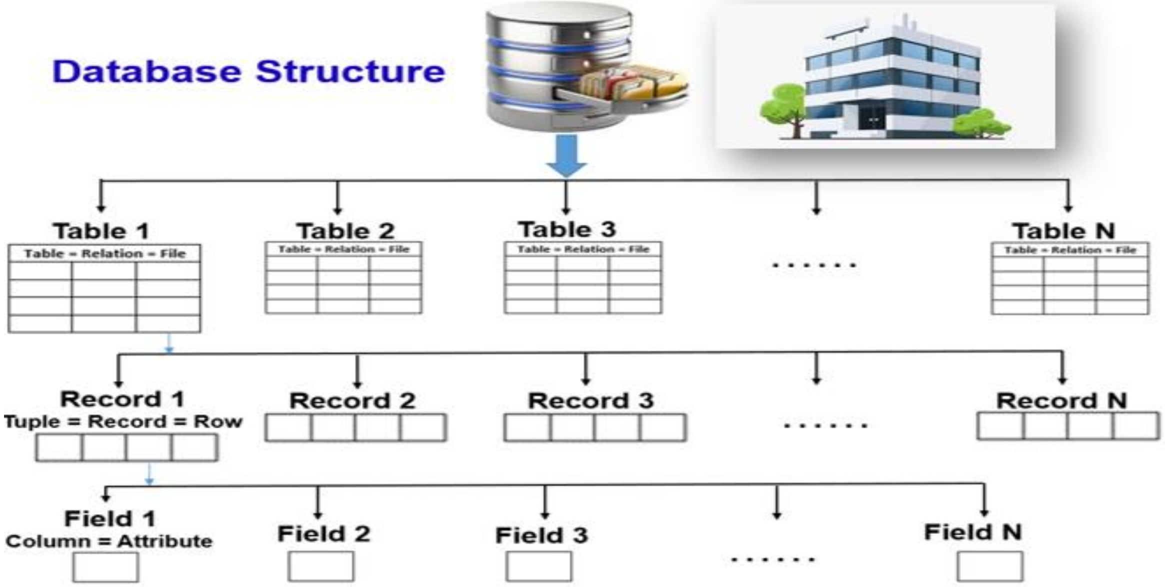
Databases is a collection of related data.
Store
▶ Store organized data
تخزين البيانات بشكل منظم ومرتب
Retrieve
▶ Retrieve it easily
استرجاع البيانات بسرعة وسهولة ودقة

Structure Query Language (SQL)

DDL Data Definition Language	DML Data Manipulation Language	DCL Data Control Language	TCL Transaction Control Language
➢ Create	➢ Insert	➢ Grant	➢ Commit
➢ Drop	➢ Delete	إعطاء صلاحيات	➢ Rollback
➢ Truncate	➢ Update	➢ Revoke	➢ Save Point
➢ Rename		سحب صلاحيات	➢ Set Transaction
➢ Alter	DQL Data Query Language		
➢ Comments	➢ Select		

