الأكاديمية العربية الدولية
Arab International Academy

استوديو إدارة خادم SQL

SQL Server Management Studio (SSMS)

م.عبدالقادر العبدالله

كلية العلوم – تخصص البرمجة

المحاور

- مقدمة عن خادم SQL
- شرح و تحميل استوديو إدارة خادم SQL
- لغة SQL
- لغة تعريف البيانات (Data Definition Language - DDL)
- انواع البيانات (Data Types)
- لغة معالجة البيانات (Data Manipulation Language - DML)
- أمثلة شاملة و أوامر عامة

المخرجات المتوقعة من الدرس

- القدرة على استخدام خادم SQL في إدارة قواعد البيانات
- فتح SSMS وإنشاء اتصال مع SQL Server
- تصنيف أوامر SQL (DDL, DML, DCL, TCL)
- القدرة على إنشاء وحذف وتعديل الجداول
- معرفة الأنواع المختلفة للبيانات و القدرة على اختيار النوع المناسب لكل عمود
- القدرة على إدخال واسترجاع البيانات
- تعديل البيانات وحذفها و كتابة استعلامات متكاملة

مقدمة عن خادم SQL

- **خادم SQL (SQL Server) :** هو نظام إدارة قواعد بيانات علائقية تم تطويره بواسطة شركة مايكروسوفت يقوم بتخزين البيانات في شكل جداول وعلاقات و تنفيذ العمليات على البيانات مثل الإدخال, التحديث , الاستعلام و والحذف ويتكون من :
  - محرك قاعدة البيانات (Database Engine) : المسؤول عن إدارة البيانات
  - خدمات التكامل (Integration Services) : لتحويل البيانات بين الأنظمة
  - خدمات التحليل (Analysis Services) : لتحليل البيانات
  - خدمات التقارير (Reporting Services) : لإنشاء تقارير تفاعلية
  - وكيل الخادم (SQL Server Agent) : لتنفيذ المهام التلقائية

مقدمة عن خادم SQL

- **استوديو إدارة خادم SQL (SSMS) :** هو واجهة رسومية تستخدم للتفاعل مع خادم SQL يتيح استعراض قواعد البيانات والجداول والعلاقات بين البيانات و تمكين كتابة استعلامات SQL وتشغيلها ومن خصائصه :
  - إمكانية إدارة قواعد البيانات والخوادم والمستخدمين بشكل كامل
  - أدوات مراقبة وتحليل الأداء
  - يتيح إمكانية جدولة وتنفيذ المهام التلقائية
  - نسخ احتياطي واستعادة البيانات
  - يدعم إدارة قواعد بيانات كبيرة ومعقدة

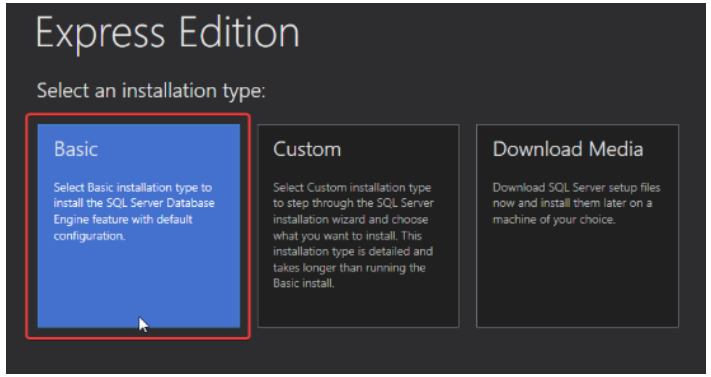
مقدمة عن خادم SQL

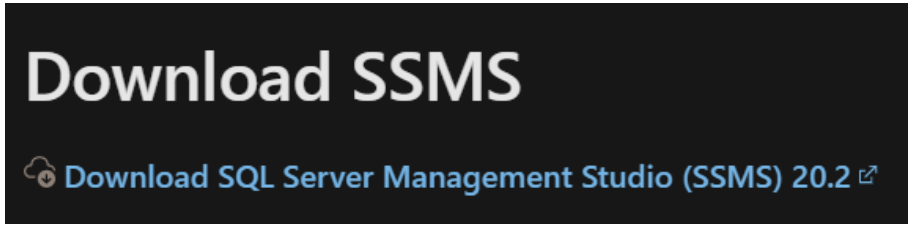
من حيث	SQL Server	SSMS
التعريف	نظام إدارة قواعد بيانات علائقية	واجهة رسومية لإدارة خادم SQL
الاستعمال	إدارة البيانات وتخزينها وتنفيذ الأوامر	توفير وسيلة سهلة للتفاعل مع الخادم
التفاعل مع البيانات	يتم عبر استعلامات SQL مباشرة	يتم عبر كتابة استعلامات أو استخدام واجهة مرئية
المهام الإدارية	ينفذ المهام الإدارية داخليا كخدمة خادم	يستخدم لإدارة SQL Server بشكل مرئي او برمجي
التثبيت	يتم تثبيته كخدمة على الخادم أو الكمبيوتر الشخصي	يثبت كبرنامج منفصل ويستخدم للاتصال ب SQL Server

شرح و تحميل استوديو إدارة خادم SQL

1) تحميل سيرفر SQL : (https://www.microsoft.com/en-us/sql-server/sql-server-downloads)

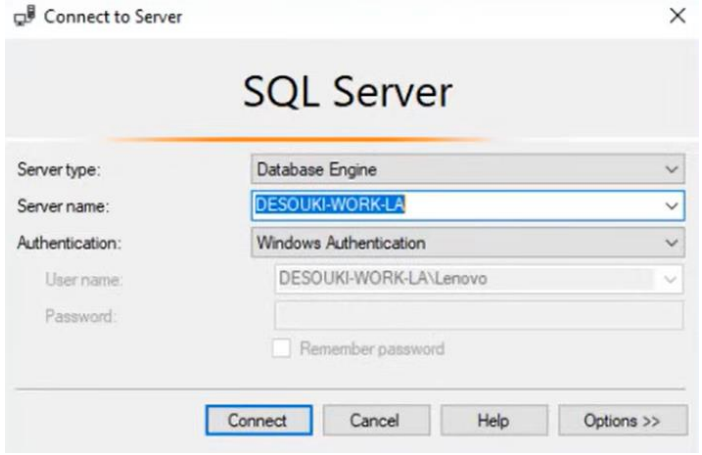
2) تحميل استوديو إدارة خادم SQL : (https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16)





شرح و تحميل استوديو إدارة خادم SQL

- **نوع الخادم (Server type) :** هذا الحقل يحدد نوع الخادم الذي ترغب بالاتصال به يحتوي على محرك قاعدة البيانات , خدمات التكامل , خدمات التحليل و خدمات التقارير .
- **اسم الخادم (Server Name) :** يمثل اسم الخادم الذي ترغب في الاتصال به يمكن أن يكون اسم جهاز الكمبيوتر المحلي أو اسم خادم بعيد .
- **طريقة المصادقة (Authentication) :** تحدد الطريقة التي يتم بها تسجيل الدخول إلى الخادم



