



# مقدمة في لغة SQL وبرنامج SQL Server

عملي مشترك

محتوى مجاني غير مخصص للبيع التجاري

02/05/2023

12

4

عملي

840

S.P



## RB Informatics ;

## قواعد معطيات 1

تحدثنا في المحاضرات السابقة عن مخططات ال ERD وتصميمها وقمنا بحل عدة مسائل لتعزيز فكرة ال ERD وال Entities والترابط بينها وبين ال Relations . هذه المحاضرة هي المحاضرة الأولى من سلسلة محاضرات ال SQL والتي ستكون أولى خطواتنا للتعامل مع ال Database بشكل فعلي.

لنبدأ على بركة الله

### بداية ما هي ال SQL ؟

- SQL هي لغة برمجة تستخدم لإدارة قواعد البيانات والعمليات على البيانات وهي لغة التخاطب مع ال Database وهي اختصار ل **Structured Query Language**.
- في المحاضرات السابقة تكلمنا عن مخططات ال ERD بشكل عام وال Entities وقلنا أنه في مفهوم ال Database فإن هذه ال Entities يتم تحويلها ل Tables . وهذا مفهوم ال Database بشكل عام (تحويل ال Entities إلى Tables) وال Attributes الخاصة بكل Entity تصبح حينها Columns في هذا ال Table , وال Data التي سنملاؤها هذه ال Table تكمن في ال Rows أو ما يعرف ب Records .
- وهذا ال Table ككل نطلق عليه اسم ال Database Object , وبشكل عام إن ال Object في ال Database قد يكون Index, Sequence, Table... الخ. فالآن نحن نريد التواصل مع هذا ال Object الرئيسي (Table) وللقيام بذلك لدينا ما يسمى بال SQL وهي كما تحدثنا تعتبر من اللغات القادرة على التخاطب مع ال Database.

### ولكن ما هي ال Tool التي سنقوم باستخدامها والتطبيق عليها؟!

إن ما سنتعلمه في هذا المقرر هو ال Pure SQL أي أنه يعمل على أي Engine مثل MySQL, MariaDB, Oracle... ولكن ما ينصح استخدامه بشدة هو ال SQL Server وذلك لأنها ال Tool الوحيدة التي تحوي ال Management وليست عبارة عن Viewer فقط. إذ أنه يمكن من خلالها القيام ب Customize لل Data بسهولة.



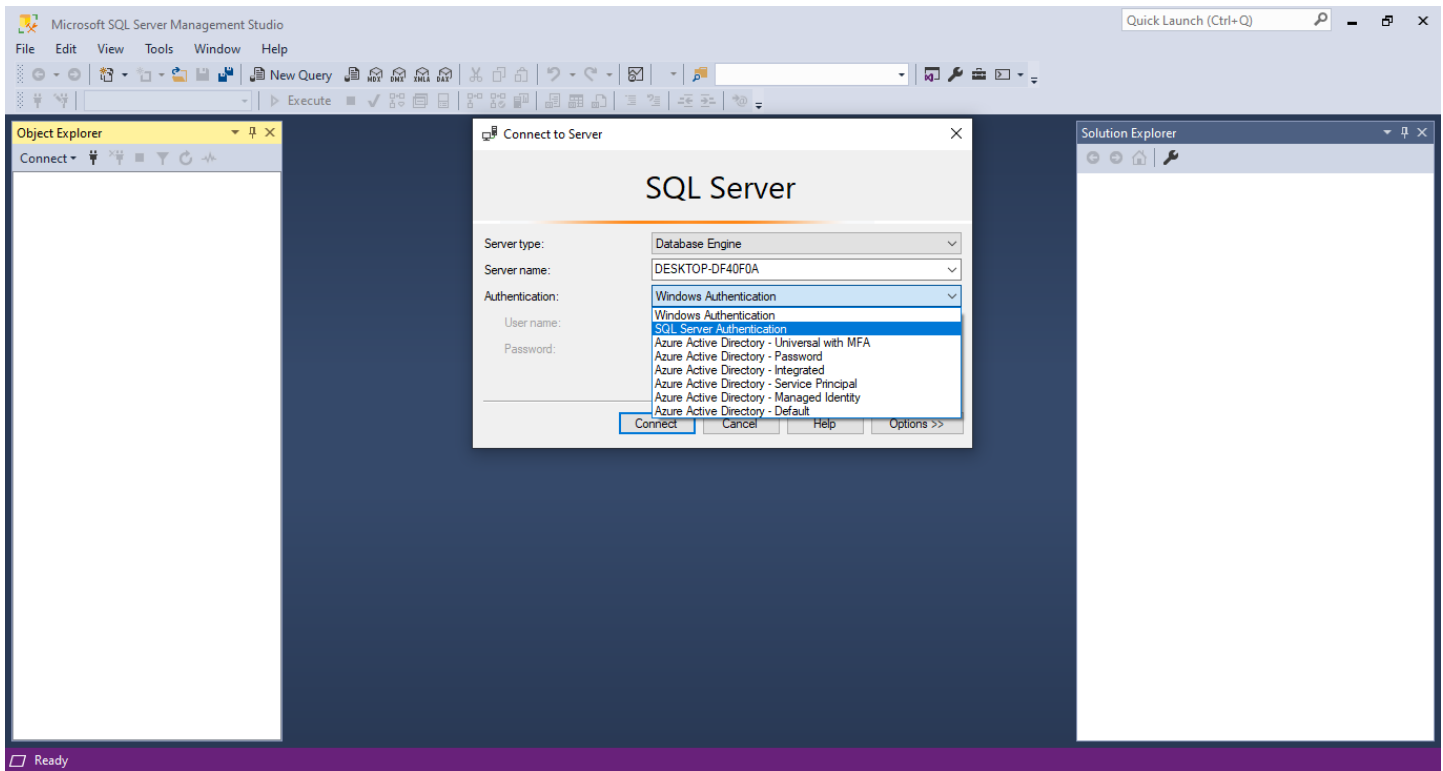
"Without big data, you are blind and deaf  
and in the middle of a freeway."

■ **ملاحظة:** أي نسخة من SQL Server بعد 2008 فهي تفني بالغرض، إذ أنه بالتأكيد يوجد تفاوت بالFeatures بين هذه النسخ ولكن كما تحدثنا فإن هذه الFeatures لا تهمنا حالياً في هذا المقرر.

■ عند تنصيب SQL Server وعندما نريد القيام ب Connect لل Engine الخاص ب SQL Server، فإنه يكون لدينا خيارين بالنسبة لل Authentication: إما **Windows Authentication** أو **SQL Server Authentication**.

■ والفرق أن ال Windows Authentication تعطي صلاحيات للدخول إلى ال Database للقراءة فقط، بينما الدخول وبناء ال User ضمن SQL Server Authentication تتيح للقراءة والتعديل أيضاً على ال Database.