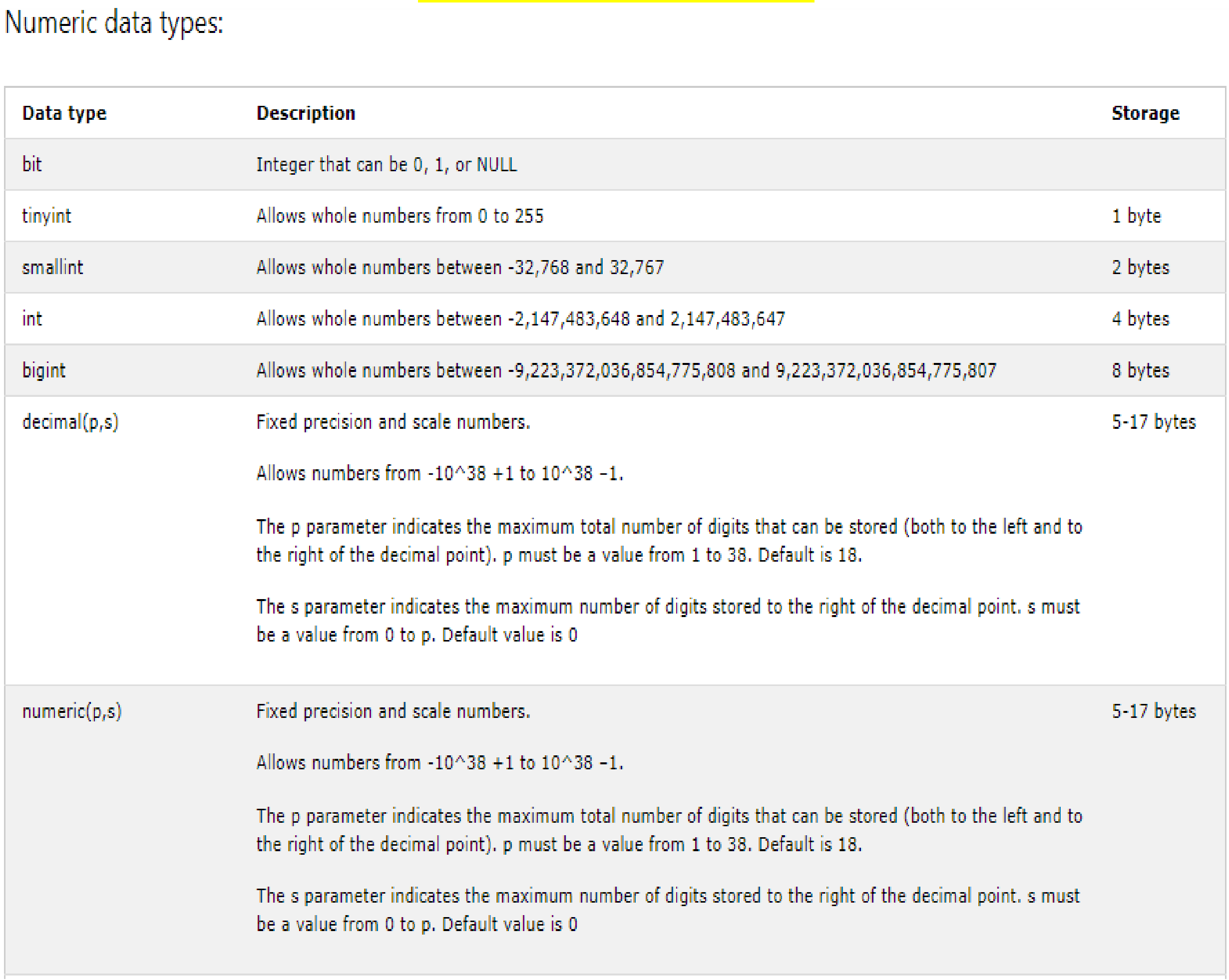


SQL Constraints

مجموعة القيود التي يتم فرضها على Tables or Records حتى نضمن صحة وسلامة البيانات المدخلة.

1. Data types (Domain)
2. Null/Not Null
3. Unique (Can be Null)
4. Check (GPA >= 0 and GPA <= 4)
5. Default (‘Egyptian’) for (Nationality)
6. Primary Key (Not Null & Unique)
7. Foreign Key (Not Null)