- مصادقة النظام (Windows Authentication) :

يستخدم بيانات تسجيل الدخول إلى نظام التشغيل للدخول إلى الخادم

- مصادقة الخادم (SQL Server Authentication) :

يتطلب إدخال اسم مستخدم وكلمة مرور خاصة في الخادم

شرح و تحميل استوديو إدارة خادم SQL

❖ **قواعد بيانات النظام (system databases) :** قواعد بيانات يتم إنشاؤها تلقائيا عند تثبيت الخادم وهي جزء أساسي من النظام لكل قاعدة بيانات وظيفتها المحددة في إدارة وتشغيل خادم SQL وهي :

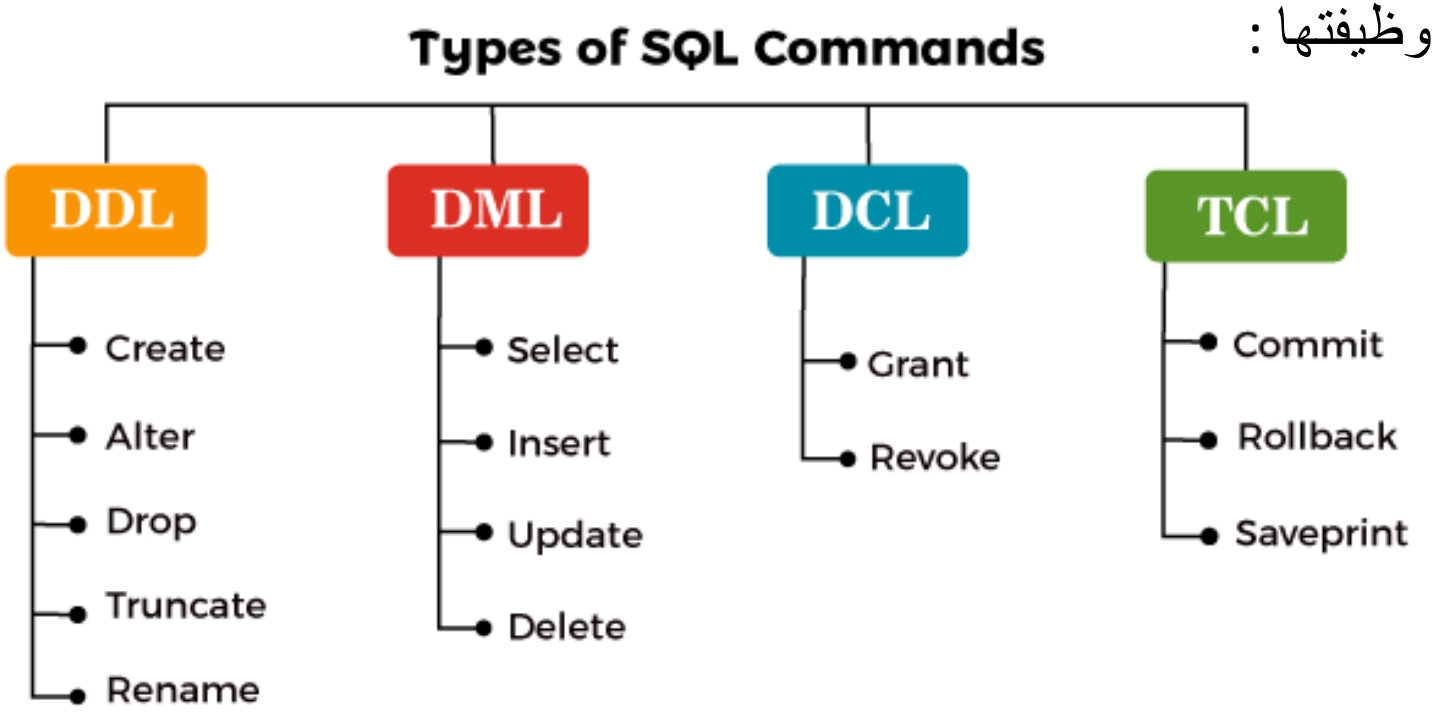
- master : تحتوي على معلومات عن الخادم بأكمله بما في ذلك جميع قواعد البيانات الأخرى حسابات تسجيل الدخول، إعدادات الخادم وعلاقات النظام
- model : تستخدم كنموذج لإنشاء قواعد بيانات جديدة
- msdb : تستخدم لتخزين البيانات المتعلقة بالمهام المجدولة والوظائف التلقائية التي يتم تنفيذها بواسطة SQL Server Agent
- tempdb : قاعدة بيانات مؤقتة تستخدم لتخزين البيانات المؤقتة أو العمليات المؤقتة التي يتم إنشاؤها أثناء تشغيل الخادم

شرح و تحميل استوديو إدارة خادم SQL

قاعدة البيانات	نوع البيانات المخزنة	الأهمية
master	أماكن تخزين ,أسماء قواعد البيانات قواعد البيانات وإعدادات الخادم	إذا تعطلت قد يتوقف الخادم عن العمل تعد العمود الفقري للخادم
model	إعدادات وقوالب افتراضية يتم نسخها إلى أي قاعدة بيانات جديدة يتم إنشاؤها	توفر طريقة لتخصيص قواعد البيانات الجديدة بشكل تلقائي
msdb	بيانات المهام المجدولة و معلومات النسخ الاحتياطي	ضرورية لأتمتة العمليات
tempdb	الجداول والفهارس المؤقتة والبيانات الناتجة عن استعلامات مؤقتة	تحسين أداء النظام أثناء تنفيذ الاستعلامات

لغة SQL

❖ **SQL (Structured Query Language) :** هي لغة قياسية تستخدم للتعامل مع قواعد البيانات تقوم بتنفيذ مجموعة متنوعة من العمليات مثل إنشاء الجداول، إضافة البيانات، تعديلها، حذفها، وإدارة الأذونات يمكن تصنيف أوامر SQL إلى عدة مجموعات رئيسية بناء على وظيفتها :



