

## Day 5

- تخصص الـ DB بيتقسم لـ 4 tracks فى Microsoft :

- 1) Writing Queries (TSQL)
  - 2) Implementation \ Development of Microsoft SQL Server Database
  - 3) Maintenance and Admin of Microsoft SQL Server Database
  - 4) SQL Server BI ( Reporting \ Visualization )
- 

\* SQL Server Engine :

- الـ SQL Server والـ Oracle هم عبارة عن Fully RDBMS Engine ( يعنى بيعملو Implementation لمعظم الـ Services \ Features الموجودة فى علم الـ RDBMS )
  - الـ Access هو RDBMS بس الـ Not Fully RDBMS بس أقدر أسخدمه فى الـ DB الصغيرة ( مفيهاش Backup / Restore / BI )
  - كل الـ Cardinalities بتاعت الـ Relations بتتحول لـ One-to-many فى الـ RDBMS ( بحيث يبقى معايا Parent معاه الـ PK و Child معاه الـ FK )
- 

- الـ SQL Server بيتعرف عنه حاجتين :

1) SQL Server Version :

- ودى بتيجى كل فترة من السنين على حسب الـ Development الجديدة وتطور الـ RDBMS Tech

2) SQL Server Edition :

- ودى بتبقى عبارة عن Money / Features ( مزايا مقابل فلوس )

---

## 1) SQL Server Versions History :

- بتقسم لـ 3 Generations :

1) الـ 1<sup>st</sup> Generation :

ودا كان مربوط بالـ Windows ( مع كل إصدار OS بينزل SQL Server Tool خاص بيه )

2) الـ 2<sup>nd</sup> Generation :

ودا بدأ يكون Windows Independent

3) الـ 3<sup>rd</sup> Generation :

ودا بداية التنافس في الـ Features والـ Security والـ Performance

1 <sup>st</sup> Generation	
SQL Server Version	Features
SQL 6.0/6.5 (1995)	First version designed specifically for Windows NT Replication
SQL Server 4.2 (1992)	Developed for Windows NT 3.1
SQL Server 1.0 (1989)	Developed by Microsoft, Sybase, and Ashton-Tate for OS/2

2 <sup>nd</sup> Generation	
SQL Server Version	Features
SQL2000	Focus on Performance and Scalability XML support Data Mining Reporting Services
SQL Server 7.0 (1999)	Restructure of Relational Server Data Transformation Services Online Analytical Processing

3 <sup>rd</sup> Generation	
SQL Server Version	Features
SQL 2014,2017,2019	Security&Performance
SQL 2012	Always On Power View File Table Sequence Data Quality Service
SQL2008/SQL2008 R2	Power Pivot Enhance SharePoint Integration T-SQL (Ranking, Merge, Output) Improve and enhance for BI Tools
SQL2005	High Availability(includes DB Mirroring) Security Enhancements (DB Schema) Integration Services SQLCLR XML and Web services supports

## \* SQL Server Editions :

- فيه 5 types of Editions ودا على حسب إنت عاوز الـ SQL Server فى إيه ؟

Edition	Description
Enterprise	For large scale, business-critical applications
Standard-Developer	For small/medium, departmental applications
BI Edition	For BI Services
Express	Entry level/learning edition
Azure	For Cloud

- عيوب الـ Cloud :

(1) إنه بيكون Available for Vendors بس

(2) محتاج Internet Network يكون High

## \* SQL Server Installation :

- محتاج وأنا بـ Install الـ SQL Server أكون Aware إنى بعمل حاجة إسمها Instance على الجهاز

- الـ Instance عبارة عن مجموعة من ( Services + Application )

- الـ 5 Main Services اللى مع الـ SQL Server :

1) DB Engine (SQL Server MSSQLSERVER) :

- ودى الـ Service اللى بـ Connect عليها عشان أقدر أتعامل مع الـ Database

2) SSIS ( SQL Server Integration Service)

3) SSRS (SQL Server Reporting Service)

4) SSAS (SQL Server Analytics Service)

5) Data Quality Service

- الـ Services دى هتبقى موجودة فى الـ Windows Background أقدر أعملهم Run \ Start وأستخدم اللى عاوزه فيهم

- الـ APPs وظيفتها إنها تكلم الـ Services وتقدر تستخدمها من خلال GUI سهل وبسيط بدل ما تستخدم الـ CMD

