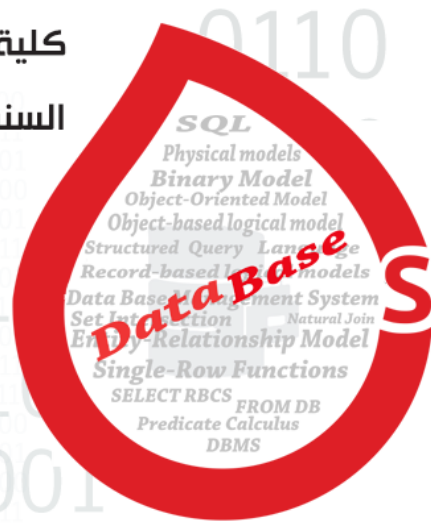


DDL Level Operations

عملي مشترك

محتوى مجاني غير مخصص للبيع التجاري



قواعد معطيات 1

30/05/2023

RB Informatics;

تحدثنا في المحاضرة السابقة عن ال Sub Queries وأنواعها وتحدثنا أيضاً عن ال Join وأنواعها، وبشكل عام في المحاضرات السابقة تحدثنا عن تعليمة Select فقط والتي تنتمي لمستوى ال DML. وفي هذه المحاضرة سنتحدث عن التعليمات الأخرى في مستوى ال DML وسنتحدث أيضاً عن بعض التعليمات في مستوى ال DDL.

هيا لنبدأ...♥

DDL – Data Definition Language

- سنتحدث بدايةً عن التعليمات الهامة في مستوى ال DDL، إن أول وأهم command في DDL هي <<CREATE>> وهي التي تقوم ببناء ال Objects في قاعدة المعطيات.
- فعلى سبيل المثال بعد أن قمنا بتصميم ERD لمشروع ما، نريد تحويل ال Entities الخاصة به إلى Tables، وهنا يجب استخدام التعليمة Create table.
- وعندما نريد عمل Create table فإن هنالك مجموعة قيود Constraints من المهم جداً معرفتهم وفهم استخدامهم وهم:

- Not Null: وتعني أن قيمة هذا الحقل إجبارية ولا يمكن أن يأخذ قيمة Null.
- Unique: وتعني أن قيمة هذا الحقل وحيدة ولا تتكرر لدى Record آخر.
- Primary Key: وتعني أن قيمة هذا الحقل يجب ان تكون وحيدة وهي إجبارية في نفس الوقت.
- Foreign Key: وتعني أن قيمة هذا الحقل يتبع لقيمة موجودة في Primary Key.
- Check: تقوم بعمل Validation على مستوى ال Row أو ال Record.
- Default: يتم من خلالها إسناد قيمة افتراضية لحقل عندما لا يتم تحديد قيمته.
- ملاحظة: إن تطبيق هذه القيود على ال Database ليس بالضرورة أن تكون على كل حقل على حدى، فعلى سبيل المثال يمكننا جعل عدة حقول (مثل ال id والاسم والعمر) Unique في آنٍ معاً.

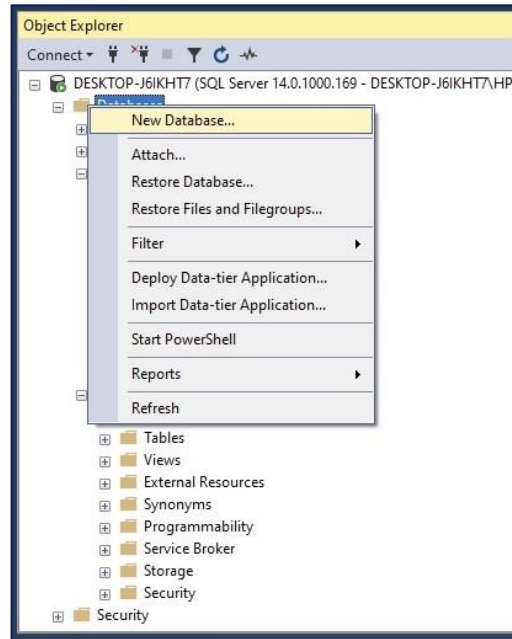


“When you change your thoughts, remember to also change your world.”