■ ولكن نحن في الحالة العامة وفي أغلب الأحيان سنقوم بالعمل ب Windows Authentication.



الآن نعود للحديث عن لغة SQL:

إذاً فقد تم تقسيم هذه اللغة إلى 4 Levels أو 4 مقاطع تبعاً لأنواع التعليمات فيها:

- |  |                                      |
|--|--------------------------------------|
| 1) Data Manipulation Language – DML:   | SELECT – INSERT – UPDATE – DELETE... |
| 2) Data Definition Language – DDL:     | CREATE – DROP – ALTER...             |
| 3) Data Control Language – DCL:        | GRANT – REVOKE...                    |
| 4) Transaction Control Language – TCL: | COMMIT – ROLLBACK – SAVEPOINT...     |

*“Data is the new oil.”*

ولكن في مقررنا لهذا الفصل سيكون الحديث بشكل رئيسي عن الـ Level الأول وأنواع التعليمات فيه (DML)، وأيضاً سنتطرق قليلاً إلى الـ DDL وسنتكلم عن بعض التعليمات فيه.

## شرح لبعض التعليمات:



- **INSERT**: نستطيع من خلالها إدخال Data جديدة إلى الـ Table.
- **UPDATE**: نستطيع من خلالها تعديل Data معينة في الـ Table.
- **DELETE**: نستطيع من خلالها حذف Data معينة في الـ Table.
- **SELECT**: وهي التعليمة الوحيدة التي يمكننا من خلالها القراءة من الـ Table وأغلب عملنا سيكون باستخدام تعليمة SELECT.

## إذا سنبداً بتعليمة SELECT, ما هي الأقسام الخاصة بهذه التعليمة؟

SELECT Selection From Criteria

ويضاف إليها GROUP BY, ORDER BY. ولكن الآن محور حديثنا سيكون عن المقاطع الثلاث الأولى.

- فليكن لدينا على سبيل المثال Table خاص بالموظفين "emp" وفيه بيانات الموظفين ومعلوماتهم، فإذا أردنا على سبيل المثال عمل Query نريد فيها استرجاع اسم الموظف ورقمه فسوف يكون شكلها كالتالي:

SELECT ID, FirstName From emp

هنا حددنا الذي نريد أن نصل إليه من هذه الـ Table أي الـ Columns

هنا حددنا الـ Source الذي نريد الذهاب إليه (Table)

- وعند تنفيذ هذه التعليمة سيظهر لنا Viewer وهو عبارة عن Table بسيط يحوي 2 Columns هما الـ ID والـ FirstName. وكما قلنا هو فقط Viewer أي أن الـ Table الأساسية تبقى كما هي ولا تتأثر بما يظهر في الـ Viewer.

## ماذا لو أردنا عرض كامل بيانات الـ Table الأساسية في الـ Viewer؟!

SELECT \* From emp

إذ أن (\*) هي دلالة على كل الحقول أو كل الـ Columns، وطالما أنه ليس هناك Criteria فسيقوم بإحضار كل الـ Records الموجودة في الـ Table emp. بالتأكيد فإن كل هذا سنقوم بشرحه بالتفصيل عند التطبيق العملي ولكن سنقوم الآن بتوضيح بعض الأمور:

1. عندما نقوم بعمل Query لـ SELECT معينة:

الـ Result تكون عبارة عن View أي يمكن اعتبارها على أنها "لقطة" من الـ Table الأساسي.

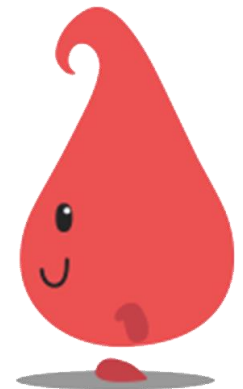
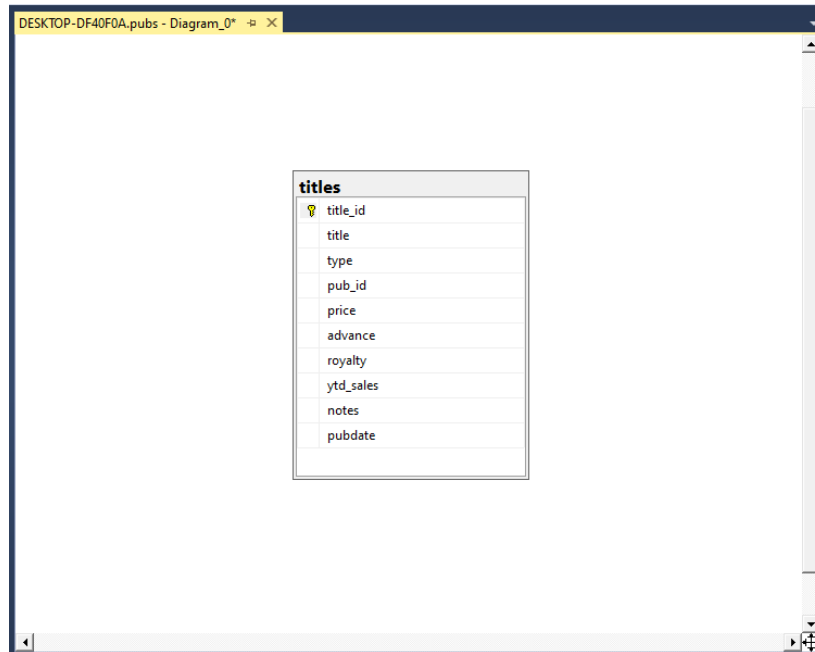
2. إن الـ SELECT لديها خصائص أخرى يمكننا استخدامها:

فيمكننا من خلالها على سبيل المثال: القيام بعمليات حسابية، دمج حقول، دمج Strings، Decode. ولكن بالتأكيد كل هذا على مستوى الـ View فكما تحدثنا الـ Table الأساسية تبقى ثابتة كما هي.