لغة SQL

نوع الأوامر	الوصف	الأوامر الشائعة	الاستخدامات
لغة تعريف البيانات (DDL)	تستخدم لإدارة هيكل قاعدة البيانات	تعديل , إفراغ الجدول ,إنشاء وحذف	إنشاء الجداول والفهارس والعلاقات تعديل أو حذف بنية الجدول
لغة معالجة البيانات (DML)	تستخدم للتعامل مع البيانات داخل الجداول	استرجاع , حذف, إدخال وتعديل	إدخال بيانات جديدة وحذف البيانات وتحديث بيانات موجودة
لغة التحكم في البيانات (DCL)	تستخدم لإدارة أذونات الوصول إلى البيانات والتحكم فيها	منح الأذونات وإلغاء الأذونات	منح أو إلغاء صلاحيات الوصول إلى البيانات أو الجداول للمستخدمين
لغة التحكم في المعاملات (TCL)	تستخدم لإدارة العمليات والتأكد من تنفيذها بالكامل أو إلغائها	حفظ التغييرات و التراجع عن التغييرات وتحديد نقطة حفظ	حفظ او التراجع عن التغييرات التي تمت أثناء المعاملات

لغة تعريف البيانات (Data Definition Language - DDL)

❖ **لغة تعريف البيانات :** تستخدم لتعريف وإدارة هيكل قاعدة البيانات وتمسح بإنشاء وتعديل وحذف الكائنات (مثل الجداول، الفهارس، والعلاقات) التي تشكل بنية قاعدة البيانات:

1) **إنشاء (CREATE) :** يستخدم لإنشاء كائن جديد في قاعدة البيانات، مثل جدول، فهرس، أو قاعدة بيانات

```
CREATE DATABASE databasename;
```

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

```
CREATE INDEX index_name
ON table_name (column1, column2, ...);
```

```
USE [your database]
GO
CREATE TABLE test_transaction (column_1 varchar(10))
GO
```

لغة تعريف البيانات (Data Definition Language - DDL)

2) تعديل (ALTER) : يستخدم لتعديل بنية الكائن الموجود بالفعل، مثل إضافة أعمدة جديدة، تعديل خصائص الأعمدة، أو حذف الأعمدة :

```
ALTER TABLE table_name
ADD column_name datatype;
```

```
ALTER TABLE table_name
MODIFY COLUMN column_name datatype;
```

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

لغة تعريف البيانات (Data Definition Language - DDL)

3) حذف (DROP) : يستخدم لحذف كائن موجود في قاعدة البيانات بالكامل (مثل جدول أو فهرس) او حذف قاعدة البيانات كاملة :

```
DROP DATABASE databasename;

DROP INDEX table_name.index_name;

DROP TABLE table_name;
```

لغة تعريف البيانات (Data Definition Language - DDL)

4) إفراغ (TRUNCATE) : يستخدم لإزالة جميع البيانات من الجدول، ولكن دون حذف الجدول نفسه :

```
TRUNCATE [TABLE] table_name;
```

```
TRUNCATE TABLE Categories;
```

لغة تعريف البيانات (Data Definition Language - DDL)

```
CREATE DATABASE الشركة;

USE الشركة;  -- لاختيار قاعدة البيانات التي سنعمل عليها

CREATE TABLE الموظفين (
    id INT PRIMARY KEY,          -- العمود الأول: رقم تعريف الموظف (أساسي)
    الاسم VARCHAR(50) NOT NULL,  -- العمود الثاني: اسم الموظف
    العمر INT,                   -- العمود الثالث: عمر الموظف
    الراتب DECIMAL(10, 2)        -- العمود الرابع: راتب الموظف
);


ALTER TABLE الموظفين ADD القسم VARCHAR(50);

ALTER TABLE الموظفين DROP COLUMN العمر;

TRUNCATE TABLE الموظفين;

DROP TABLE الموظفين;

DROP DATABASE الشركة;
```

❖ مثال : يتم انشاء قاعدة بيانات وانشاء جدول داخلها والتعديل عليه ثم حذف الجدول وقاعدة البيانات :

Results | Messages

	TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	PRECISION	LENGTH	SCALE	RADIX	NULLABLE	REMARKS	COLUMN_DEF	SQL_DATA_TYPE	SQL_DATETIME_SUB	CHAR_OCTET_LENGTH	ORDINAL_POSITION	IS_NULLABLE	SS_DATA_T
1	الشركة	dbo	الموظفين	id	4	int	10	4	0	10	0	NULL	NULL	4	NULL	NULL	1	NO	56
2	الشركة	dbo	الموظفين	الاسم	12	varchar	50	50	NULL	NULL	0	NULL	NULL	12	NULL	50	2	NO	39
3	الشركة	dbo	الموظفين	العمر	4	int	10	4	0	10	1	NULL	NULL	4	NULL	NULL	3	YES	38
4	الشركة	dbo	الموظفين	الراتب	3	decimal	10	12	2	10	1	NULL	NULL	3	NULL	NULL	4	YES	106
5	الشركة	dbo	الموظفين	القسم	12	varchar	50	50	NULL	NULL	1	NULL	NULL	12	NULL	50	5	YES	39

Results | Messages

	TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	PRECISION	LENGTH	SCALE	RADIX	NULLABLE	REMARKS	COLUMN_DEF	SQL_DATA_TYPE	SQL_DATETIME_SUB	CHAR_OCTET_LENGTH	ORDINAL_POSITION	IS_NULLABLE	SS_DATA_T

انواع البيانات (Data Types)

1) **البيانات الدقيقة (Exact Numerics) :** تستخدم لتمثيل القيم العددية التي يجب أن تكون دقيقة دون تقريب أو أخطاء ناتجة عن الكسور. هذه الأنواع مناسبة للأرقام الصحيحة والقيم المالية أو الأرقام ذات الدقة المحددة وتقسم الى :

- **أنواع البيانات الصحيحة (Integer Data Types) :** هي قيم عددية لا تحتوي على أي جزء كسري. دائما ما تكون هذه الأعداد مخزنة باستخدام مساحة تخزين ثابتة، حيث يحدد نوع البيانات عدد البايتات المستخدمة والنطاق الذي يمكن تمثيله

Data type	Range	Storage
bit	0 or 1	1 bit **
tinyint	0 to 255	1 byte
smallint	-2^15 (-32,768) to 2^15-1 (32,767)	2 bytes
int	-2^31 (-2,147,483,648) to 2^31-1 (2,147,483,647)	4 bytes
bigint	-2^63 (-9,223,372,036,854,775,808) to 2^63-1 (9,223,372,036,854,775,807)	8 bytes

انواع البيانات (Data Types)

- **الأنواع ذات الدقة والثبات (Fixed Precision and Scale Data Types) :** هذه الأنواع تستخدم لتخزين الأرقام التي تحتاج إلى دقة ثابتة (بدون تقريب) تعتبر هذه الأنواع ضرورية عند التعامل مع البيانات المالية أو الحسابات التي تتطلب دقة عالية

- numeric
- decimal
- smallmoney
- money

Data type	Range	Storage
Decimal [(p [, s])] or Numeric [(p [, s])]	-10^38 + 1 to 10^38 - 1	See **Precision** table