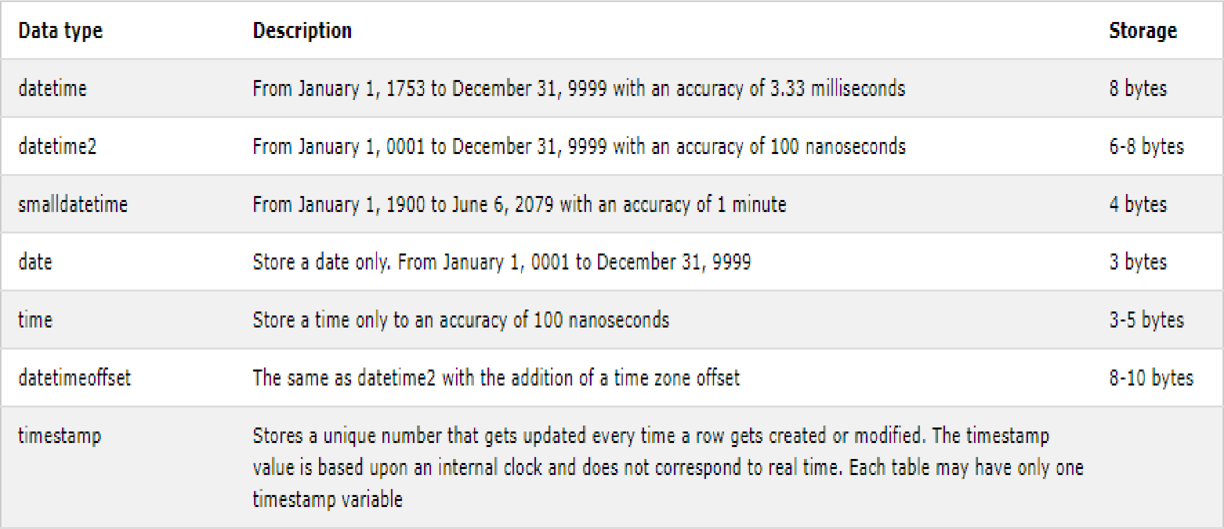
SQL Data Types



Numeric data types:

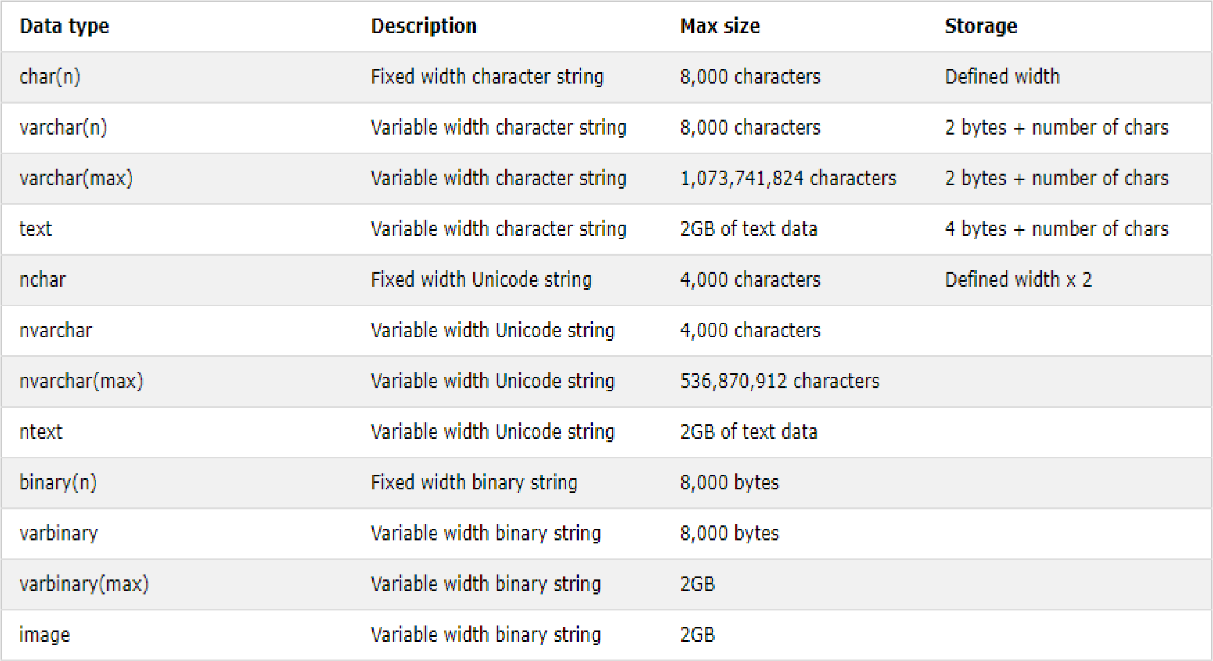
Data type	Description	Storage
bit	Integer that can be 0, 1, or NULL	
tinyint	Allows whole numbers from 0 to 255	1 byte
smallint	Allows whole numbers between -32,768 and 32,767	2 bytes
int	Allows whole numbers between -2,147,483,648 and 2,147,483,647	4 bytes
bigint	Allows whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807	8 bytes
decimal(p,s)	Fixed precision and scale numbers. Allows numbers from -10^38 +1 to 10^38 –1. The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18. The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0	5-17 bytes
numeric(p,s)	Fixed precision and scale numbers. Allows numbers from -10^38 +1 to 10^38 –1. The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18. The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0	5-17 bytes