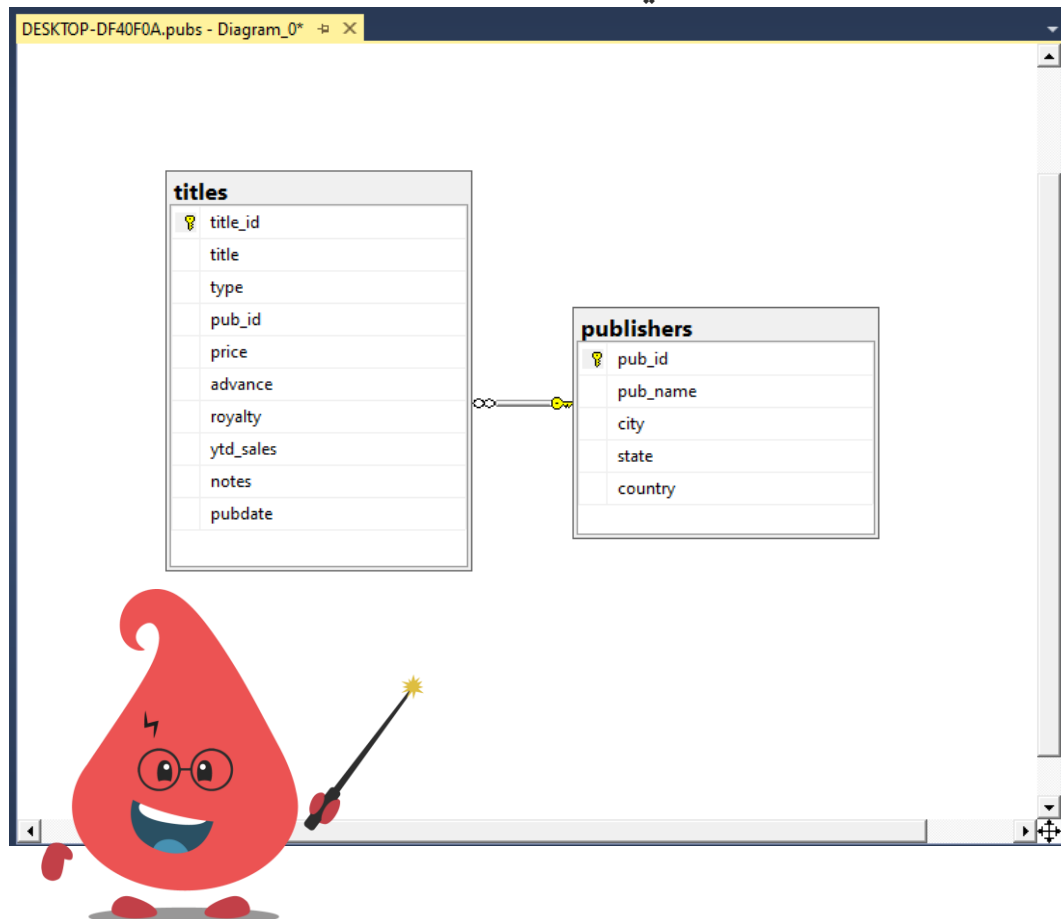
"Data that is loved tends to survive."

## الآن سنذهب إلى التطبيق العملي:

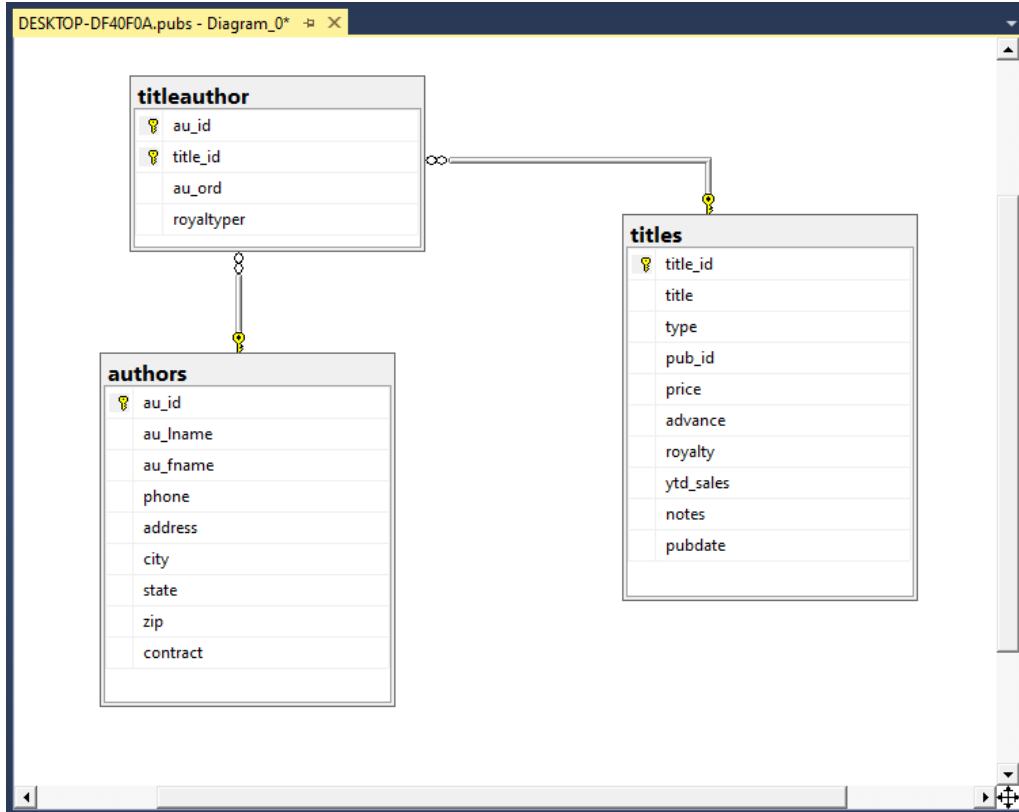
- لكي نعمل على ال Database الخاصة ب SQL Server فإننا سنعتمد على مثال يدعى Pubs وهي عبارة عن Database خاصة ببيع الكتب. **لنبدأ:**



- لدينا أول كيان titles والذي يعبر عن الكتاب وفيه العديد من attributes من نوع وسعر و .... ويعتبر هذا الكيان هو الكيان الرئيسي ومن ضمن attributes الخاصة به نجد pub\_id وهو بالتأكيد عبارة عن foreign key من كيان publishers أي أن كيان publishers هو parent لكيان titles.