Precision Table

Precision	Storage bytes
1 - 9	5
10-19	9
20-28	13
29-38	17

Data type	Range	Storage
money	-922,337,203,685,477.5808 to 922,337,203,685,477.5807	8 bytes
smallmoney	-214,748.3648 to 214,748.3647	4 bytes

انواع البيانات (Data Types)

2) البيانات التقريبية (Approximate Numerics) : أنواع بيانات تستخدم لتخزين الأرقام العشرية يتم تمثيل هذه القيم بدقة تقريبية مما يعني أنها قد تؤدي إلى أخطاء بسيطة في التقريب :

- Float
- Real

Data type	Range	Size
float	-1.79E+308 to -2.23E-308, 0 and 2.23E-308 to 1.79E+308	depends on **n** in table below
real	-3.40E + 38 to -1.18E - 38, 0 and 1.18E - 38 to 3.40E + 38	4 Bytes

n value	Precision	Size
1-24	7 digits	4 bytes
25-53	15 digits	8 bytes

انواع البيانات (Data Types)

3) البيانات الزمنية والتاريخية (Date and Time) : تستخدم لتخزين معلومات التواريخ والأوقات هذه الأنواع تتيح تمثيل البيانات المرتبطة بالتواريخ والأوقات بدقة وهي مفيدة في إدارة الأحداث الزمنية و سجلات العمليات :

- Datetime : يستخدم لتخزين التاريخ والوقت معا بدقة تقارب الثانية
- Smalldatetime : يستخدم لتخزين التاريخ والوقت مع دقة أقل
- Date : يستخدم لتخزين التواريخ فقط بدون وقت
- Time : يستخدم لتخزين الوقت فقط بدون تاريخ
- Datetime2 : نسخة محسنة من Datetime تدعم نطاقا أوسع ودقة أعلى
- datetimeoffset : يستخدم لتخزين التاريخ والوقت مع فارق التوقيت بين المناطق

انواع البيانات (Data Types)

4) البيانات النصية (Character Strings) : تستخدم لتخزين النصوص أو السلاسل النصية :

- char : يستخدم لتخزين نصوص ذات طول ثابت إذا كانت النصوص أقصر من الطول المحدد يتم ملء الباقي بمسافات
- Varchar : يستخدم لتخزين نصوص ذات طول متغير يتم تخزين النصوص فقط بالحجم الفعلي لها دون إضافات
- Varchar(max) **:** يستخدم لتخزين نصوص طويلة جدا

انواع البيانات (Data Types)

5) البيانات النصية تدعم معيار Unicode (Unicode Character Strings) : تستخدم لتخزين النصوص التي تحتوي على أحرف متعددة اللغات مثل الأحرف العربية، الصينية، اليابانية، أو الرموز الخاصة :

- nchar : يستخدم لتخزين النصوص ذات الطول الثابت مع دعم معيارUnicode
- nvarchar : يستخدم لتخزين النصوص ذات الطول المتغير الثابت مع دعم معيارUnicode
- nvarchar(max) : يستخدم لتخزين النصوص الطويلة جدا مع دعم Unicode

انواع البيانات (Data Types)

6) البيانات الثنائية (Binary Strings) **:** تستخدم لتخزين البيانات بتنسيق ثنائي تعتبر هذه الأنواع مثالية لتخزين البيانات مثل الصور، الفيديوهات، المستندات، أو البيانات المشفرة التي يتم تمثيلها في هيئة بتات :

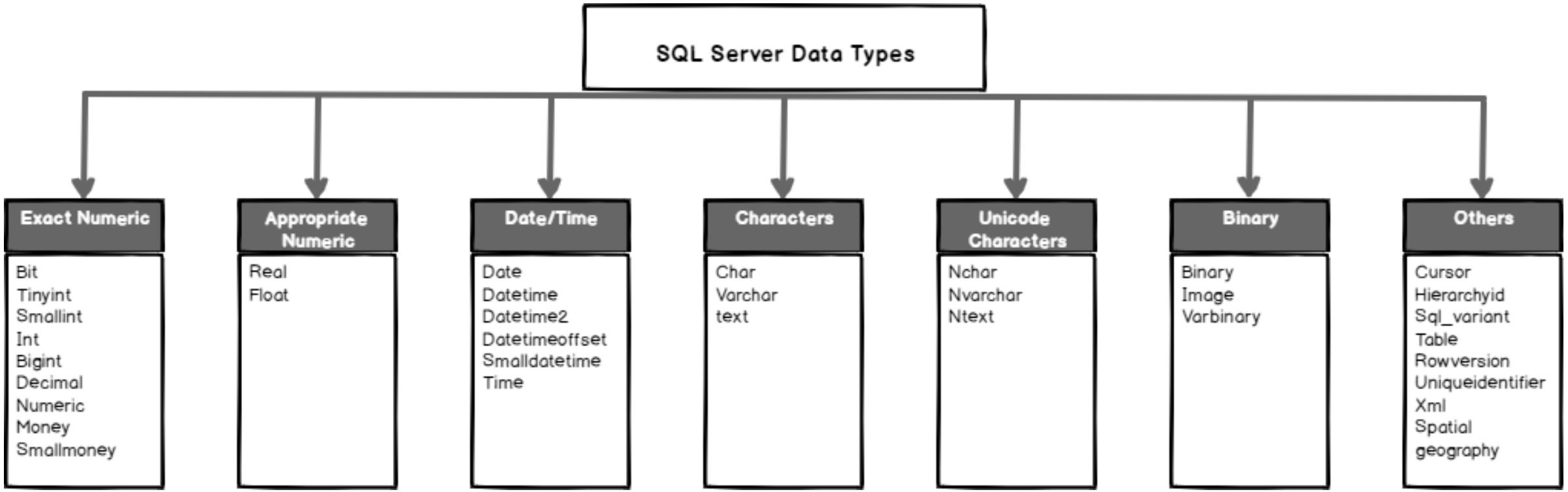
- binary : يستخدم لتخزين البيانات الثنائية بطول ثابت إذا كانت البيانات المدخلة أقصر من الطول المحدد يتم تعبئة المساحة المتبقية بالأصفار
- varbinary : يستخدم لتخزين البيانات الثنائية بطول متغير يتم تخزين البيانات فقط بالحجم الفعلي لها
- varbinary(max) **:** يستخدم لتخزين البيانات الثنائية الكبيرة، مثل الصور أو ملفات الفيديو

انواع البيانات (Data Types)

7) أنواع بيانات اخرى :

- timestamp : يستخدم لإنشاء قيمة فريدة يتم تحديثها تلقائيا عند تعديل الصف
- hierarchyid : يستخدم لتمثيل البيانات الهرمية مثل الشجرات أو التسلسلات الوظيفية
- Xml : يستخدم لتخزين ومعالجة البيانات بتنسيق XML
- Table : يستخدم لتخزين جدول بيانات مؤقت داخل إجراء مخزن أو متغير
- Geography : يمثل البيانات الجغرافية (الإحداثيات على سطح الأرض)