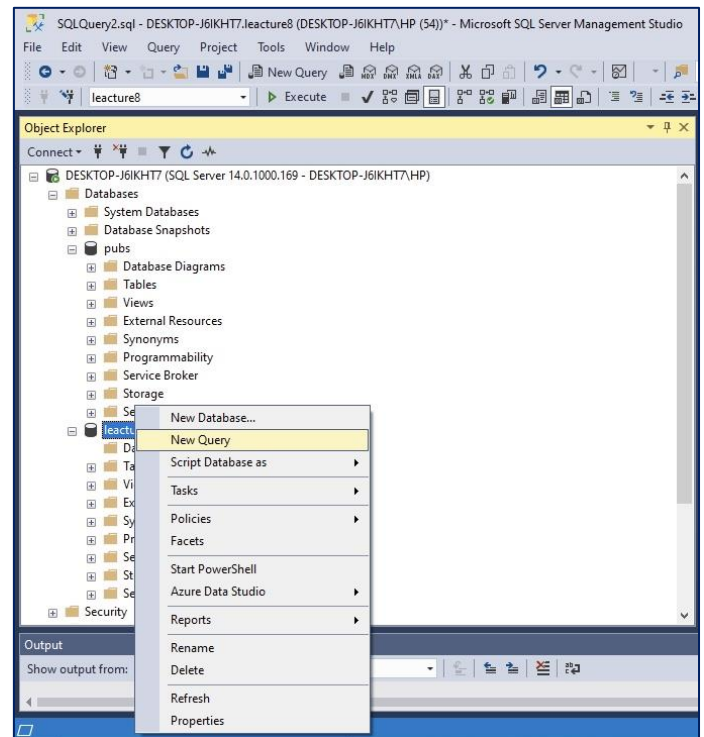
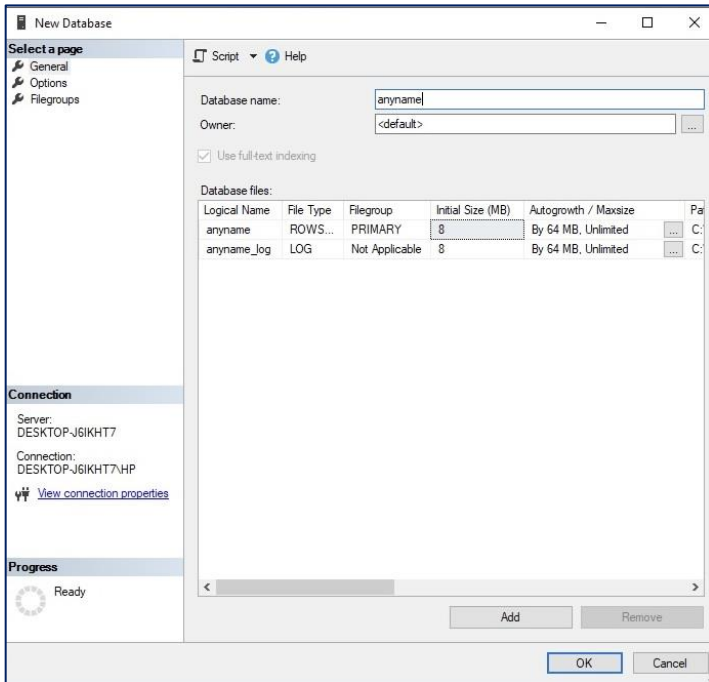
لنذهب للتطبيق العملي:

- سنقوم بفتح SQL Server ونقوم بعمل New Database... أي أننا نريد إنشاء Database خاصة مختلفة عن pubs التي نعمل عليها:



- نقوم بإعطاء اسم لهذه ال Database:

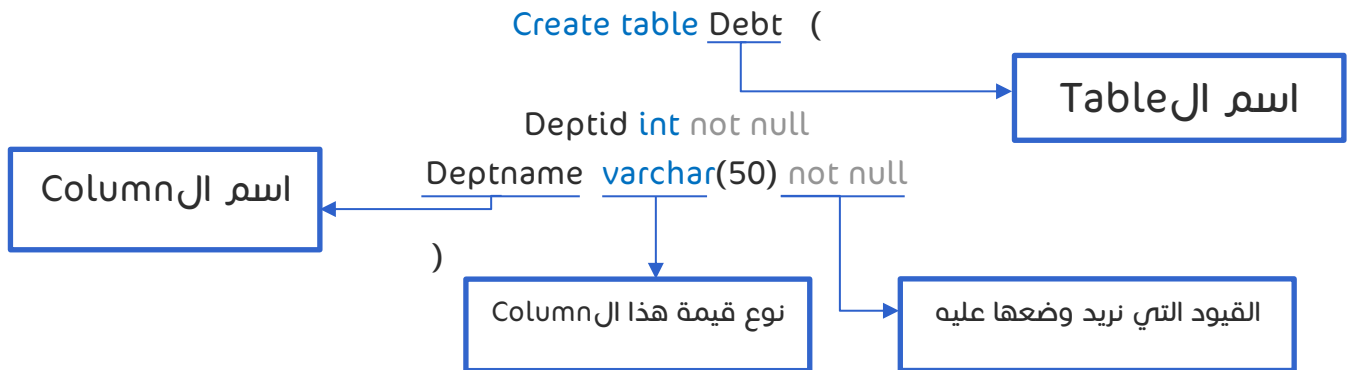
- الآن على هذه ال Database التي أنشأناها نقوم بعمل New Query:



"It is better to fail in originality than to succeed in imitation."

- والآن يمكننا كتابة التعليمات وال Queries التي نريد.

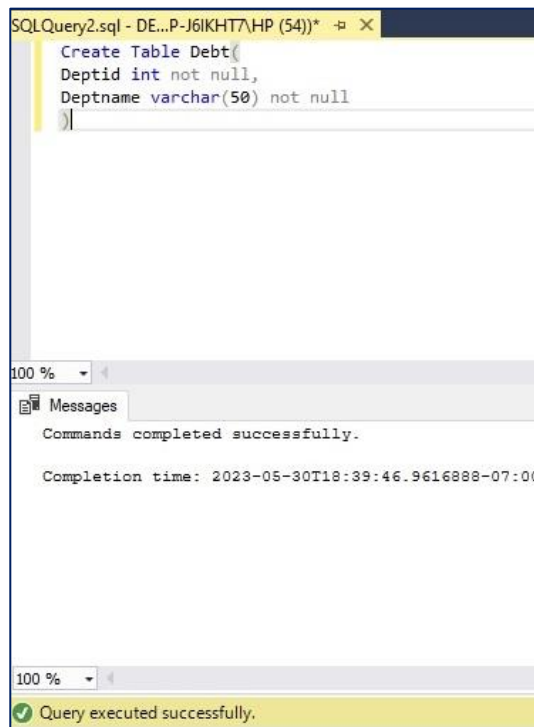
■ سنقوم الآن بإنشاء أول Table في ال Database الجديدة:



■ **ملاحظة:** في المثال السابق وعند إنشاء ال Column الثاني في ال Table قمنا بتحديد قيمته باستخدام `varchar()` بدلاً من `char()`, والفرق بينهما أن ال `char` تقوم بحجز حجم حسب القيمة التي نعطيها ولكن هذا الحجم يكون ثابت على كل ال Rows في ال Table (أي سواء استخدمناه كاملاً أم لا فإنه قد تم حجزه), بينما ال `varchar` تقوم بالحجز على قدر القيمة التي ندخلها لكل record ولكن بالتأكيد هذا الحجم يكون لحد معين وهو القيمة التي نحددها.

■ فعندما استخدمنا في المثال السابق `varchar(50)` فهذا يعني أن ال maximum length لل name هو 50 ولكن لو أدخلنا اسم مكون من 20 حرف مثلاً فيستم حجز 20 حرف فقط لا غير.

■ ونلاحظ بالتأكيد ظهور النتيجة التي تدل على نجاح العملية السابقة:



```

SQLQuery2.sql - DE...P-J6IKHT7\HP (54))* - X
Create Table Debt(
Deptid int not null,
Deptname varchar(50) not null
)