- ممكن أ Connect على الـ Services بإستخدام الـ CMD وممكن الـ Excell برضو

- الـ Apps المستخدمة فى الـ SQL Server DB :

### 1) SSMS ( SQL Server Management Studio) :

- ودا بقدر من خلاله أكتب الـ Queries وأشوف تأثيرها وأتعامل مع الـ Database

- بـ Connect على الـ MSSQLSERVER Service

### 2) SQL Server Data Tools :

- بقدر من خلاله أعمل BI ( Integration و Reporting و Analytics على الـ Data)

- بـ Connect على الـ 3 Services ( SSIS \ SSRS \ SSAS )

### 3) Data Quality Client :

- بقدر من خلاله أتأكد من الـ Quality بتاع الـ Data

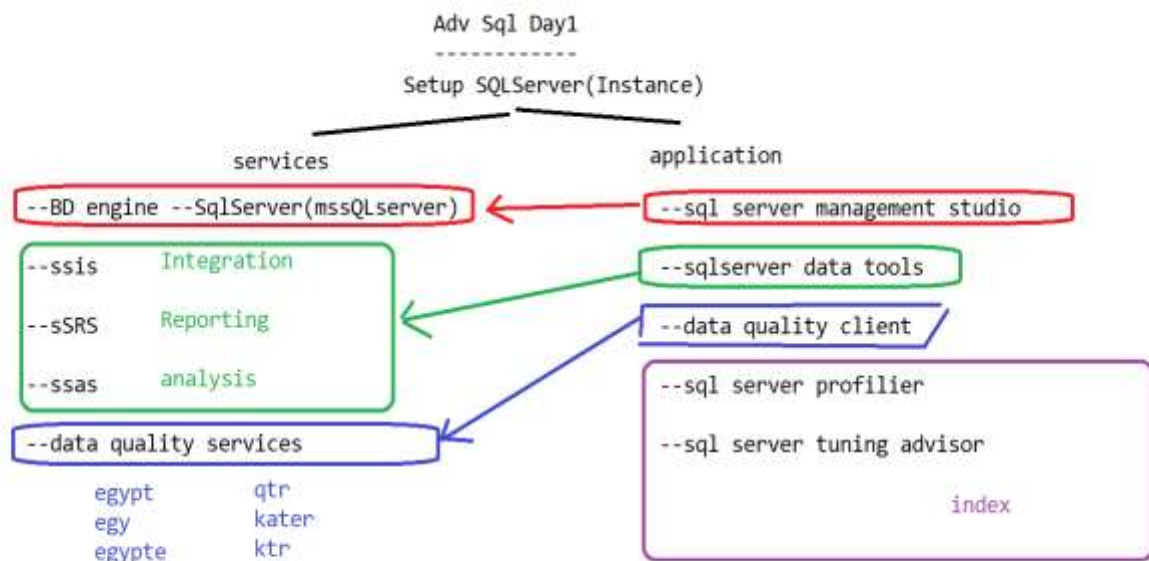
( Egypt = egypt = Gypt )

- بـ Connect على الـ Data Quality Service

### 4) SQL Server Profiler :

### 5) SQL Server Tuning Advisor :

- الإثنين بيستخدموا فى الـ Performance ودا هيكون شرحه بـ



- أقدر أعمل Multiple Install لـ SQL Server على نفس الجهاز ( دا ممكن أستفيد منه إن لو عندي DB كتير و Users كتير أوزعهم على الـ Services ) وببمساعء على تحسين الـ Performance وءا لإن كل Service واءءه جزء من الـ Processor والـ Memory
- فله نوعين من الـ instance وهم ( Default / Named )

### 1) Default Instance :

- فى أول Setup
- بياخد أسم الجهاز
- أقدر أ Connect عن طريق :
- .
- Local
- Pc-Name
- IP ( Current PC)

### 2) Named Instance :

- بعء أول Setup ( لو عاوز أسطب Service ثانى )
- بءاخذ إسم مءءء أنا بءءءه للـ Instance
- أقدر أ Connect عن طريق :
- . / Instance Name
- Local / Instance Name
- Pc-Name / Instance Name
- IP (Pc) / Instance Name

- 
- ممكن أعمل Multiple Installation لـ Named Instance بأسماء مءءءة ولكن الـ Default هيا واءءة ( 1<sup>st</sup> Instance ) وممكن تكون Named عاءى بس Default
  - فاءة الـ Named Instance هى لو عاوز أعمل Remote Connection