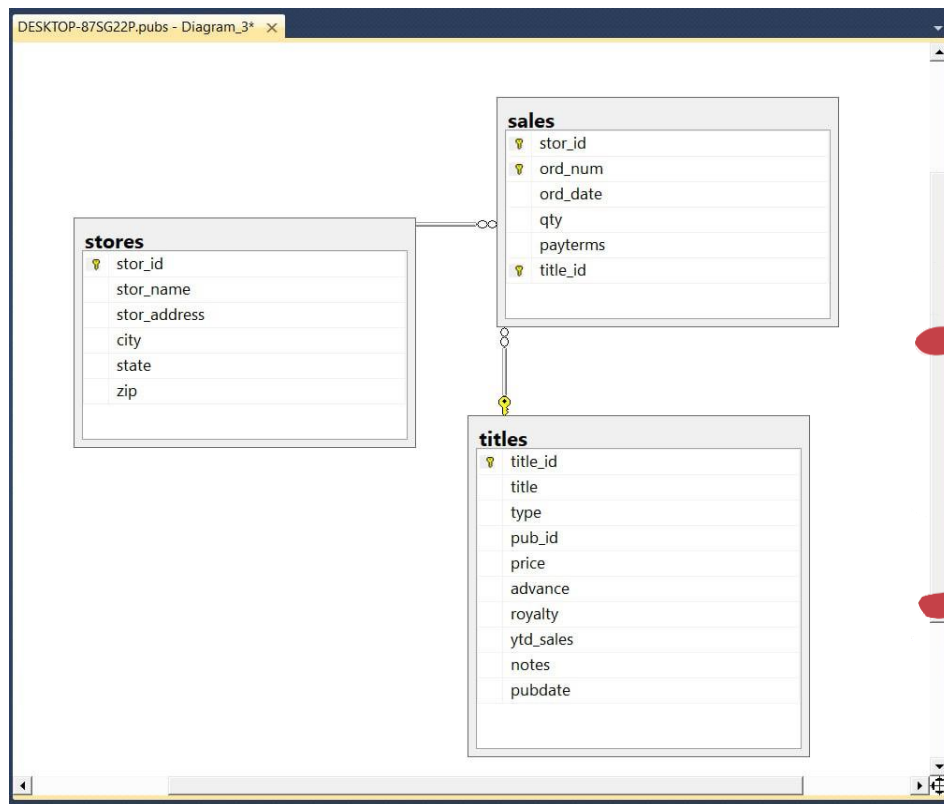
- والعلاقة بين هذين الكيانين هي one to many من طرف ال publishers.
- فيمكن لدار النشر نشر أكثر من كتاب ولكن لا يمكن لكتاب ما أن يتم نشره من قبل أكثر من دار نشر.
- وأيضاً لدينا كيان authors والذي يعبر عن المؤلفين.
- ومنطقياً فإن هناك علاقة بين ال authors وال titles هي من نوع many to many.
- إذ أنه يمكن للمؤلف أن يؤلف أكثر من كتاب ويمكن أيضاً أن يكون هنالك كتاب من تأليف مؤلف أو أكثر.
- ولذلك توجب كسر هذه العلاقة باستخدام الكيان titleauthor والذي يحوي foreign keys من هذه الكيانات.



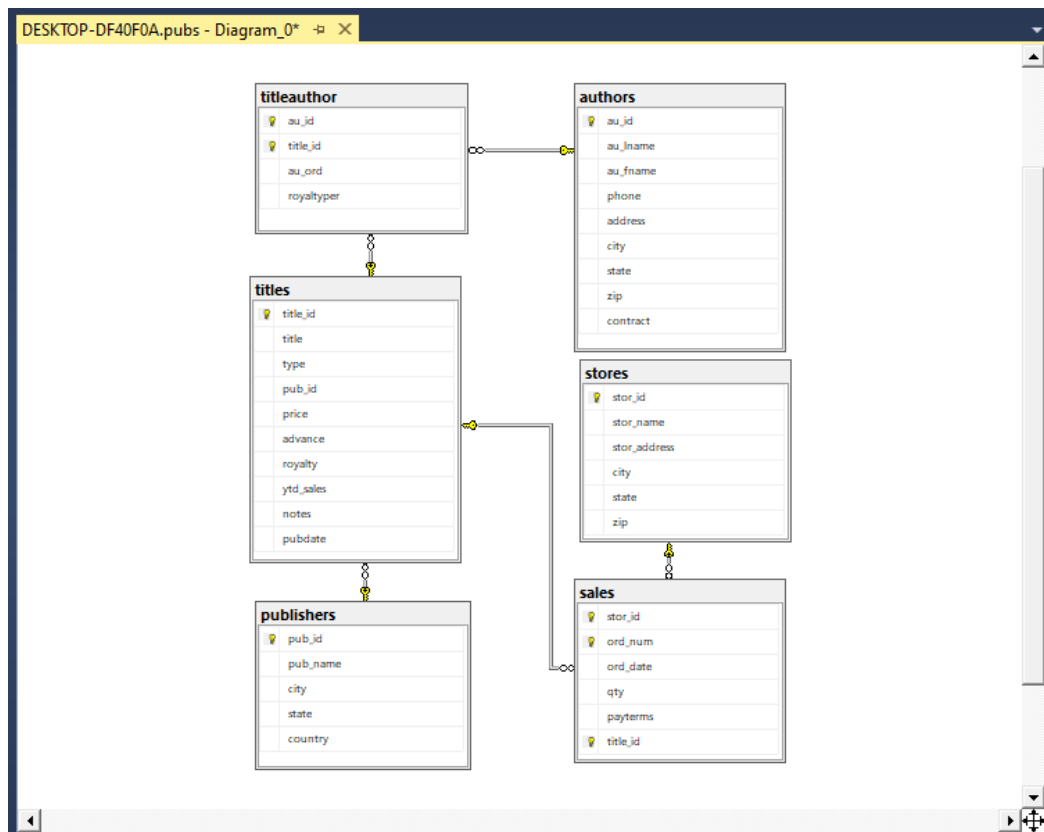
- وأيضاً يوجد لدينا كيان stores والذي يعبر عن المتاجر التي تباع فيها الكتب.
- وبشكل منطقي أيضاً فإن العلاقة بين كيان ال stores وال titles هي many to many.
- فيمكن أن يقوم متجر ما ببيع أكثر من كتاب ويمكن لكتاب معين أن يتم بيعه في متجر أو أكثر.
- وقد تم كسر هذه العلاقة باستخدام الكيان sales والذي يعبر عن المبيعات وتم وضع ال foreign keys فيه.



“You are valuable just because you exist. Not because of what you do or what you have done, but simply because you are.”



وسنكتفي بهذه الكيانات لهذه المحاضرة. إذاً فيكون الآن شكل ال Diagram الخاص بال Database لدينا لحد الآن هو كالتالي:

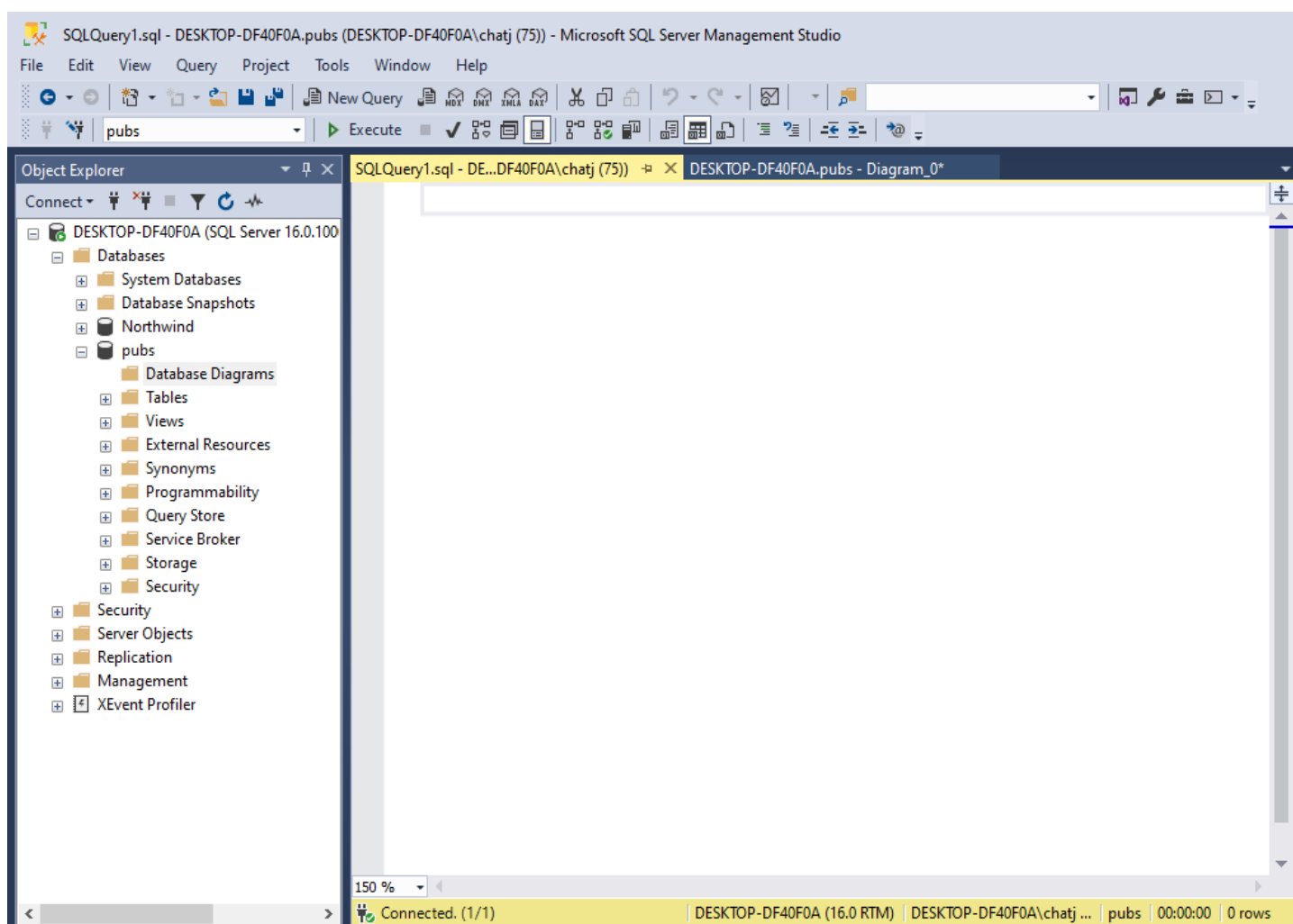


## الآن سنقوم ببعض التطبيق العملي:

- لكي نقوم بعمل Queries في ال SQL Server يجب أن نقوم بعمل New Query:



- وعند الضغط عليه يقوم بفتح editor جديد:



- وهنا يمكننا الآن كتابة التعليمات التي نريد.



"Life is short. Smile while you still have teeth."



على سبيل المثال لو قمنا بكتابة: `SELECT * From titles` ثم قمنا بالضغط على Execute :

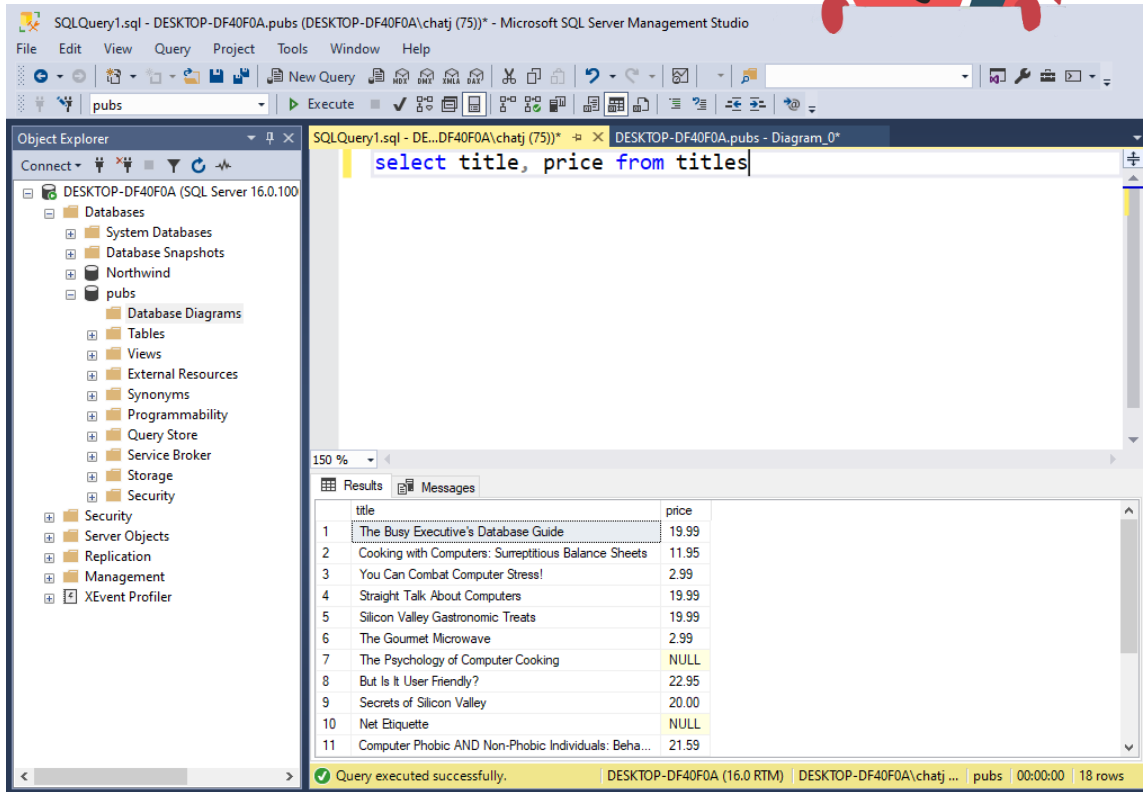
	title_id	title	type	pub_id	price	advance	royalty	ytd_s
1	BU1032	The Busy Executive's Database Guide	business	1389	19.99	5000.00	10	4095
2	BU1111	Cooking with Computers: Surreptitious Balance Sheets	business	1389	11.95	5000.00	10	3876
3	BU2075	You Can Combat Computer Stress!	business	0736	2.99	10125.00	24	1872
4	BU7832	Straight Talk About Computers	business	1389	19.99	5000.00	10	4095
5	MC2222	Silicon Valley Gastronomic Treats	mod_cook	0877	19.99	0.00	12	2032
6	MC3021	The Gourmet Microwave	mod_cook	0877	2.99	15000.00	24	2224
7	MC3026	The Psychology of Computer Cooking	UNDECIDED	0877	NULL	NULL	NULL	NUL
8	PC1035	But Is It User Friendly?	popular_comp	1389	22.95	7000.00	16	8780
9	PC8888	Secrets of Silicon Valley	popular_comp	1389	20.00	8000.00	10	4095
10	PC9999	Net Etiquette	popular_comp	1389	NULL	NULL	NULL	NUL

