب مع كل دار نشر كل الكتب الموجودة في جدول الـ titles .... لذلك فإن الـ Cross join لا يفضل استخدامها في أي حالة.
- يجب عند استخدام الـ inner join استخدام on لوضع شرط الـ join .
- يمكننا أيضاً مع الـ join استخدام criteria و شروط منطقية كالمعتاد.

```

select t.title , p.pub_name
from titles t
inner join publishers p
on t.pub_id = p.pub_id
where t.price > 10
and p.country = 'USA'

```

	title	pub_name
1	The Busy Executive's Database Guide	Algodata Infosystems
2	Cooking with Computers: Sureptitious Balance Sheets	Algodata Infosystems
3	Straight Talk About Computers	Algodata Infosystems
4	Silicon Valley Gastronomic Treats	Binnet & Hardley
5	But Is It User Friendly?	Algodata Infosystems
6	Secrets of Silicon Valley	Algodata Infosystems
7	Computer Phobic AND Non-Phobic Individuals: Beha...	Binnet & Hardley
8	Is Anger the Enemy?	New Moon Books
9	Prolonged Data Deprivation: Four Case Studies	New Moon Books

Query executed successfully. DESKTOP-J6IKHT7 (14.0 RTM) DESKTOP-J6IKHT7\HP (54) pubs 00:00:00 12 rows

### ■ بالنسبة لـ Left join و الـ Right join:

■ Left join: تقوم بعمل الـ inner join إضافة إلى أنها تقوم بإحضار الـ records من الـ left table والتي ليس لها مقابل في الـ Right.

■ Right join: تقوم بعمل الـ inner join إضافة إلى أنها تقوم بإحضار الـ records من الـ right table والتي ليس لها مقابل في الـ left.

```
select t.title , p.pub_name
from titles t
left outer join publishers p
on t.pub_id = p.pub_id
```



ولكن في هذه الحالة لن نلاحظ الفرق لأنه لا يوجد أي كتاب بدون دار نشر وستظهر النتيجة كما في الـ inner join.

```
select t.title , p.pub_name
from titles t
on t.pub_id = p.pub_id
```

فتكون نتيجة تنفيذ التعليمة السابقة :

The screenshot shows a SQL query window with the following query:

```
select t.title , p.pub_name
from titles t
right outer join publishers p
on t.pub_id = p.pub_id
```

The results are displayed in a table with two columns: title and pub\_name. The table contains 23 rows. The first three rows show titles with corresponding publishers. The remaining 20 rows show NULL for the title column and the publisher name for the corresponding publisher.

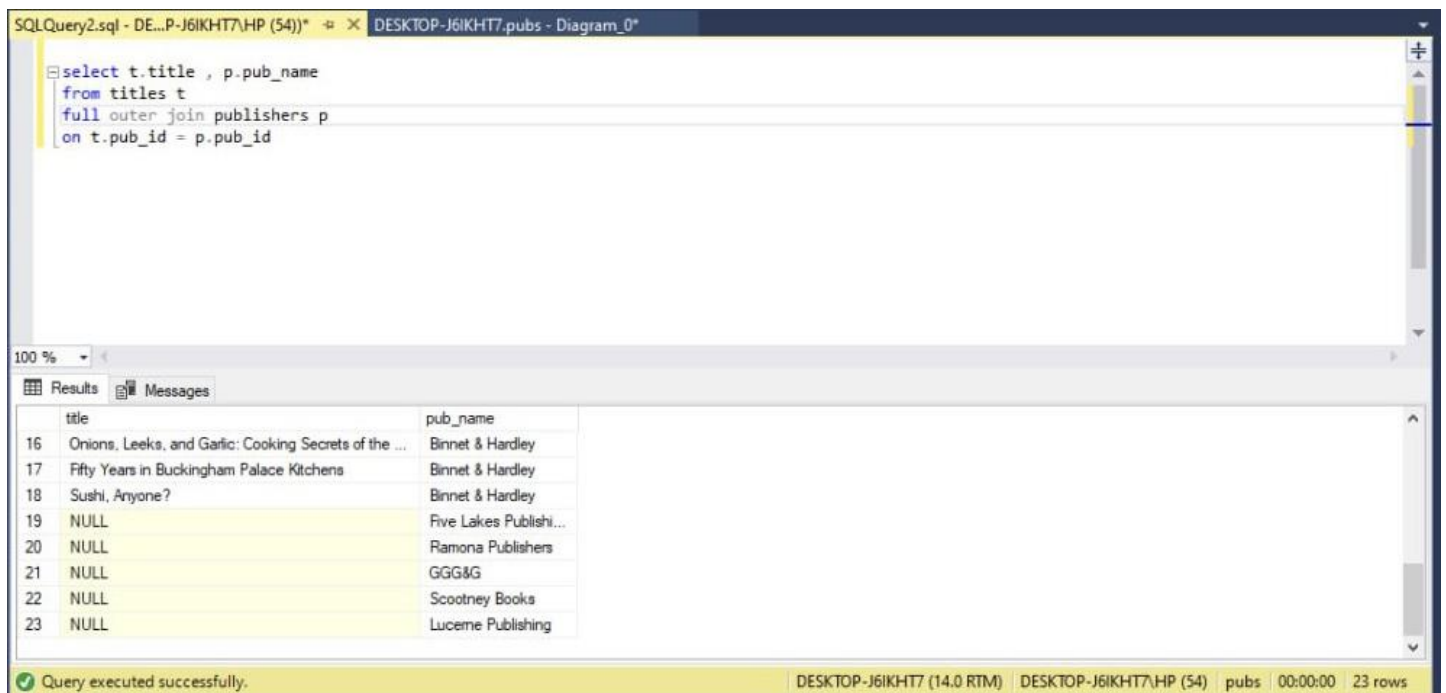
title	pub_name
But Is It User Friendly?	Algodata Infosy...
Secrets of Silicon Valley	Algodata Infosy...
Net Etiquette	Algodata Infosy...
NULL	Five Lakes Publ...
NULL	Ramona Publish...
NULL	GGG&G
NULL	Scootney Books
NULL	Luceme Publish...