```

Messages

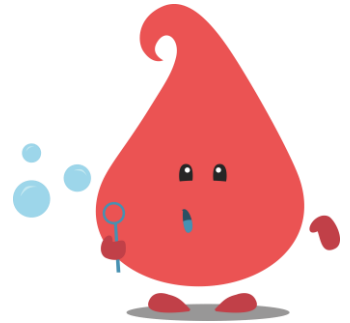
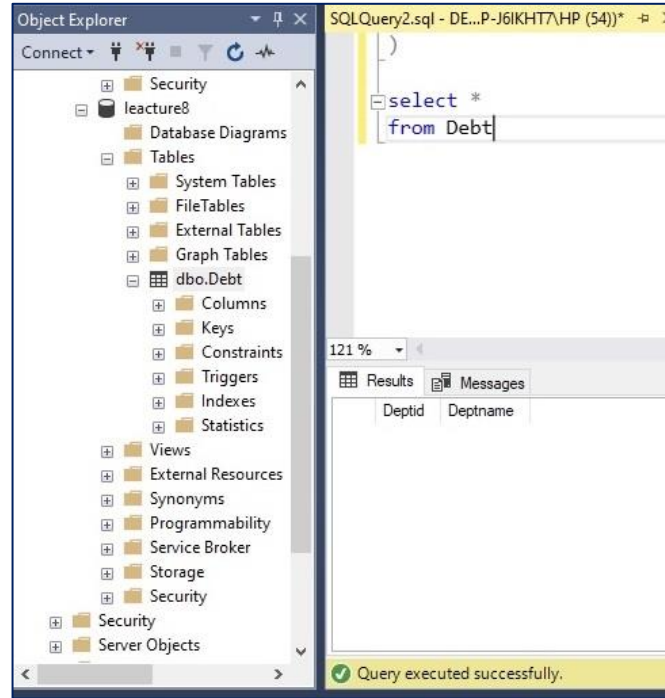
Commands completed successfully.

Completion time: 2023-05-30T18:39:46.9616888-07:00

Query executed successfully.

■ وللتأكد يمكننا التحقق من وجود هذه ال Table ضمن ال Database التي قمنا بإنشائها... وبالتأكيد لو حاولنا عمل Select وجلب المعلومات من هذه ال Table ستكون النتيجة فارغة:

“Don't let yesterday take up too much of today.”



ملاحظة: عند الانتهاء من إنشاء ال Table أي Create Table فلو أردنا بعد ذلك التعديل على بنية هذه ال Table عندها يتوجب علينا استخدام التعليمة الثانية من DDL وهي **ALTER**. أيضاً بالنسبة للقيود على الحقول أو ال Constraints فيمكننا وضعها عند إنشاء ال Table من البداية, أو يمكننا إضافتها فيما بعد وذلك أيضاً باستخدام **ALTER**.

سننفذ الآن بعض التعليمات باستخدام ALTER وسنذكر الغرض من كل منها:

مثال 1:

ALTER Table Debt Add Constraint PK_Dept primary key(Deptid);

من خلال التعليمة السابقة قمنا بالتعديل على ال table Debt حيث أضفنا Constraint اسمه PK_Dept نوعه primary key وذلك على ال Deptid Column.

مثال 2:

ALTER Table Debt Add Constraint Un_Dept Unique(Deptname);

من خلال التعليمة السابقة قمنا بالتعديل على ال Table Debt حيث أضفنا Constraint اسمه Un_Dept نوعه Unique وذلك على ال Deptname Column.

مثال 3:

ALTER Table Debt Add Col_b int Not Null;

من خلال التعليمة السابقة قمنا بالتعديل على ال Table Debt حيث أضفنا Column جديد اسمه Col_b وال Value الخاصة به هي int وأضفنا إليه Constraint هي Not Null.

مثال 4:

ALTER Table Debt Add Constraint ch_Dept check (Col_b > 0);

من خلال التعليمة السابقة قمنا بالتعديل على ال Table Debt حيث أضفنا Constraint اسمه ch_Dept يتحقق فيما إذا كانت قيم ال Col_b هي أكبر من الصفر.

“Either you run the day or the day runs you.”



```
SQLQuery3.sql - DE...P-J6IKHT7\HP (54)) * X DESKTOP-J6IKHT7\lecture8 - Diagram_0"
ALTER TABLE Debt
ADD CONSTRAINT PK_Debt PRIMARY KEY (Deptid);
ALTER TABLE Debt
ADD CONSTRAINT Un_Debt Unique (Deptname);
ALTER TABLE Debt ADD col_b int NOT NULL;
ALTER TABLE Debt
ADD CONSTRAINT Ch_Debt CHECK (col_b > 0);
```

121 %

Messages

Commands completed successfully.

Completion time: 2023-05-30T20:28:29.9596619-07:00

121 %

Query executed successfully.