- نلاحظ أنه عند ذلك يقوم بإحضار جميع ال Columns وال Records الخاصة بال titles, أي أننا عندما وضعنا \* فذلك يعني أننا لا نريد عمل فلترة لل attributes الخاصة بال titles بل نريدها جميعها.
- وطالما أننا أيضاً لم نستخدم Criteria في التعليمة أي لم نقوم بعمل فلترة لل Values فعندها سيقوم بإحضار جميع ال Records الخاصة بال titles وعرضها.
- فلو أردنا على سبيل المثال إحضار بعض من ال Data وليس جميعها، مثلاً: نريد عرض اسم الكتاب مع سعره:  
`SELECT title, price From titles`
- لاحظ أننا استخدمنا الفاصلة ( , ) للفصل بين ال attributes ففي هذه الحالة سيكون ال View عبارة عن 2 Columns منفصلين.



“Just when the caterpillar thought the world was ending, he turned into a butterfly.”





The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL statement:

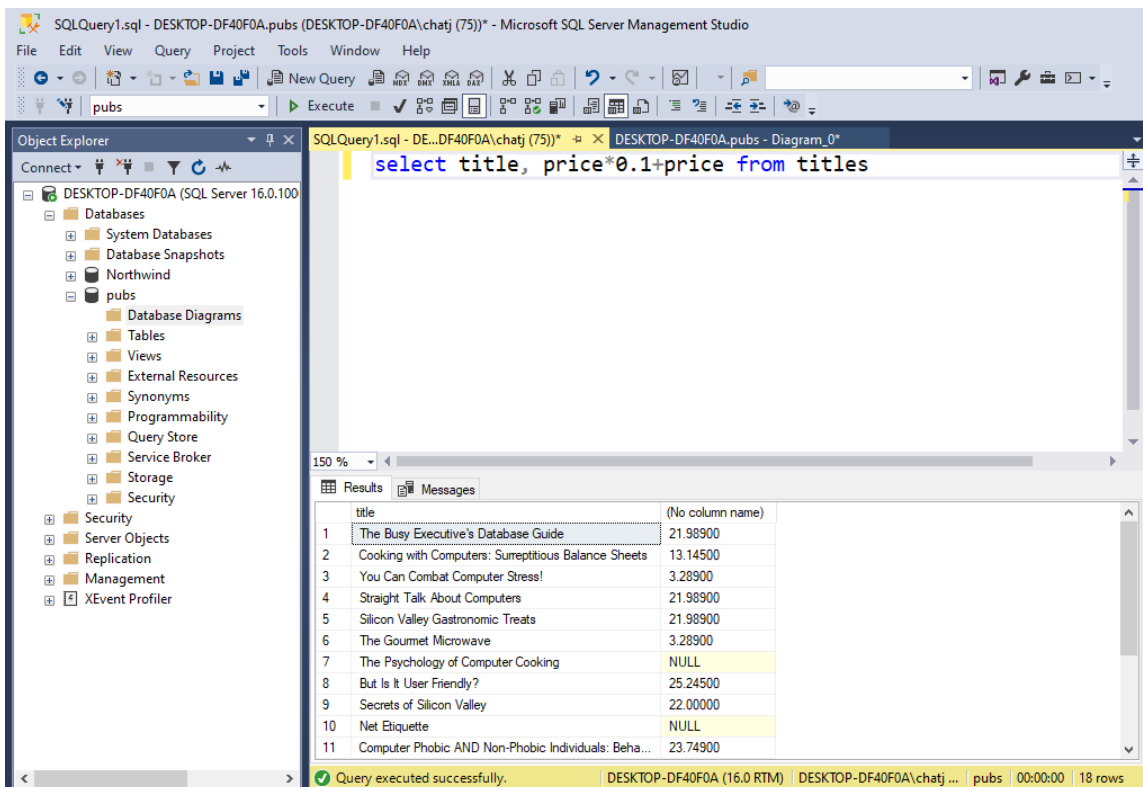
```
select title, price from titles
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show 11 rows of data from the 'titles' table in the 'pubs' database.

	title	price
1	The Busy Executive's Database Guide	19.99
2	Cooking with Computers: Surreptitious Balance Sheets	11.95
3	You Can Combat Computer Stress!	2.99
4	Straight Talk About Computers	19.99
5	Silicon Valley Gastronomic Treats	19.99
6	The Gourmet Microwave	2.99
7	The Psychology of Computer Cooking	NULL
8	But Is It User Friendly?	22.95
9	Secrets of Silicon Valley	20.00
10	Net Etiquette	NULL
11	Computer Phobic AND Non-Phobic Individuals: Beha...	21.59

The status bar at the bottom indicates: Query executed successfully. DESKTOP-DF40F0A (16.0 RTM) DESKTOP-DF40F0A\chatj ... pubs | 00:00:00 | 18 rows

- ذكرنا أنه يمكن ل SELECT القيام بعمليات حسابية أو Decode, لنأخذ مثلاً على ذلك:  
لو أردنا أن نعرض اسم الكتاب مع سعره بعد الزيادة ولتكن الزيادة فرضاً 10%, يمكننا بكل بساطة تنفيذ  
التعليمة :



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL statement:

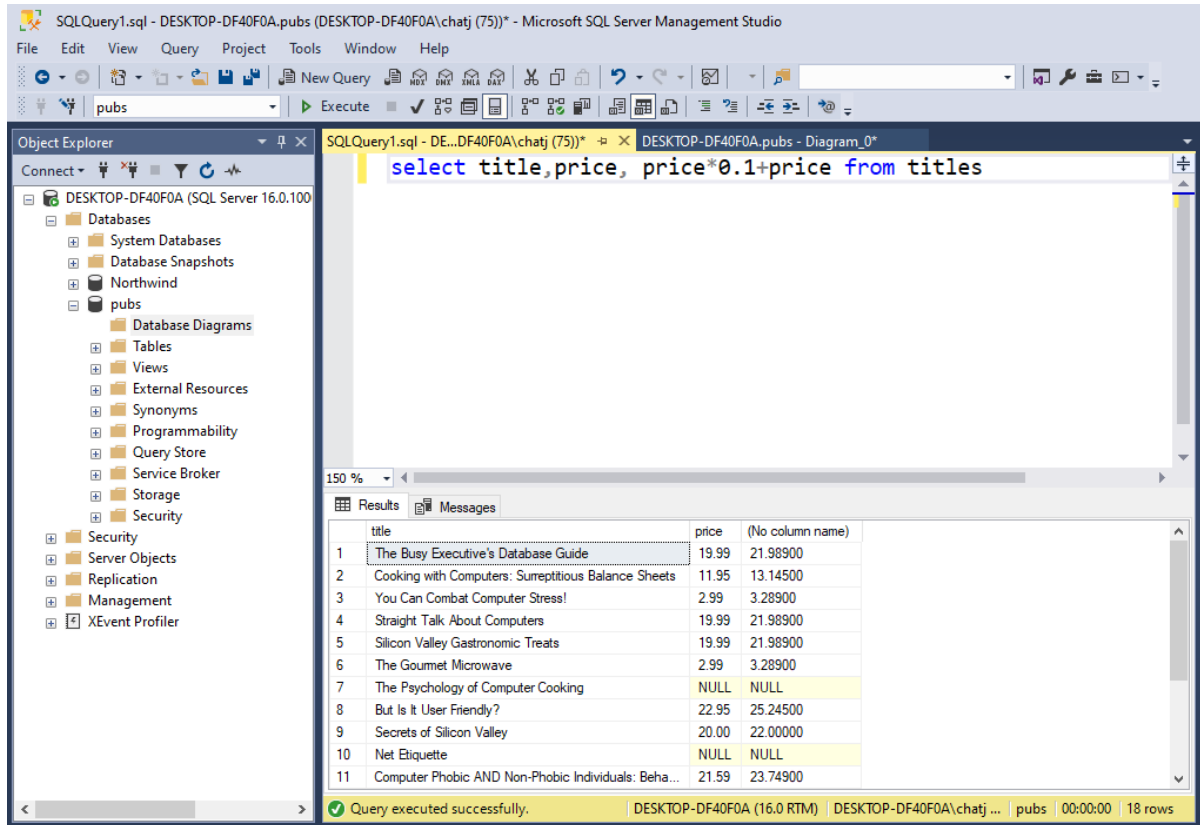
```
select title, price*0.1+price from titles
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show 11 rows of data from the 'titles' table in the 'pubs' database, with an additional calculated column for the price after a 10% increase.