## \* Authorization Vs Authentication :

- الفرق بينهم إن :

الـ Authentication هي التأكد من الشخص ( باستخدام أى طريقة User \ Password )

الـ Authorization هي التأكد من صلاحيات الشخص (Access / Permissions)

## Authentication :

- عندى نوعين من الـ Authentications وهم :

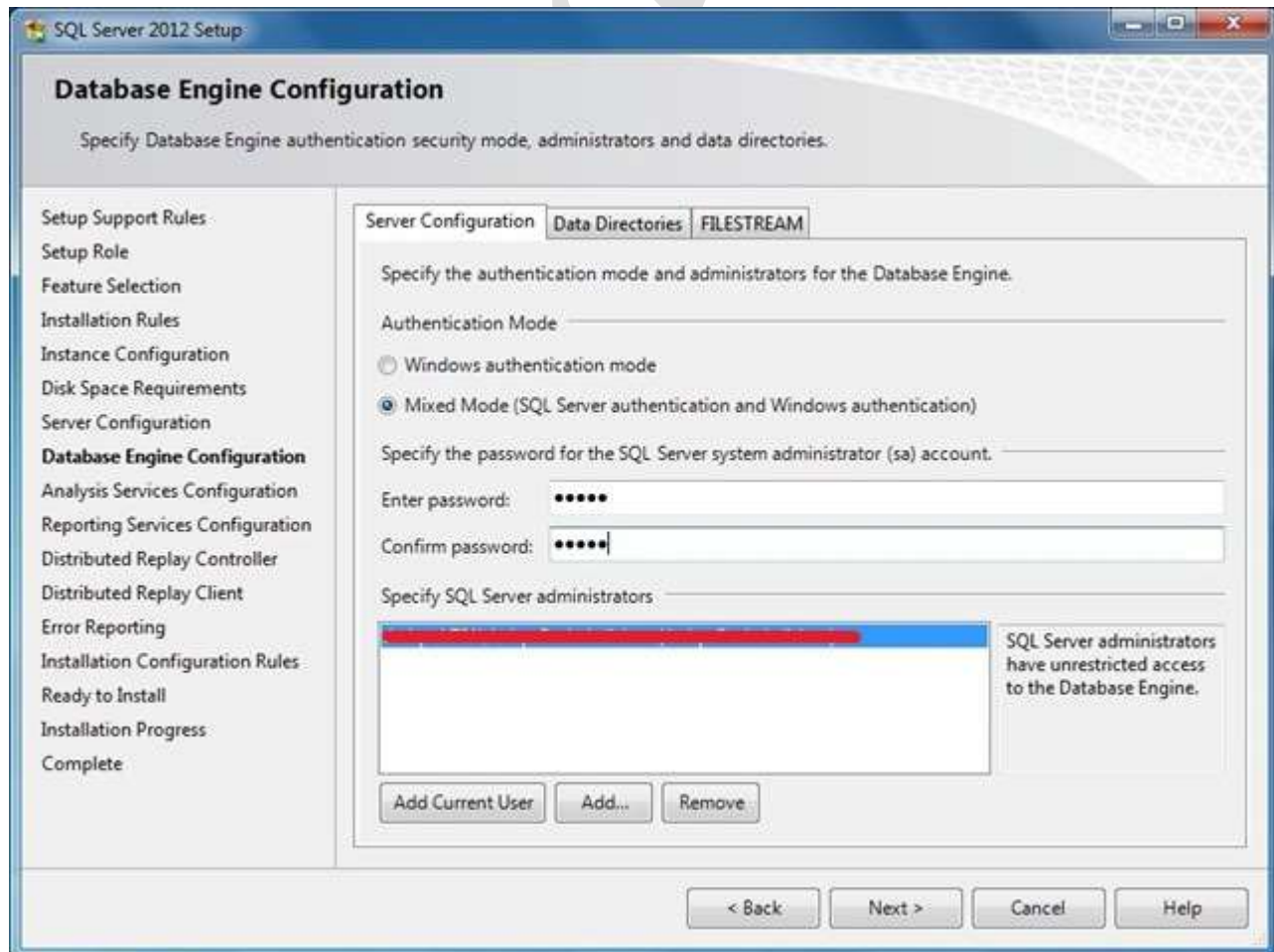
### 1) Windows Authentication :