Date and Time data types:

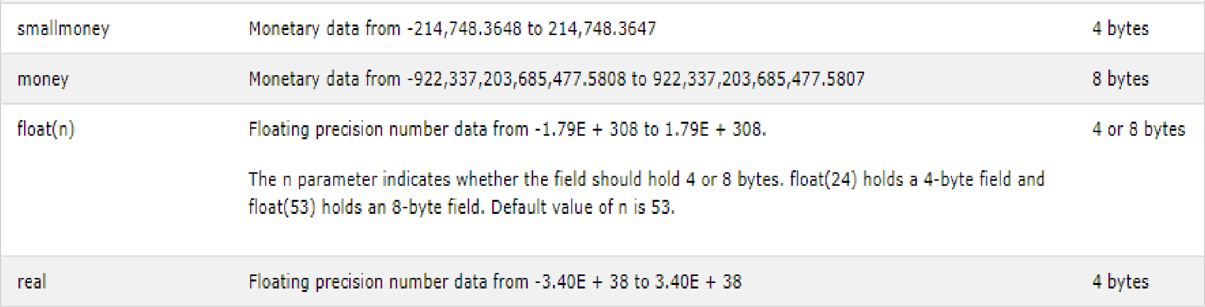


Data type	Description	Storage
datetime	From January 1, 1753 to December 31, 9999 with an accuracy of 3.33 milliseconds	8 bytes
datetime2	From January 1, 0001 to December 31, 9999 with an accuracy of 100 nanoseconds	6-8 bytes
smalldatetime	From January 1, 1900 to June 6, 2079 with an accuracy of 1 minute	4 bytes
date	Store a date only. From January 1, 0001 to December 31, 9999	3 bytes
time	Store a time only to an accuracy of 100 nanoseconds	3-5 bytes
datetimeoffset	The same as datetime2 with the addition of a time zone offset	8-10 bytes
timestamp	Stores a unique number that gets updated every time a row gets created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. Each table may have only one timestamp variable	

String data types:



Data type	Description	Max size	Storage
char(n)	Fixed width character string	8,000 characters	Defined width
varchar(n)	Variable width character string	8,000 characters	2 bytes + number of chars
varchar(max)	Variable width character string	1,073,741,824 characters	2 bytes + number of chars
text	Variable width character string	2GB of text data	4 bytes + number of chars
nchar	Fixed width Unicode string	4,000 characters	Defined width x 2
nvarchar	Variable width Unicode string	4,000 characters	
nvarchar(max)	Variable width Unicode string	536,870,912 characters	
ntext	Variable width Unicode string	2GB of text data	
binary(n)	Fixed width binary string	8,000 bytes	
varbinary	Variable width binary string	8,000 bytes	
varbinary(max)	Variable width binary string	2GB	
image	Variable width binary string	2GB	



smallmoney	Monetary data from -214,748.3648 to 214,748.3647	4 bytes
money	Monetary data from -922,337,203,685,477.5808 to 922,337,203,685,477.5807	8 bytes
float(n)	Floating precision number data from -1.79E + 308 to 1.79E + 308. The n parameter indicates whether the field should hold 4 or 8 bytes. float(24) holds a 4-byte field and float(53) holds an 8-byte field. Default value of n is 53.	4 or 8 bytes
real	Floating precision number data from -3.40E + 38 to 3.40E + 38	4 bytes