انواع البيانات (Data Types)



لغة معالجة البيانات (Data Manipulation Language - DML)

❖ **لغة معالجة البيانات :** تستخدم للتعامل مع البيانات المخزنة في الجداول داخل قاعدة البيانات تهدف إلى إدخال البيانات الجديدة، استرجاع البيانات الموجودة، تعديل البيانات الحالية، أو حذف البيانات غير المرغوب فيها:

1) **إدخال (INSERT) :** يستخدم لإضافة صفوف جديدة إلى جدول معين في قاعدة البيانات

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

لغة معالجة البيانات (Data Manipulation Language - DML)

2) استرجاع (SELECT) : يستخدم لاسترجاع البيانات من جدول أو أكثر بناء على شروط معينة

```
SELECT column1, column2, ...
FROM table_name;
```

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

لغة معالجة البيانات (Data Manipulation Language - DML)

3) تعديل (UPDATE) : يستخدم لتحديث القيم في صفوف معينة داخل الجدول

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;
```

لغة معالجة البيانات (Data Manipulation Language - DML)

4) حذف (DELETE) : يستخدم لحذف صفوف معينة من جدول في قاعدة البيانات

```
DELETE FROM table_name WHERE condition;
```

```
DELETE FROM table_name;
```

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

لغة معالجة البيانات (Data Manipulation Language - DML)

❖ مثال :

```sql
-- Create a table HelloWorld

CREATE TABLE HelloWorld (
    Id INT IDENTITY,
    Description VARCHAR(1000)
)


-- DML Operation INSERT, inserting a row into the table
INSERT INTO HelloWorld (Description) VALUES ('Hello World')


-- DML Operation SELECT, displaying the table
SELECT * FROM HelloWorld


-- Select a specific column from table
SELECT Description FROM HelloWorld


-- Display number of records in the table
SELECT Count(*) FROM HelloWorld


-- DML Operation UPDATE, updating a specific row in the table
UPDATE HelloWorld SET Description = 'Hello, World!' WHERE Id = 1


-- Selecting rows from the table (see how the Description has changed after the update?)
SELECT * FROM HelloWorld


-- DML Operation - DELETE, deleting a row from the table
DELETE FROM HelloWorld WHERE Id = 1
```

- إنشاء الجدول
- إدخال بيانات جديدة
- استرجاع جميع البيانات
- استرجاع عمود محدد
- عرض عدد الصفوف في الجدول
- تعديل البيانات
- حذف صف من الجدول

أمثلة شاملة و أوامر عامة

- **تحديد جميع الصفوف والأعمدة من الجدول :**

```
SELECT *
FROM table_name
```

```
SELECT col1, col2, col3
FROM table_name
```

- **تعديل صف محدد :**

```
UPDATE HelloWorlds
SET HelloWorld = 'HELLO WORLD!!!'
WHERE Id = 5
```

- **حذف جميع الصفوف:**

```
DELETE
FROM Helloworlds
```

```
TRUNCATE TABLE HelloWords
```

أمثلة شاملة و أوامر عامة

- **التعليقات :** هي أجزاء من النص داخل الكود تستخدم لتوضيح أو توثيق الكود للمبرمجين، ولكنها تتجاهل تماما عند تنفيذ الأوامر تساعد التعليقات في تحسين قراءة الكود وفهمه خاصة عند العمل في فرق أو عند مراجعة الكود لاحقا.

```
-- This is a comment
SELECT *
FROM MyTable -- This is another comment
WHERE Id = 1;
```

```
/* This is
a multi-line
comment block. */
SELECT Id = 1, [Message] = 'First row'
UNION ALL
SELECT 2, 'Second row'
/* This is a one liner */
SELECT 'More';
```

```
/*
SELECT *
FROM CommentTable
WHERE Comment = '/*'
*/
```

أمثلة شاملة و أوامر عامة

- تحديد الصفوف التي تطابق شرطا محدد :

```
SELECT <column names>
FROM <table name>
WHERE <condition>
```

```
SELECT FirstName, Age
FROM Users
WHERE LastName = 'Smith' AND (City = 'New York' OR City = 'Los Angeles')
```

- تحديث جميع الصفوف :

```
UPDATE Scores
SET score = score + 1
```

- الحصول على عدد صفوف الجدول :

```
SELECT COUNT(*) AS [TotalRowCount] FROM table_name;
```

أمثلة شاملة و أوامر عامة

- تصفية الصفوف باستخدام عبارة WHERE :

```
SELECT *
FROM sys.objects
WHERE type = 'IT'
```

- فرز النتائج باستخدام ORDER BY :

```
SELECT *
FROM sys.objects
ORDER BY create_date
```

- تصفية المجموعات باستخدام HAVING :

```
SELECT type, count(*) as c
FROM sys.objects
GROUP BY type
HAVING count(*) < 10
```

أمثلة شاملة و أوامر عامة

- **إرجاع عدد محدد من الصفوف الأولى فقط :**

```
SELECT TOP 10 *
FROM sys.objects
```

- **استخدام AS :** هي كلمة محجوزة في تستخدم لإنشاء أسماء مستعارة للأعمدة أو الجداول أثناء كتابة الاستعلامات

```
SELECT CustomerID AS ID, CustomerName AS Customer
FROM Customers;
```

```
CREATE TABLE AliasNameDemo (id INT,firstname VARCHAR(20),lastname VARCHAR(20))

INSERT INTO AliasNameDemo
VALUES      (1,'MyFirstName','MyLastName')

SELECT FirstName +' '+ LastName As FullName
FROM   AliasNameDemo
```

أمثلة شاملة و أوامر عامة

- **استخدام = :** يمكن استخدام علامة المساواة كطريقة بديلة لإعطاء أسماء مستعارة للأعمدة بدلا من استخدام الكلمة المحجوزة AS

```
CREATE TABLE AliasNameDemo (id INT,firstname VARCHAR(20),lastname VARCHAR(20))

INSERT INTO AliasNameDemo
VALUES      (1,'MyFirstName','MyLastName')

SELECT FullName = FirstName +' '+ LastName
FROM   AliasNameDemo
```

- **إدراج صفوف متعددة من البيانات :**

```
INSERT INTO USERS VALUES
(2, 'Michael', 'Blythe'),
(3, 'Linda', 'Mitchell'),
(4, 'Jillian', 'Carson'),
(5, 'Garrett', 'Vargas');
```

أمثلة شاملة و أوامر عامة

- إدراج من تحديد نتائج الاستعلام :

```
INSERT INTO Table_name (FirstName, LastName, Position)
SELECT FirstName, LastName, 'student' FROM Another_table_name
```

- إدراج في أعمدة محددة :