- يكون الـ Windows Admin = SQL Server Admin

### 2) SQL Server Authentication

- بتحدد أنت كـ DBA الـ Users والـ Password الخاصة بيهم وبتحدد إمكانية أن الـ user يقدر يغير الـ Password ( فى الـ Setup و ممكنة فى الـ SSMS برضو )

- عشان أقدر أعمل كذا بختار Mixed mode فى الـ SQL Server Installation



- أقدر أغير الـ Users فى الـ SSMS عن طريق :

Server right click → Properties → Security → Mixed → server right click → Restart ( عشان يطبق التغيرات ويشغله تانى )

Security Folder → Logins Folder right click → New Login → Enter User\Pass

- كدا أنا عملت login لـ User يقدر يـ Connect على الـ Server بيه ( بس مش هيقدر يشوف الـ Databases اللى على الـ Server لأنه مش User فى الـ DB )

- عشان أخليه User فى الـ Database :

Database المطلوبة → Security Folder → user Folder right click → new user

- برضو هيبقا شايف الـ DB بس مش قادر يشوف الـ Tables ودا موضوع الـ **Authorization**

- فى أول Setup لو عملت الـ Mixed هيتـ Create 2 Admins

1) Win Admin

2) Sa Admin ( Remote Admin )

( بيطلب منك password لـ Sa Admin )

**\* Select Top (Number) \* \ Columns**

**From (table) :**

- بيختار أول Number of Records في الـ Table ويعرضهم

- ممكن أحدد Records معينة باستخدام الـ Where Clause

( هينفذ الـ Where Clause الأول وبعدين يختار الـ Top (Number) of Records )

- لو استخدمت **with ties** **Select top (4)** لازم أستخدم معاها **Order by** وإستخدامها هيكون بتختار

أعلى record بناء على الـ Column value اللي مستخدمه في الـ Order by وبتجيب الـ

Records اللي الـ Column value بتاعها زي الـ 4<sup>th</sup> Record بس ( فبالتالي ممكن يكون ناتجها

4 or more records ( الباقي هما الـ Records اللي زي الـ 4<sup>th</sup> record )

---

**\* Select new id () :**

- دى عبارة عن Built in function بترجع قيمة لـ GUID ( Global Unversed ID ) ودا بيكون

ID على مستوى الـ Server ( Unique على مستوى الـ Server ) وكمان بيبقى Randomized في

كل مرة بتـ Run فيها

**Select \* , new id ()**

**From Students**

( - هيعرض الـ data الخاصة بكل الـ students وكمان مع كل Student هيبقى ليه GUID )

- في كل Run هيفتلف الـ GUID بتاع كل Record )

- لو استخدمت الـ new Id() في order by هيعمل ترتيب عشوائي في كل مرة بيـ Run

---

**Execution Order :**

**From → Join → on | Where → Grouping → Having (aggregate Functions) →**

**Select (distinct \ aggregate Functions ) → Order by → Top**



### **\*DB Object Full Path :**

- لكل Database object وهم الـ rule \ stored procedure \ function \ view \ Table ليهم Full Path وهو عبارة عن :

[Server name] . [ DB Name] . [ Schema Name ] . [Object Name]

- فإيدته إنى لو واقف فى Db معينة وعاوز أجيب Data من Db ثانية أقدر أستخدم الـ Full Path ومش هيعترض فى الـ Query
- فى الـ SSMS انت بتـ Connect على الـ Server وتعمل Use لـ Database وهو بيستخدم الـ Default Schema (dbo) وبالتالي بيحتاج منك إسم الـ Object بس

---

### **\* Select \* Into Table :**

- عبارة عن DDL Query بيعمل Copy لـ data + Structure من Table لـ other Table
- بـ Create Table بدلالة Table
- ممكن بدل الـ \* أختار Columns عادى
- لو عاوز أخذ الـ Structure بدون الـ Data هيتم عمل Where Clause معاها شرط مش بيتحقق ( فبالـتالى مفيش Data )

Select \* into New\_Table

From Old\_Table

Where 1 = 2

- طريقة ثانية لأخذ الـ Structure بس وهى إن أخذ الـ Queries Script عن طريق :  
Table Right Click → Script → هيطهر الكود