Data Definition Language (DDL)

- Create DB or its Objects like (tables, index, functions, views, stored procedure, triggers)

```
Create Database Database_Name ;
```

```
Create Table Table_Name
(
 Column_1     DataType     Constraints ,
 Column_2     DataType     Constraints ,
   ............        ..............        ............. ,
 Column_N     DataType     Constraints
) ;
```

```
Create Database Company ;
```

```
Create Table Employees
(
Emp_ID     int     Primary Key ,
Emp_Name     varchar (50)     Not Null ,
Salary       money       Not Null ,
FUN     varchar (20)     Not Null ,
Phone     varchar (20)     Unique ,
Address     nvarchar (50)     Not Null
) ;
```

Note:

```
Primary key (ID),
Foreign key (Dept_ID) References Department (Dept_ID)
```

- Drop DB or Tables.

```
Drop Database Database_Name ;

Drop Table Table_Name ;
```

- Truncate Remove all records from a table.

```
Truncate Table Table_Name ;
```

```
Rename Database Old_Database_Name To New_Database_Name ;
```

- Alter the structure of the DB .

```
Alter Table Table_Name
Add   New_Column   DataType   Constraints ;

EX: Alter Table  Employees
    Add   Email  varchar(50) null ;

Alter Table Table_Name
Alter Column_Name   DataType   Constraints ;

Alter Table Table_Name
Modify  Column_Name   DataType   Constraints ;

Alter Table Table_Name
Drop Column Column_Name ;

EX: Alter Table Employees
Drop Column Home_Phone ;

Alter Table Old_Table_Name
Rename to New_Table_Name ;
```

```
Alter Table Table_Name
Add   Constraint   Constraint_Name   SQL_Constraint (Column) ;

EX: Alter Table  Students
    Add   Constraint   U_Phone   Unique (Phone);

    Alter Table  Students
    Add   Constraint   Ch_GPA   Check (GPA >= 0 and GPA <= 4);

    Alter Table  Students
    Add Constraint D_Nationality Default ('Egyptian') for (Nationality);
```

Data Manipulation Language (DML)

```
Insert Into Table_Name (Column1, Column2, Column3, ….)
Values (Value1, Value2, Value3, ….) ,
          (Value1, Value2, Value3, ….) ,
          (Value1, Value2, Value3, ….) ;

Insert Into Table_Name
Values (Value1, Value2, Value3, ….) ;
```

```
Insert Into Employees
Values (1 , 'Omar', 3000 , 'Sales' , '+0100' , 'المعادى') ,
          (2 , 'Amal', 5000 , 'Marketing' , Null , 'المهندسين') ,
          (3 , 'Mona', 6000 , 'Services' , '+0111' , 'حلوان') ,
          (4 , 'Hamza', 3500 ,'Sales' , '+0122' , 'شبرا') ,
          (5 , 'Ali', 4000 , 'Services' , Null , 'الزمالك') ,
          (6 , 'Alia', 3000 , 'Sales' , '+0115' , 'م. نصر') ,
          (7 , 'Zeinab', 7000 , 'Finance' , '+0116' , 'الهرم') ,
          (8 , 'Aly', 4500 , 'Marketing' , Null , 'رمسيس') ;
```

```
Delete From Table_Name ;    =    Truncate Table Table_Name ;

Delete * From Table_Name ;

Delete From Table_Name
Where Condition ;

EX: Delete From Employees
      Where Emp_ID = 333 ;
```

```
Update Table_Name Set Column = Value ;

EX: Update Employees Set Salary = Salary + (Salary * 0.07) ;

Update Table_Name Set Column = Value
Where Condition ;

EX: Update Employees Set Salary = 10000
      Where Emp_ID = 333 ;
```