Query executed successfully.

و نلاحظ، بالنتيجة ظهور دور النشر التي لم تقم بنشر أي كتب.

بالنسبة للـ full join فهي left join و right join في آن معاً أي من خلالها نستطيع جلب الكتب التي ليس لها دور نشر و دور النشر التي لم تقم بنشر أي من الكتب .

```
select t.title , p.pub_name
from titles t
full outer join publishers p
on t.pub_id = p.pub_id
```



The screenshot shows a SQL query window with the following text:

```
select t.title , p.pub_name
from titles t
full outer join publishers p
on t.pub_id = p.pub_id
```

Below the query window, the 'Results' pane displays the following data:

	title	pub_name
16	Onions, Leeks, and Garlic: Cooking Secrets of the ...	Binnet & Hardley
17	Fifty Years in Buckingham Palace Kitchens	Binnet & Hardley
18	Sushi, Anyone?	Binnet & Hardley
19	NULL	Five Lakes Publishi...
20	NULL	Ramona Publishers
21	NULL	GGG&G
22	NULL	Scotney Books
23	NULL	Luceme Publishing

The status bar at the bottom indicates: Query executed successfully. DESKTOP-J6IKHT7 (14.0 RTM) DESKTOP-J6IKHT7\HP (54) pubs 00:00:00 23 rows

ولكن كل الأمثلة السابقة على الـ Join كانت في حال أن العلاقة صريحة بين الـ tables ولكن ماذا لو كانت العلاقة بين 2 table غير صريحة؟! أي أن هناك جدول كسر بينهما .