- طب لو عاوز العكس؟؟ الـ Data بدون الـ Structure

هستخدم الـ Insert based on Select ( بس دى بتنقل Values من Table لـ other table )

Insert into Table2

Select (same Structure) from Table1

- لازم يكون الـ structure فى الـ Two tables واحد

- أو ممكن يكون الـ T2 جزء من الـ T1 وبالتالي هحدد الـ Columns المستخدمة فى الـ Select

## \* Ranking Functions :

- عبارة عن Special Built in Functions يستخدم لإجابة على بعض الـ Business Questions
- عبارة عن 4 Functions :

- 1) Row\_Number
- 2) Dense\_Rank
- 3) NTiles(Group)
- 4) Rank()

- هدفهم إنهم يحطوا أرقام في الـ Tables أقدر استخدمها بعد كذا في الـ Business
- الأرقام بتدل على Record Rank بالنسبة لـ Criteria معينة

### 1) Row\_Number :

- بيتم ترتيب الـ Table باستخدام Column معين
- بعد الترتيب بيتم إعطاء كل Record رقم من 1 لـ N بيدل على الـ Rank الخاص بالـ Record في الـ Table بناء على الـ Ordering Column

- Syntax :

**Select \* , Row\_Number () over (order by Salary desc) As RN**

**From Employee**

- الـ Output هيكون ترتيب الـ Employees تنازلي باستخدام الـ salary وكل employee معاه column اسمه RN بيدل على الـ Rank

eid	ename	esal	did	RN
15	ahmed	10000	10	1
14	ali	10000	10	2
12	eman	9000	10	3
1	nada	9000	10	4
2	reem	9000	10	5
3	khalid	8000	10	6
7	mohamed	7000	20	7
8	sayed	7000	20	8
6	hassan	6000	20	9
5	omar	6000	20	10
9	sally	5000	30	11
10	shimaa	4000	30	12
11	hana	4000	30	13
12	lama	3000	30	14

## 2) Dense\_Rank :

- سيتم ترتيب الـ Table باستخدام Column معين

- بعد الترتيب سيتم إعطاء كل Record رقم من 1 لـ N بديل على الـ Rank الخاص بالـ Record في الـ Table بناء على الـ Ordering Column بشروط ( لو تساوى أكثر من Record في نفس الـ ordering Column Value هياخدو نفس الرقم في الـ Rank )

- Syntax :

Select \* , Dense\_Rank () over (order by Salary desc) As DN

From Employee

- الـ Output هيكون ترتيب الـ Employees تنازلي باستخدام الـ salary وكل employee معاه column اسمه DN بديل على الـ Rank

eid	ename	esal	did	DR
15	ahmed	10000	10	1
14	ali	10000	10	1
12	eman	9000	10	2
1	nada	9000	10	2
2	reem	9000	10	2
3	khalid	8000	10	3
7	mohamed	7000	20	4
8	sayed	7000	20	4
6	hassan	6000	20	5
5	omar	6000	20	5
9	sally	5000	30	6
10	shimaa	4000	30	7
11	hana	4000	30	7
12	lama	3000	30	8

- الـ Employees المتساوية في الـ Esal معاهم نفس الـ DR Rank

- يستخدم الـ Dense Rank لو مش عاوز تكرر الـ Value Ranking

### 3) Number of Tiles :

- سيتم ترتيب الـ Table باستخدام Column معين

- بعد الترتيب سيتم تقسيم الـ Table لـ Number of Groups

- سيتم تقسيم الـ Groups بعدد Records مقارب بحيث يكون النقص دائماً في آخر الـ Groups ( يعني لو عندى فى المثال 14 Employee وعاوز أقسمهم لـ 3 Groups فبالتالى هيبقى العدد فى كل group كالتى :

Group 1 → 5 records

Group 2 → 5 records

Group 3 → 4 records

طبعا الـ Group 1 معاه Rank Number = 1 ..... وهكذا G2 و G3 )

- النقص يكون فى آخر Groups بحيث يكون النقص فى كل Group عبارة عن 1 فقط ( يعني لو كان 13 Employee كان هيبقى التقسيم

G1 → 5 , G2 → 4 , G3 → 4

النقص فى آخر جروب بـ 1 واللى قبل الأخير بـ 1 وهكذا )

- Syntax :