Data Query Language (DQL)

```
Select *
From Table_Name ;

Select  Column_Name(s)
From Table_Name ;
```

```
Select *
From Employees ;
```

Emp_ID	Emp_Name	Salary	FUN	Phone	Address
1	Omar	3000	Sales	+0100	المعادى
2	Amal	5000	Marketing	Null	المهندسين
3	Mona	6000	Services	+0111	حلوان
4	Hamza	3500	Sales	+0122	شبرا
5	Ali	4000	Services	Null	الزمالك
6	Alia	3000	Sales	+0155	م. نصر
7	Zeinab	7000	Finance	+0166	الهرم
8	Aly	4500	Marketing	Null	رمسيس

```
Select  Emp_ID , Emp_Name , Salary
From Employees ;
```

Emp_ID	Emp_Name	Salary
1	Omar	3000
2	Amal	5000
3	Mona	6000
4	Hamza	3500
5	Ali	4000
6	Alia	3000
7	Zeinab	7000
8	Aly	4500

```
Select *
From Table_Name
Where Condition ;
```

Comparison Operators	Logical Operators	Other Operators
=	And	IN
> , <	1 && 1 = 1 1 && 0 = 0 0 && 1 = 0 0 && 0 = 0	Like / Not Like
>= , <=	OR	Is Null / Is Not Null
<> or !=	1 || 1 = 1 1 || 0 = 1 0 || 1 = 1 0 || 0 = 0	Between
	Not	

```
Select Emp_ID , Emp_Name , Salary
From Employees
Where Emp_ID = 3 ;
```

Emp_ID	Emp_Name	Salary
3	Mona	6000

```
Select  *
From Employees
Where Salary >= 5000 ;
```

Emp_ID	Emp_Name	Salary	FUN	Phone	Address
2	Amal	5000	Marketing	Null	المهندسين
3	Mona	6000	Services	+0111	حلوان
7	Zeinab	7000	Finance	+0166	الهرم

```
Select Emp_ID , Emp_Name , Salary
From Employees
Where Emp_ID = 4  And  Emp_Name = ‘Hamza’ ;
```

Emp_ID	Emp_Name	Salary
4	Hamza	3500

```
Select Emp_ID , Emp_Name , Salary
From Employees
Where Emp_ID = 5  And  Salary > 5000 ;
```

Emp_ID	Emp_Name	Salary
No Results		

```
Select Emp_ID , Emp_Name , Salary
From Employees
Where Emp_ID = 5  OR  Salary >= 5000 ;
```

Emp_ID	Emp_Name	Salary
5	Ali	4000
2	Amal	5000
3	Mona	6000
7	Zeinab	7000

```
Select  *
From Employees
Where Not Nationality = ‘Egyptian’ ;
```

```
Select  *
From Employees
Where Nationality != ‘Egyptian’ ;
```

```
Select Emp_ID , Emp_Name , Salary
From Employees
Where Emp_ID IN  (1 , 3 , 5) ;
```

Emp_ID	Emp_Name	Salary
1	Omar	3000
3	Mona	6000
5	Ali	4000

```
Select *
From Employees
Where Emp_Name IN  (‘Ali’ , ‘Omar’ , ‘Amal’) ;
```

Emp_ID	Emp_Name	Salary	FUN	Phone	Address
5	Ali	4000	Services	Null	الزمالك
1	Omar	3000	Sales	+0100	المعادى
2	Amal	5000	Marketing	Null	المهندسين

```
Select  *
From Employees
Where Emp_Name Like  ‘A%’ ;
```

Emp_ID	Emp_Name	Salary	FUN	Phone	Address
2	Amal	5000	Marketing	Null	المهندسين
5	Ali	4000	Services	Null	الزمالك
6	Alia	3000	Sales	+0155	م. نصر
8	Aly	4500	Marketing	Null	رمسيس

```
Select  *
From Employees
Where Emp_Name Like  ‘Z%B’ ;
```