	title	(No column name)
1	The Busy Executive's Database Guide	21.98900
2	Cooking with Computers: Surreptitious Balance Sheets	13.14500
3	You Can Combat Computer Stress!	3.28900
4	Straight Talk About Computers	21.98900
5	Silicon Valley Gastronomic Treats	21.98900
6	The Gourmet Microwave	3.28900
7	The Psychology of Computer Cooking	NULL
8	But Is It User Friendly?	25.24500
9	Secrets of Silicon Valley	22.00000
10	Net Etiquette	NULL
11	Computer Phobic AND Non-Phobic Individuals: Beha...	23.74900

The status bar at the bottom indicates: Query executed successfully. DESKTOP-DF40F0A (16.0 RTM) DESKTOP-DF40F0A\chatj ... pubs | 00:00:00 | 18 rows

أو لو أردنا عرض الكتاب مع سعره قبل وبعد.... :



- ولكن نلاحظ أن هذا العمود الجديد الذي تم فيه حساب السعر بعد الزيادة ليس لديه اسم (No Column Name) ويطلق على هذا ال Column اسم Calculated Column. ونطلق عليه Calculated لأنه لم يتم عرض ال Data كما هي وإنما قام بإحداث تغييرات عليها.
- ونعيد ونؤكد بأن ال Table الأساسي لا يتأثر بهذه العملية إنما ال Viewer فقط هو الذي يتأثر.

### ولكن كيف يمكننا إضافة اسم لهذا ال Column؟!

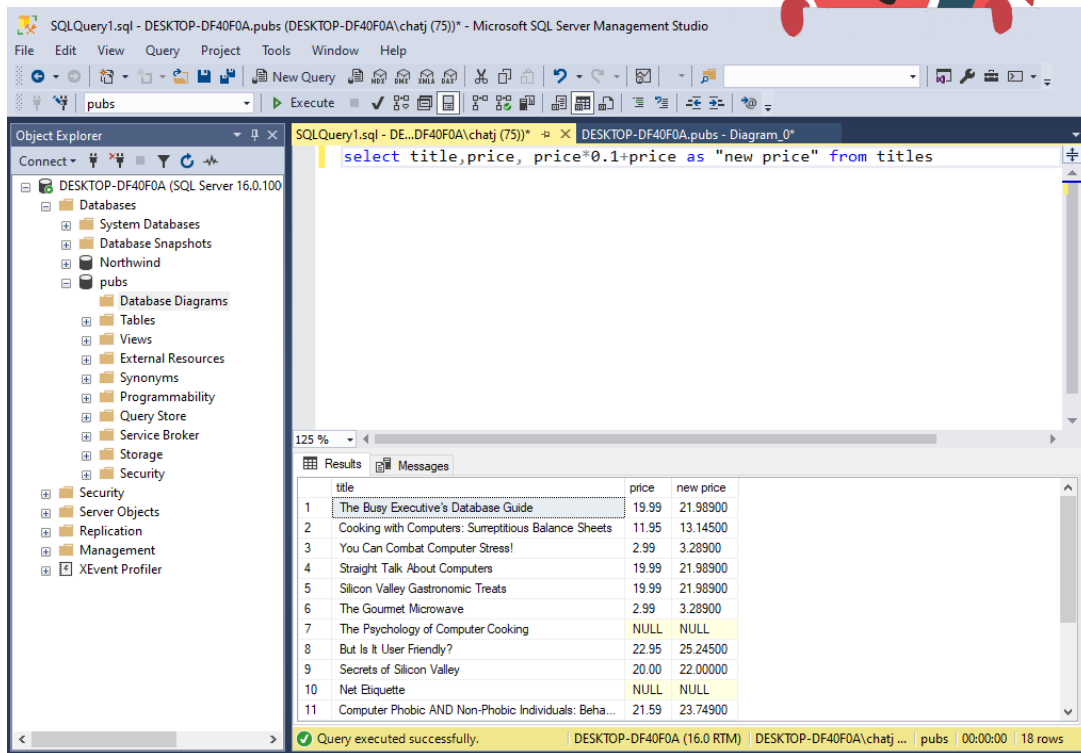
من تعليمة ال SELECT نفسها. إذ أننا نقوم بتطبيق ما يعرف ب alias وهي باختصار تعني إضافة اسم ل Column سيظهر في نتيجة ال Query. نقوم بتطبيقها على مثالنا مباشرة على الشكل التالي:

`SELECT title, price, price*0.1 + price as "new price" From titles`

وهذا ما يدعى ب alias, أي أننا ومن خلال استخدام هذه التعليمة استطعنا تحديد اسم لل Calculated Column الناتج

"Our wounds are often the openings into the best and most beautiful part of us."