Select \* , NTiles (3) over (order by Salary desc) As G

From Employee

- الـ Output هيكون ترتيب الـ Employees تنازلى باستخدام الـ salary هيتم تقسيم الـ Employees لـ 3 Groups كل Group ليه Ranking value

eid	ename	esal	did	G
15	ahmed	10000	10	1
14	ali	10000	10	1
12	eman	9000	10	1
1	nada	9000	10	1
2	reem	9000	10	1
3	khalid	8000	10	2
7	mohamed	7000	20	2
8	sayed	7000	20	2
6	hassan	6000	20	2
5	omar	6000	20	2
9	sally	5000	30	3
10	shimaa	4000	30	3
11	hana	4000	30	3
12	lama	3000	30	3

#### 4) Rank :

- سيتم ترتيب الـ Table بالـ ordering Column

- يبقى لكل Record في الـ result set عبارة عن Rank ( بحيث المتشابه يياخدوا نفس قيمة الـ Rank ولكن الـ Next Rank Value بتعتمد على الـ Previous Rank Value و عدد الـ Records في الـ previous Rank )

- Syntax :

**Select \* , Rank () over (order by Salary desc) As R**

**From Employee**

- الـ Output عبارة عن ترتيب الـ Employee بـ Salary و كل Employee معاه Column يسمى R معاه الـ rank بتاع الـ Employee في الـ ordering

- مع معلومة إن المتساوي هياخد نفس الـ Rank

- كمان الـ Rank Value بتعتمدا على الـ Previous Rank Value و عدد الـ Records  
( Rank Value = Previous Rank + Records ) بحيث

	Fname	Lname	Salary	R
1	Ahmed	Negm	3600	1
2	Ahmed	Ali	3600	1
3	Amr	Omran	2500	3
4	Edward	Hanna	2500	3
5	Maged	Raof	2500	3
6	Kamel	Mohamed	1800	6
7	Noha	Mohamed	1600	7
8	Hanaa	Sobhy	800	8
9	Mariam	Adel	750	9
10	Ahmed	gomaa	NULL	10

- عشان أقدر أستخدم الـ RN , DN , G , R Columns لازم يكون اللي فات موجود في SubQuery  
وأستخدم الـ where Clause في الـ outer Query ( عشان يبقى شايف الـ Columns )

**Syntax :**

**Select \***

**From ( Select \* , Row\_Number () over (order by Salary desc) As RN**

**Select \* , Dense\_Rank () over (order by Salary desc) As DR**

**Select \* , NTiles (3) over (order by Salary desc) As G**

**Select \* , Rank () over (order by Salary desc) As RN ) As NewTable**

**Where RN = 5**

**DR = 2**

**G = 2**

---

**\* Partitioning :**

- ممكن أستخدم الـ Partitioning مع الـ Ranking Function ودا هيتم تقسيم الـ Table لـ  
Partitions بناء على الـ Partitioning Column ومع كل Partition هيتم إستخدام الـ ranking  
بإستخدام الـ Ordering Column

**- Syntax :**

**Select \* , Row\_Number () over (Partition by Depart ID order by Salary desc)**  
**As RN**

**Select \* , Dense\_Rank () over (Partition by Depart ID order by Salary desc) As**  
**DN**

**Select \* , NTiles (3) over (Partition by Depart ID order by Salary desc) As G**

**Select \* , Rank () over (Partition by Depart ID order by Salary desc) As G**

**From Employee**

## \* SQL Data types :

- انت محتاج تعرف الـ data اللى معاك تتخزن فى حجم قد إيه و نوعها عشان تقدر تحدد الـ DT المناسب
- عبارة عن 6 Categories of Datatype وهم :

- 1) Numeric Datatype
  - 2) Decimal Datatype
  - 3) Character \ String Datatype
  - 4) Datetime Datatype
  - 5) Binary Datatype
  - 6) others
- 

### 1) Numeric Datatype :

- Bit → Boolean Values (0,1) (True \ False)
- Tiny int → 1Byte
- Small int → 2 Byte
- Int → 4 Byte
- Big int → 8 Byte

يمكن استخدام الـ Unsigned مع الـ int بأنواعها

---

### 2) Decimal Datatype :

- Small money → 4 byte , 4 floating points
- Money → 8byte , 4 floating points
- Real → , 7 floating points
- Float → , more than 7 floating points
- Dec → datatype and validation on datatype

Dec ( Number of digits , number of floating points)