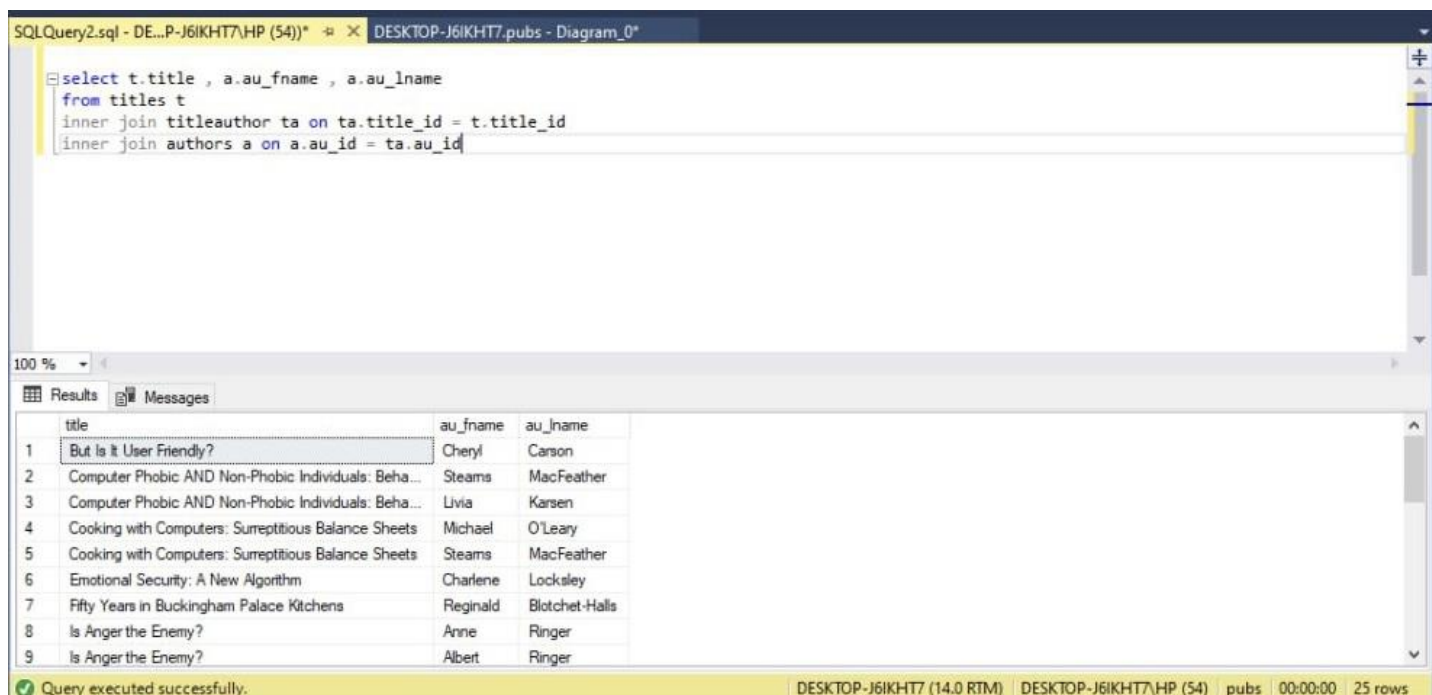
■ على سبيل المثال نريد جلب أسماء الكتب مع مؤلفيها  
كما نعلم أن العلاقة بين هذين الجدولين هي  
many to many ويوجد بينهما جدول كسر وهو الـ title author

Setting goal is the first step  
in turning the invisible into  
the visible

في هذه الحالة يكون استخدام join كالتالي:

```
select t.title , a.au_fname, a.au_lname
from titles t
inner join titleauthor ta
on t.title_id = ta.title_id
inner join author a
on a.au_id = ta.au_id
```

و تكون نتيجة تنفيذ التعليمة السابقة كالتالي:



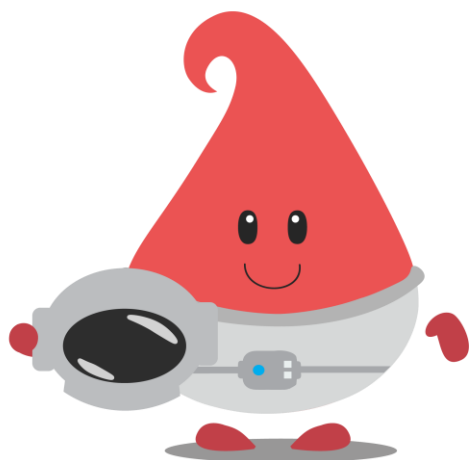
The screenshot shows a SQL query window with the following text:

```
select t.title , a.au_fname , a.au_lname
from titles t
inner join titleauthor ta on ta.title_id = t.title_id
inner join authors a on a.au_id = ta.au_id
```

Below the query window, the 'Results' tab is active, displaying a table with 9 rows and 3 columns: title, au\_fname, and au\_lname.

	title	au_fname	au_lname
1	But Is It User Friendly?	Cheryl	Carson
2	Computer Phobic AND Non-Phobic Individuals: Beha...	Steams	MacFeather
3	Computer Phobic AND Non-Phobic Individuals: Beha...	Livia	Karsen
4	Cooking with Computers: Surreptitious Balance Sheets	Michael	O'Leary
5	Cooking with Computers: Surreptitious Balance Sheets	Steams	MacFeather
6	Emotional Security: A New Algorithm	Charlene	Locksley
7	Fifty Years in Buckingham Palace Kitchens	Reginald	Blotchet-Halls
8	Is Anger the Enemy?	Anne	Ringer
9	Is Anger the Enemy?	Albert	Ringer

At the bottom of the window, a status bar indicates: 'Query executed successfully. DESKTOP-J6IKHT7 (14.0 RTM) | DESKTOP-J6IKHT7\HP (54) pubs 00:00:00 25 rows'.



If you can dream it,  
you can do it

The End ♥