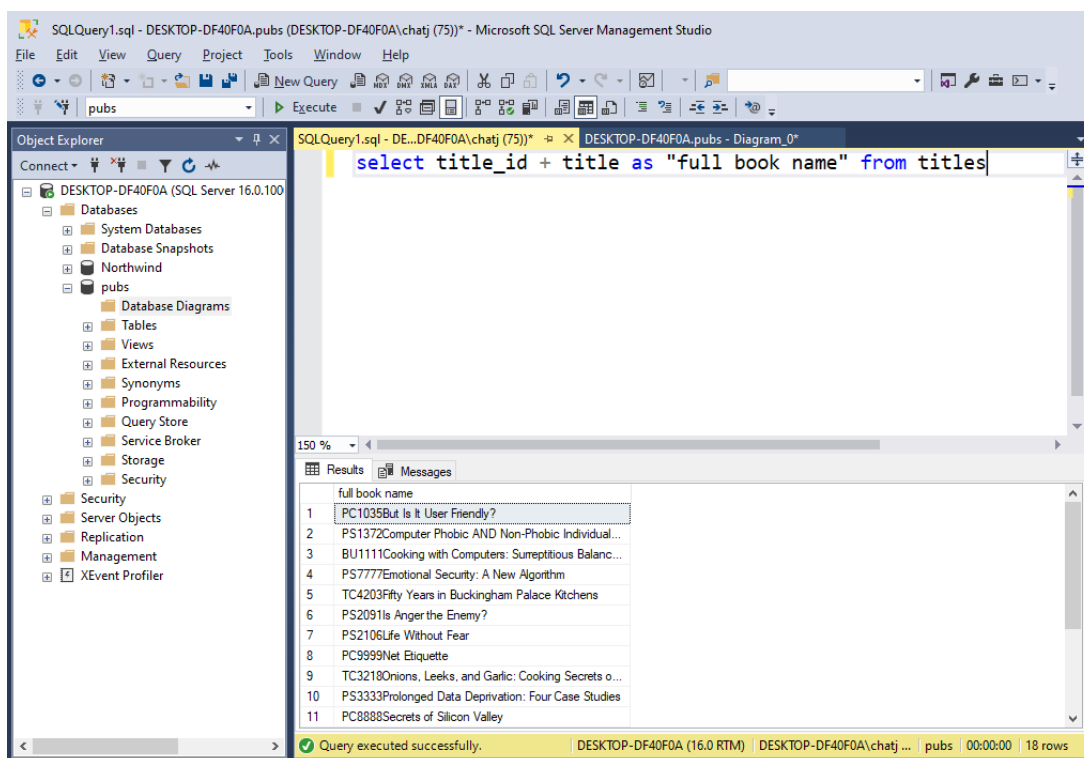




- وبالتأكيد يمكننا في حقل الـ Viewer تعديل أسماء الـ Columns بشكل عام سواء الـ Calculated أو غيرها.
- نستطيع أيضاً من خلال الـ SELECT دمج 2 Columns في Column واحد وذلك باستخدام ( + ). على سبيل المثال:

`SELECT title_id + title as "full book name" From titles`

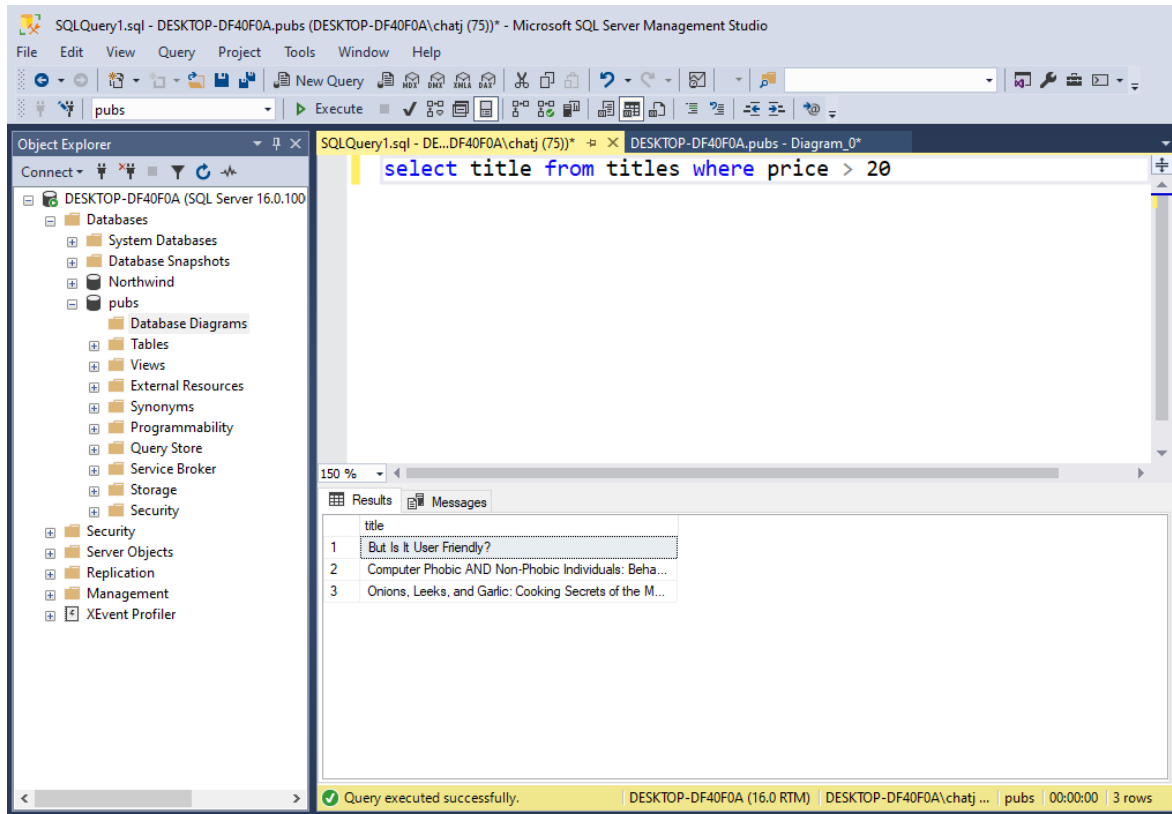
أي هنا ستصبح قيمة title\_id و title موجودة بنفس الـ Field وكأنهما Column واحد.



■ بالنسبة للقسم الثالث من تعليمة SELECT التي تحدثنا عنها والتي هي Criteria والتي تتمثل في تعليمة WHERE. فنحن من خلالها نقوم بوضع شروط معينة على ال Values لكي نقوم بعمل Customize وفترة لل Records الموجودة لدينا. مثال:

```
SELECT title From titles
WHERE price > 20
```

وعندها سيقوم فقط بعرض ال Records التي قيمة price attribute فيها أكثر من 20.



■ خلاصة: ضمن ال SELECT Statement لا يكون هنالك أي تعديل على ال Database وإنما التعديل خاص بالتعليمات INSERT, UPDATE, DELETE والتي سنتناولها في المحاضرات القادمة.

[روابط للتحميل:](#)

Northwind and Pubs Databases



SQL Server Express



SQL Server Management System



نهاية المحاضرة