يعنى فى حالة Dec(5,2) يعنى خمس أرقام منهم 2 عشرى ( و أى حاجة أقل من كذا عادى مثلا 3 و 2 )

### 3) Character Datatype :

- char (number) :

Fixed length of maximum size (number)

- varchar (number) :

Variable Length of maximum size (number)

- nchar (number) :

Fixed length of maximum size (number) \ Accept Unicode

- nvarchar (number) :

Variable Length of maximum size (number) \ Accept Unicode

- nvarchar (MAX) :

Variable Length of maximum size (up to 2GB) \ Accept Unicode

- الـ Unicode هى اللغات غير الـ English

- الفرق بين الـ Fixed و الـ variable هى إن الـ Fixed بيحجز الـ Maximum سواء كان أقل أو قد الـ maximum لكن الـ variable على حسب الـ value بيحجز الـ bytes اللى تـ Fit الـ value

Fixed = Max size Number

Variable <= Max size Number

---

### 4) Datetime Datatype :

- Date :

MM\DD\YYYY

- Time :

HH : MM : SS.000

- فيه إن تقسم الثانية على 1000

- time(7) :



**HH : MM : SS.0000000**

- فيه إن تقسم الثانية على 10,000,000

- **smalldatetime :**

**MM / DD / YYYY HH:MM : 00**

الثانية بتكون أصفار

- **datetime :**

**MM / DD / YYYY HH:MM : SS.000**

- **datetime2(7)**

**MM / DD / YYYY HH:MM : SS.0000000**

- **datetimeoffset :**

**Data + Time + Timezone**

**11/24/2020 10:30 +2:00 (cairo timezone)**

---

**5) Binary Datatype :**

- **binary**

- **Image**

---

**6) others :**

**Xml**

**Unique Identifier**

**Sql\_Variant**

**\* SQL :**

**- Syntax :**

**Select** instructor\_name ,

**Case**

**When** salary >= 3000 **then** 'High salary'

**When** salary < 3000 **then** 'Low salary'

**Else** ' No Value'

**End** as newsal

**From** Instructor

- تستخدم Case فى حالة إن علوز أعرض حاجة بناء على Condition

( فى المثال السابق هو بيعرض High او Low فى column اسمه nwsal فى حالة الـ condition على الـ salary )

- يستخدم الـ case مع الـ Select \ Update

**Update** instructors

**Set** Salary =

**Case**

**When** salary >= 3000 **then** Salary \* 1.10

**Else** salary < 3000 **then** Salary \* 1.20

**End**

- لو كان الـ Condition عبارة عن حالة واحدة ( If \ Else ) ممكن تستخدم الـ Function iif()

بدل الـ Case Keyword

**Select** instructor\_name , **iif** ( salary >= 3000 , 'High' , ' Low' )

**From** Instructor

- دى زى الـ Ternary operator فى الـ programming

## \* Convert vs Cast

- الإيتين بيستخدموا لتحويل الـ Date لـ String

- الفرق الجوهرى إن الـ convert بتاخد معاها parameter تالت بيعبر عن الـ Format الـ date

- الـ Convert أفضل من الـ cast فى الإستخدام مع الـ date

- syntax :

**Select** Convert ( varchar(20) , getdate() , 102)

**Select** cast (getdate() as varchar(20) )

- كل Format ليه رقم بيدل عليه

- ممكن أستخدم Function اسمها format وأقدر أنا أكتب الـ Format as a string

**Select** Format ( getdate() , “dd / mm / yyyy ”

Some of Formats :

dd – mm – yyyy → 23 – 11 – 2020

dddd – mmmm – yyyy → Monday November 2020

ddd – mmm – yy → Mon Nov 20

dddd → Monday

mmmm → November

hh : mm : ss → 02 : 02 : 55

HH → 14 (24 hours)

hh tt → 02 PM

dd-mm-yyyy HH:mm:ss tt → 23-11-2020 02:03:15 PM

---

Format with dd Vs Day Function :

**Select** Format ( getdate() , “dd”) → return String

**Select** day (getdate()) → return Int

## **eomonth** Function (End of month) :

- بترجع آخر يوم فى الشهر الـ input ليها

Select eomonth ( get date () ) → هترجع آخر يوم فى الشهر الحالى

Select eomonth ( '1-1-2020' ) → هترجع آخر يوم فى شهر 1

Ahmed Negas