```
INSERT INTO USERS (FIRST_NAME, LAST_NAME)
VALUES ('Stephen', 'Jiang');
```

- طباعة رسالة في تبويب الرسائل :

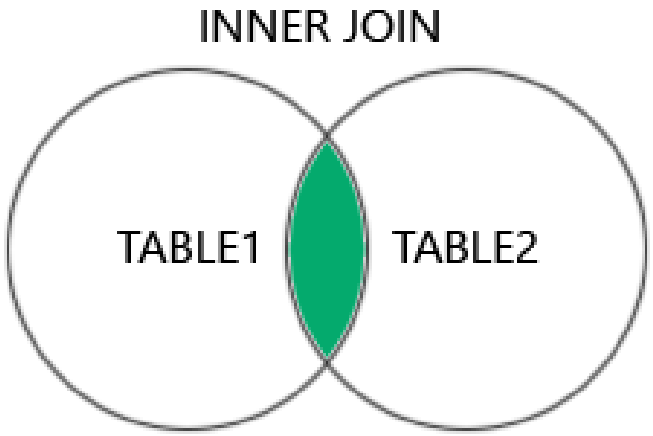
```
PRINT 'Hello World!';
```

أمثلة شاملة و أوامر عامة

- **Joins :** هي تقنية تستخدم لربط البيانات بين جدولين أو أكثر في قاعدة البيانات بناء على علاقة مشتركة بينهما من انواعها :
  - INNER JOIN : تعيد الصفوف المشتركة فقط بين الجداول، بناء على شرط الربط
  - LEFT JOIN : تعيد جميع الصفوف من الجدول الأيسر مع البيانات المطابقة من الجدول الأيمن إذا لم توجد بيانات مطابقة ترجع NULL
  - RIGHT JOIN : تعيد جميع الصفوف من الجدول الأيمن مع البيانات المطابقة من الجدول الأيسر إذا لم توجد بيانات مطابقة ترجع NULL
  - FULL JOIN : تعيد جميع الصفوف من كلا الجدولين، مع مطابقة البيانات حيثما أمكن إذا لم توجد مطابقة ترجع NULL

أمثلة شاملة و أوامر عامة

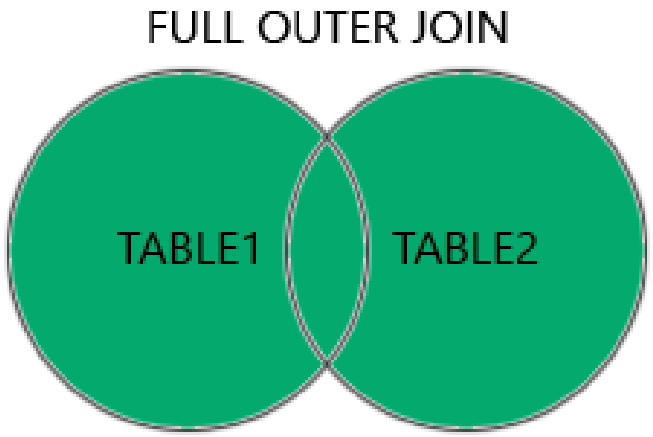
```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```




```
SELECT Products.ProductID, Products.ProductName, Categories.CategoryName
FROM Products
JOIN Categories ON Products.CategoryID = Categories.CategoryID;
```

```
SELECT ProductID, ProductName, CategoryName
FROM Products
INNER JOIN Categories ON Products.CategoryID = Categories.CategoryID;
```

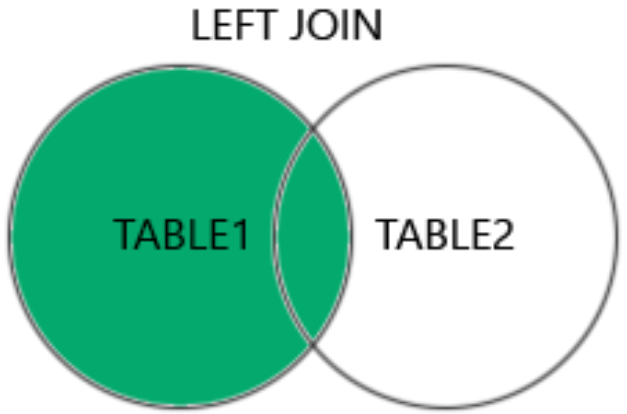
```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```



أمثلة شاملة و أوامر عامة

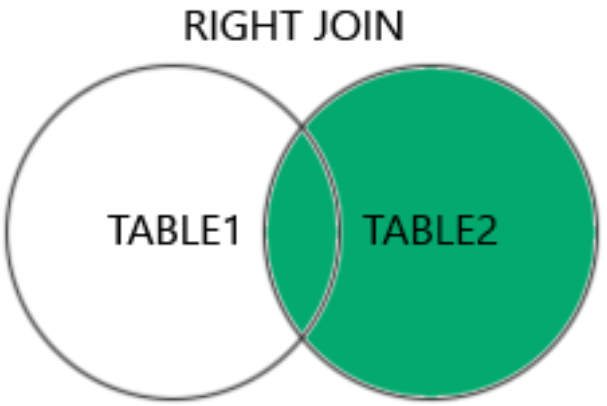
```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerName;
```




```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT Orders.OrderID, Employees.LastName, Employees.FirstName
FROM Orders
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
ORDER BY Orders.OrderID;
```



أمثلة شاملة و أوامر عامة

❖ **مثال : 1)** إنشاء عدة جداول واعمدة في قاعدة البيانات باستخدام لغة **SQL**

- جدول التصنيفات
- جدول العملاء
- جدول الأقسام
- جدول الموظفون
- جدول الطلبات
- جدول المنتجات
- جدول الموردون

```
CREATE TABLE categories
( category_id int NOT NULL,
  category_name char(50) NOT NULL,
  CONSTRAINT categories_pk PRIMARY KEY (category_id)
);

CREATE TABLE customers
( customer_id int NOT NULL,
  last_name char(50) NOT NULL,
  first_name char(50) NOT NULL,
  favorite_website char(50),
  CONSTRAINT customers_pk PRIMARY KEY (customer_id)
);

CREATE TABLE departments
( dept_id int NOT NULL,
  dept_name char(50) NOT NULL,
  CONSTRAINT departments_pk PRIMARY KEY (dept_id)
);

CREATE TABLE employees
( employee_number int NOT NULL,
  last_name char(50) NOT NULL,
  first_name char(50) NOT NULL,
  salary int,
  dept_id int,
  CONSTRAINT employees_pk PRIMARY KEY (employee_number)
);
```

```
CREATE TABLE orders
( order_id int NOT NULL,
  customer_id int,
  order_date date,
  CONSTRAINT orders_pk PRIMARY KEY (order_id)
);

CREATE TABLE products
( product_id int NOT NULL,
  product_name char(50) NOT NULL,
  category_id int,
  CONSTRAINT products_pk PRIMARY KEY (product_id)
);