ملاحظة: عند إضافة أي من ال Constraints على ال Table فإنه تلقائياً يقوم بعمل Check لل Data ضمن هذا ال Table. على سبيل المثال إذا أضفنا Unique Constraint على Column وكان هذا ال Column هو Unique أساساً عند ذلك تعيد التعليمه خطأ في النتيجة. أما إذا تم إضافة constraint بعد تواجد data في هذه ال table التي نضيف ال constraint عليها أي بعد إضافة records, عند ذلك إذا كانت قيم هذه ال records بالنسبة لهذا ال column موافقة لشرط ال constraint فسيتم تنفيذه بشكل صحيح, أما إذا كانت مخالفة له عند ذلك يحدث error.

على سبيل المثال في table معين هناك price column ويوجد عدة records لها prices مختلفة.. وأضفنا على ال price column شرط أي constraint معين وهو (price < 200) Check, عندها لو كانت كل ال prices في هذه ال table أقل من 200 فسيتم تنفيذها بينما لو كان يوجد بالفعل record له price أكبر من 200 فلن يتم تنفيذها.

سنحدث الآن عن تعليمة INSERT:

عندما نريد إضافة record جديد إلى Table معينة يجب أولاً تحديد هذه ال Table ثم بعد ذلك يجب إدخال ال Values لكل ال Columns الموجودين في هذه ال table.

مثال:

Insert Into Debt Values (1, 'sales', 1)

هنا حددنا اسم ال Table التي نريد الإضافة فيها

وهنا ندخل ال Values الموافقة لل Columns الموجودة في هذه ال table والتي هي في مثالنا Deptid, Deptname, Col_b



```
SQLQuery3.sql - DE...P-J6IKHT7\HP (54)) * X DESKTOP-J6IKHT7\lecture8 - Diagram_0"
INSERT INTO Debt VALUES (1, 'Sales', 1)

select * from Debt
```

121 %

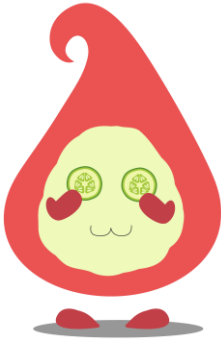
Results

Deptid	Deptname	col_b
1	Sales	1

Query executed successfully.

ولو قمنا بعد تنفيذ هذه التعليمة بعمل Select من ال Debt table ستظهر لنا النتيجة:

- **ملاحظة 1:** لو كان ال id على سبيل المثال هو identity أي أنه يأخذ قيم بنفسه كل مرة..... عند ذلك ليس من الضروري إدخال قيم له عند عمل Insert.
- **ملاحظة 2:** لو حاولنا إعادة تنفيذ التعليمة السابقة نفسها مرة أخرى فإنها ستفشل لأنها أولاً لم تحقق ال primary key على مستوى ال Debtid ولم تحقق ال Unique على مستوى ال Debnam.



```
SQLQuery3.sql - DE...P-J6IKHT7\HP (54)) * x DESKTOP-J6IKHT7\lecture8 - Diagram_0"
INSERT INTO Debt VALUES (1, 'Sales', 1)

121 %
Messages
(1 row affected)
Completion time: 2023-05-30T20:30:23.7400580-07:00

121 %
Query executed successfully.
```



- وبنفس الطريقة لو أضفنا record وأعطينا ال Col_b الخاص به قيمة 1- (على سبيل المثال) أي أننا قمنا بعمل خلل في الشرط:

```
SQLQuery3.sql - DE...P-J6IKHT7\HP (54)) * x DESKTOP-J6IKHT7\lecture8 - Diagram_0"
INSERT INTO Debt VALUES (2, 'managment', -1)

121 %
Messages
Msg 547, Level 16, State 0, Line 15
The INSERT statement conflicted with the CHECK constraint "Ch_Debt". The conflict occurred in database "lecture8", table "dbo.Debt",
The statement has been terminated.
Completion time: 2023-05-30T20:53:10.8296665-07:00

121 %
Query completed with errors.
```

```
SQLQuery3.sql - DE...P-J6IKHT7\HP (54)) * x DESKTOP-J6IKHT7\lecture8 - Diagram_0"
Create Table Emp(
Empid int not null,
Empname varchar(50) not null,
Deptid int);

121 %
Messages
Commands completed successfully.
Completion time: 2023-05-30T20:59:40.6774025-07:00

121 %
Query executed successfully.
```

- سنقوم الآن ببناء Table أخرى وهي Emp بنفس الطريقة:

```
Create Table Emp(
Empid int not null,
Empname varchar(50) not null,
Deptid int )
```

والآن نقوم بعمل Constraint على Deptid في ال Emp table وجعله Foreign Key:



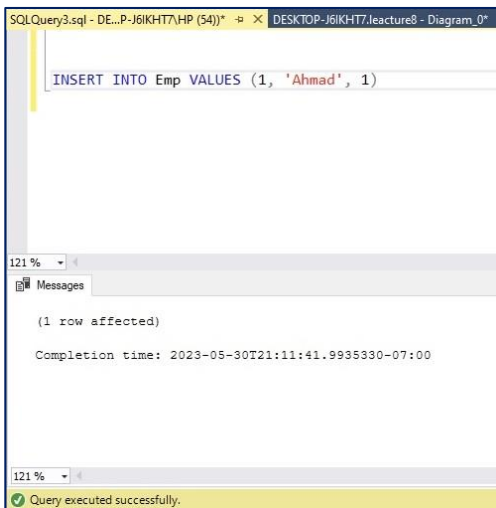
ALTER Table Emp

Add Constraint FK_Emp Foreign key(Deptid)

References Debt(Deptid);

لاحظ أنه عند وضع Foreign Key فلا بد أن نذكر المرجع أو المصدر الذي سيأخذ قيمة منه وفي مثالنا هو Deptid في ال Debt table.

نقوم الآن على سبيل المثال بعمل insert إلى ال Emp Table:

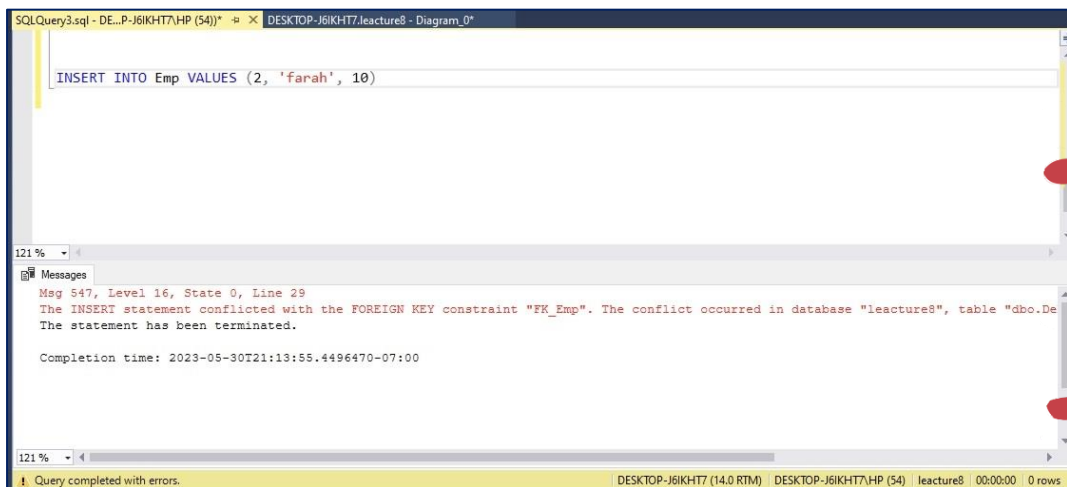


Insert Into Emp Values (1, 'Ahmad', 1);

ونلاحظ نجاح العملية.

لو قمنا أيضاً بعمل العملية التالية:

Insert Into Emp Values (2, 'Farah', 10);

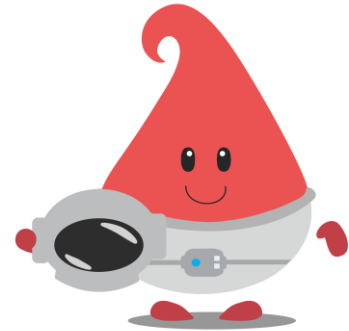
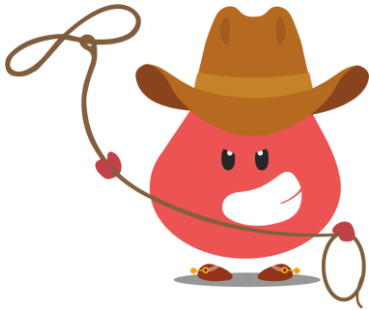
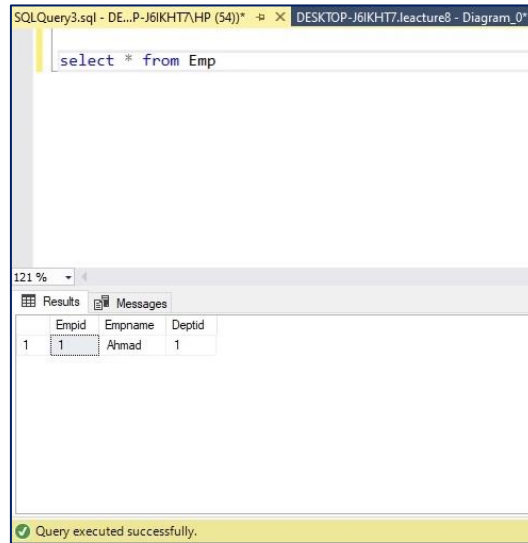