CREATE TABLE suppliers
( supplier_id int NOT NULL,
  supplier_name char(50) NOT NULL,
  city char(50),
  state char(50),
  CONSTRAINT suppliers_pk PRIMARY KEY (supplier_id)
);
```

أمثلة شاملة و أوامر عامة

2) إضافة البيانات الى :

- جدول التصنيفات
- جدول العملاء
- جدول الأقسام

```
INSERT INTO categories
(category_id, category_name)
VALUES
(25, 'Deli');

INSERT INTO categories
(category_id, category_name)
VALUES
(50, 'Produce');

INSERT INTO categories
(category_id, category_name)
VALUES
(75, 'Bakery');

INSERT INTO categories
(category_id, category_name)
VALUES
(100, 'General Merchandise');

INSERT INTO categories
(category_id, category_name)
VALUES
(125, 'Technology');
```

```
INSERT INTO customers
(customer_id, last_name, first_name, favorite_website)
VALUES
(4000, 'Jackson', 'Joe', 'techonthenet.com');

INSERT INTO customers
(customer_id, last_name, first_name, favorite_website)
VALUES
(5000, 'Smith', 'Jane', 'digminecraft.com');

INSERT INTO customers
(customer_id, last_name, first_name, favorite_website)
VALUES
(6000, 'Ferguson', 'Samantha', 'bigactivities.com');

INSERT INTO customers
(customer_id, last_name, first_name, favorite_website)
VALUES
(7000, 'Reynolds', 'Allen', 'checkyourmath.com');

INSERT INTO customers
(customer_id, last_name, first_name, favorite_website)
VALUES
(8000, 'Anderson', 'Paige', NULL);

INSERT INTO customers
(customer_id, last_name, first_name, favorite_website)
VALUES
(9000, 'Johnson', 'Derek', 'techonthenet.com');
```

```
INSERT INTO departments
(dept_id, dept_name)
VALUES
(500, 'Accounting');

INSERT INTO departments
(dept_id, dept_name)
VALUES
(501, 'Sales');
```

أمثلة شاملة و أوامر عامة

❖ **مثال : 2)** إضافة البيانات الى :

- جدول الموظفون
- جدول الطلبات
- جدول المنتجات
- جدول الموردون

```
INSERT INTO suppliers
(supplier_id, supplier_name, city, state)
VALUES
(100, 'Microsoft', 'Redmond', 'Washington');

INSERT INTO suppliers
(supplier_id, supplier_name, city, state)
VALUES
(200, 'Google', 'Mountain View', 'California');

INSERT INTO suppliers
(supplier_id, supplier_name, city, state)
VALUES
(300, 'Oracle', 'Redwood City', 'California');

INSERT INTO suppliers
(supplier_id, supplier_name, city, state)
VALUES
(400, 'Kimberly-Clark', 'Irving', 'Texas');

INSERT INTO suppliers
(supplier_id, supplier_name, city, state)
VALUES
(500, 'Tyson Foods', 'Springdale', 'Arkansas');

INSERT INTO suppliers
(supplier_id, supplier_name, city, state)
VALUES
(600, 'SC Johnson', 'Racine', 'Wisconsin');

INSERT INTO suppliers
(supplier_id, supplier_name, city, state)
VALUES
(700, 'Dole Food Company', 'Westlake Village', 'California');

INSERT INTO suppliers
(supplier_id, supplier_name, city, state)
VALUES
(800, 'Flowers Foods', 'Thomasville', 'Georgia');

INSERT INTO suppliers
(supplier_id, supplier_name, city, state)
VALUES
(900, 'Electronic Arts', 'Redwood City', 'California');
```

```
INSERT INTO products
(product_id, product_name, category_id)
VALUES
(1,'Pear',50);

INSERT INTO products
(product_id, product_name, category_id)
VALUES
(2,'Banana',50);

INSERT INTO products
(product_id, product_name, category_id)
VALUES
(3,'Orange',50);

INSERT INTO products
(product_id, product_name, category_id)
VALUES
(4,'Apple',50);

INSERT INTO products
(product_id, product_name, category_id)
VALUES
(5,'Bread',75);

INSERT INTO products
(product_id, product_name, category_id)
VALUES
(6,'Sliced Ham',25);

INSERT INTO products
(product_id, product_name, category_id)
VALUES
(7,'Kleenex',null);
```

```
INSERT INTO orders
(order_id, customer_id, order_date)
VALUES
(1,7000,'2016/04/18');

INSERT INTO orders
(order_id, customer_id, order_date)
VALUES
(2,5000,'2016/04/18');

INSERT INTO orders
(order_id, customer_id, order_date)
VALUES
(3,8000,'2016/04/19');

INSERT INTO orders
(order_id, customer_id, order_date)
VALUES
(4,4000,'2016/04/20');

INSERT INTO orders
(order_id, customer_id, order_date)
VALUES
(5,null,'2016/05/01');
```

```
INSERT INTO employees
(employee_number, last_name, first_name, salary, dept_id)
VALUES
(1001, 'Smith', 'John', 62000, 500);

INSERT INTO employees
(employee_number, last_name, first_name, salary, dept_id)
VALUES
(1002, 'Anderson', 'Jane', 57500, 500);

INSERT INTO employees
(employee_number, last_name, first_name, salary, dept_id)
VALUES
(1003, 'Everest', 'Brad', 71000, 501);
```

أمثلة شاملة و أوامر عامة

3) الاستعلام يستخدم **INNER JOIN** لربط جدولين: **customers** و **orders** بهدف جلب بيانات العملاء وطلباتهم بناء على علاقة مشتركة بينهما

```
SELECT customers.customer_id, orders.order_id, orders.order_date
FROM customers
INNER JOIN orders
ON customers.customer_id = orders.customer_id
ORDER BY customers.customer_id;
```

customer_id	order_id	order_date
4000	4	2016/04/20
5000	2	2016/04/18
7000	1	2016/04/18
8000	3	2016/04/19

4) الاستعلام يستخدم **LEFT OUTER JOIN** لربط جدولين: **customers** و **orders** بهدف عرض بيانات جميع العملاء، حتى العملاء الذين ليس لديهم طلبات

```
SELECT customers.customer_id, orders.order_id, orders.order_date
FROM customers
LEFT OUTER JOIN orders
ON customers.customer_id = orders.customer_id
ORDER BY customers.customer_id;
```

customer_id	order_id	order_date
4000	4	2016/04/20
5000	2	2016/04/18
6000	NULL	NULL
7000	1	2016/04/18
8000	3	2016/04/19
9000	NULL	NULL

أمثلة شاملة و أوامر عامة

5) الاستعلام يستخدم **RIGHT OUTER JOIN** لربط جدولين: **customers** و **orders** الهدف من الاستعلام هو عرض جميع الطلبات من جدول **orders** حتى الطلبات التي لا ترتبط بأي عميل في جدول **customers**

```
SELECT customers.customer_id, orders.order_id, orders.order_date
FROM customers
RIGHT OUTER JOIN orders
ON customers.customer_id = orders.customer_id
ORDER BY customers.customer_id;
```

customer_id	order_id	order_date
NULL	5	2016/05/01
4000	4	2016/04/20
5000	2	2016/04/18
7000	1	2016/04/18
8000	3	2016/04/19

6) الاستعلام يستخدم **FULL OUTER JOIN** لربط جدولين: **customers** و **orders** الهدف من الاستعلام هو عرض جميع البيانات من كلا الجدولين، سواء كانت هناك مطابقة بينهما أم لا

```
SELECT customers.customer_id, orders.order_id, orders.order_date
FROM customers
FULL OUTER JOIN orders
ON customers.customer_id = orders.customer_id
ORDER BY customers.customer_id;
```

customer_id	order_id	order_date
NULL	5	2016/05/01
4000	4	2016/04/20
5000	2	2016/04/18
6000	NULL	NULL
7000	1	2016/04/18
8000	3	2016/04/19
9000	NULL	NULL

أمثلة شاملة و أوامر عامة

7) هذا الاستعلام هو استرجاع جميع الصفوف من جدول **suppliers** حيث يقع **supplier_id** بين القيم **300** و**600** (بما يشمل القيمتين 300 و600)

```
SELECT *
FROM suppliers
WHERE supplier_id BETWEEN 300 AND 600;
```

supplier_id	supplier_name	city	state
300	Oracle	Redwood City	California
400	Kimberly-Clark	Irving	Texas
500	Tyson Foods	Springdale	Arkansas
600	SC Johnson	Racine	Wisconsin

8) هذا الاستعلام يقوم بجلب جميع الصفوف من جدول **orders** التي تكون قيم **order_date** الخاصة بها بين **19 أبريل 2016** و **1 مايو 2016** (بما يشمل التاريخين)

```
SELECT *
FROM orders
WHERE order_date BETWEEN CAST('2016/04/19' AS DATE) AND CAST('2016/05/01' AS DATE);
```

order_id	customer_id	order_date
3	8000	2016/04/19
4	4000	2016/04/20
5	NULL	2016/05/01

أمثلة شاملة و أوامر عامة

9) يقوم هذا الاستعلام بإدخال بيانات من جدول **employees** إلى جدول **customers**

```
INSERT INTO customers
(customer_id, last_name, first_name)
SELECT employee_number AS customer_id, last_name, first_name
FROM employees
WHERE employee_number < 1003;
```

customer_id	last_name	first_name	favorite_website
4000	Jackson	Joe	techonthenet.com
5000	Smith	Jane	digminecraft.com
6000	Ferguson	Samantha	bigactivities.com
7000	Reynolds	Allen	checkyourmath.com
8000	Anderson	Paige	NULL
9000	Johnson	Derek	techonthenet.com
1001	Smith	John	NULL
1002	Anderson	Jane	NULL

10) يقوم هذا الاستعلام بتحديث بيانات صف محدد في جدول **suppliers** يتم تحديد الصف المطلوب تحديثه بناء على شرط معين في عبارة **WHERE**

```
UPDATE suppliers
SET supplier_id = 150,
    supplier_name = 'Apple',
    city = 'Cupertino'
WHERE supplier_name = 'Google';
```

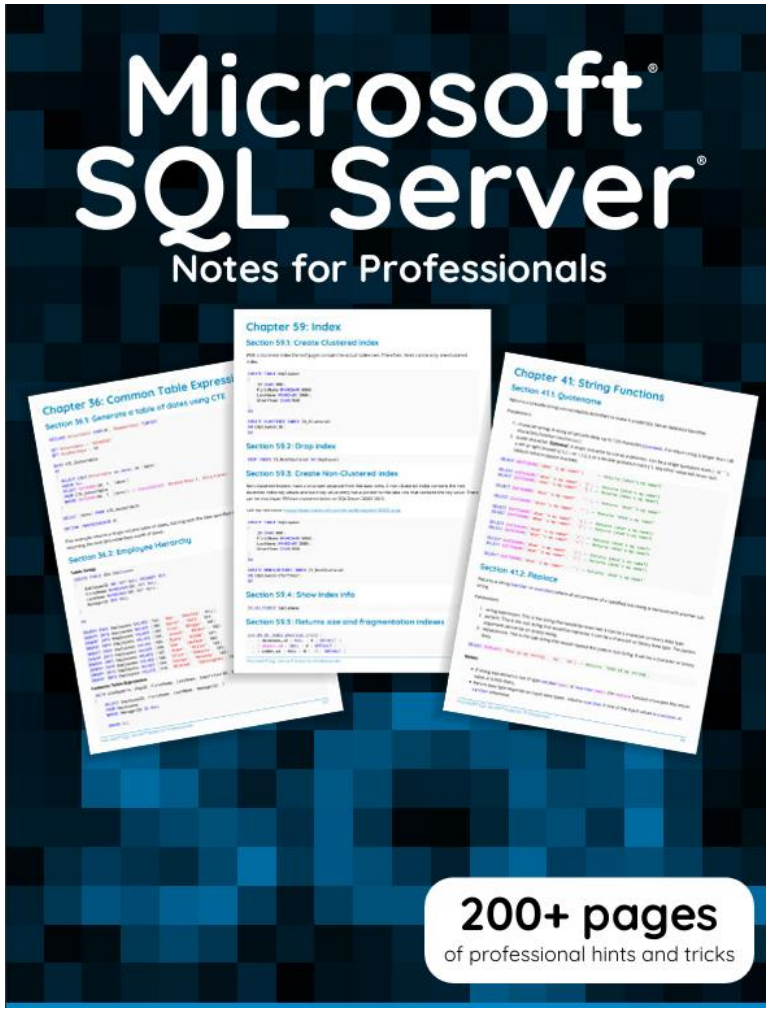
supplier_id	supplier_name	city	state
100	Microsoft	Redmond	Washington
150	Apple	Cupertino	California
300	Oracle	Redwood City	California
400	Kimberly-Clark	Irving	Texas
500	Tyson Foods	Springdale	Arkansas
600	SC Johnson	Racine	Wisconsin
700	Dole Food Company	Westlake Village	California
800	Flowers Foods	Thomasville	Georgia
900	Electronic Arts	Redwood City	California

روابط خارجية

عنوان الفيديو	الرابط
MS SQL Server For Beginners	https://youtube.com/playlist?list=PL1DUmTEdeA6J6oDLTveTt4Z7E5qEfFluE&feature=shared

المراجع

* Microsoft SQL Server Notes for Professionals



شكرا لكم