سنلاحظ فشل العملية وذلك لأن ال Value الثالثة التي أضفناها والتي من المفترض أن تكون Foreign Key تابعة ل Deptid في ال Debt table هي غير صحيحة. فلا يوجد في ال Debt table أي Deptid بالقيمة 10.

سنتحدث الآن عن تعليمة UPDATE:

Select * from Emp

لو قمنا بدايةً بعرض الـ Data الموجودة في Emp table:



الآن لو أردنا على سبيل المثال التعديل على اسم هذا الموظف:

Update Emp Set Empname = 'Samer' Where Empid = 1

الـ Table المراد تعديلها

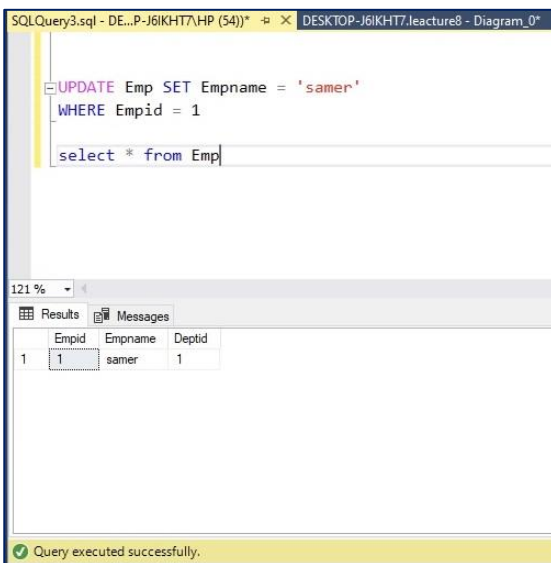
الـ Column المراد
تعديله

الـ Value الجديدة

مكان التعديل أو
الـ Records التي
سنقوم بتعديلها

ملاحظة: لو لم نضع Where في تعليمة الـ Update فسيقوم بتنفيذ التعديل على كل الـ Records.

الآن لو قمنا بعرض الـ table مرة أخرى للتحقق:



ملاحظة: يفضل دائماً في الـ Update والـ Delete وعند تحديد الـ Criteria أن يكون الشرط على primary key لكي لا يقوم بعمل scan لكل الـ table.

"Success is not final; failure is not fatal: It is the courage to continue that counts."

سنحدث الآن عن تعليمة DELETE:

■ الآن لو قمنا بعرض Emp table بعد الحذف سنجد أنها فارغة:

DELETE Emp Where Empid = 1

تصحيح أخطاء المحاضرات السابقة:

المحاضرة الرابعة (الصفحة 2):

الخطأ: ولكن نحن في الحالة العامة وفي أغلب الأحيان سنقوم بالعمل ب **SQL Server Authentication**.

الصواب: ولكن نحن في الحالة العامة وفي أغلب الأحيان سنقوم بالعمل ب **Windows Authentication**.

المحاضرة الرابعة (الصفحة 8):

يتم استبدال الصورة القديمة بهذه الصورة الجديدة:

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

نهاية المقرر

إلى هنا نصل إلى ختام القسم العملي من مادة قواعد المعطيات 1...

لقاؤنا يتجدد مع مادة قواعد المعطيات 2 في السنة الرابعة
سامحونا إن أخطأنا فجلّ من لا يخطئ ولا تنسونا من صالح دعائكم



كل الحب من فريق مادة قواعد المعطيات 1

#we_carry_your_O2

#Viva_RBCs

#Database1