Emp_ID	Emp_Name	Salary	FUN	Phone	Address
7	Zeinab	7000	Finance	+0166	الهرم

```
Select  *
From Employees
Where Emp_Name Like  ‘Al_’ ;
```

Emp_ID	Emp_Name	Salary	FUN	Phone	Address
5	Ali	4000	Services	Null	الزمالك
8	Aly	4500	Marketing	Null	رمسيس

```
Where Nationality Not Like ‘Egyptian’ ;
```

```
Select Emp_Name , FUN , Address
From Employees
Where Phone Is Null ;
```

Emp_Name	FUN	Address
Amal	Marketing	المهندسين
Ali	Services	الزمالك
Aly	Marketing	رمسيس

```
Select Emp_Name , FUN , Phone , Address
From Employees
Where Phone Is Not Null ;
```

Emp_Name	FUN	Phone	Address
Omar	Sales	+0100	المعادى
Mona	Services	+0111	حلوان
Hamza	Sales	+0122	شبرا
Alia	Sales	+0155	م. نصر
Zeinab	Finance	+0166	الهرم

```
Select Emp_ID , Emp_Name , Salary
From Employees
Where Salary Between 4000 And 6000 ;
```

Emp_ID	Emp_Name	Salary
5	Ali	4000
8	Aly	4500
2	Amal	5000
3	Mona	6000

```
Select St_ID  ,  St_Name  ,  GPA
From Students
Where GPA Between 2.8 And 3.5 ;
```

Functions (Max-Min-Sum-Avg-Count- Distinct)

```
Select Function (Column)
From Table_Name ;
```

```
Select Max (Salary)
From Employees ;
```

No Column Name
7000

```
Select Min (Salary) as Min_Salary
From Employees ;
```

Min_Salary
3000

```
Select Sum (Salary) as Sum_Salary
From Employees ;
```

Sum_Salary
36000

```
Select Max (Salary) as Max , Min (Salary) as Min , Avg (Salary) as Average
From Employees ;
```

Max	Min	Average
7000	3000	4500

```
Select Count (*) as Count_Record
From Employees ;
```

Count_Record
8

```
Select Count (Phone) as Count_Phone
From Employees ;
```

Count_Phone
5

```
Select Distinct (FUN) as Functions
From Employees ;
```

Functions
Finance
Marketing
Sales
Services

```
Select Distinct (Salary)
From Employees ;
```

```
Select Max (GPA) as First_One
From Students ;
```

Group By (Classification)

```
Select FUN , Max (Salary) as Max_Salary , Min (Salary) as Min_Salary , Avg (Salary) as Average
From Employees
Group By FUN ;
```

FUN	Max_Salary	Min_Salary	Average
Finance	7000	7000	7000
Marketing	5000	4500	4750
Sales	3500	3000	3166.67
Services	6000	4000	5000

```
Select FUN , Max (Salary) as Max_Salary , Min (Salary) as Min_Salary , Avg (Salary) as Average
From Employees
Group By FUN
Having Avg (Salary) >= 5000 ;
```

FUN	Max_Salary	Min_Salary	Average
Finance	7000	7000	7000
Services	6000	4000	5000

```
Select FUN , Max (Salary) as Max_Salary , Min (Salary) as Min_Salary , Avg (Salary) as Average
From Employees
Where Emp_ID IN  (1 , 2 , 3 , 4 , 5 )
Group By FUN
Having Avg (Salary) >= 5000 ;
```

FUN	Max_Salary	Min_Salary	Average
Services	6000	4000	5000
Marketing	5000	5000	5000

Order By (Sorting)

```
Select Emp_ID , Emp_Name , Salary , FUN
From Employees
Order By Salary ASC ;
```

Emp_ID	Emp_Name	Salary	FUN
1	Omar	3000	Sales
6	Alia	3000	Sales
4	Hamza	3500	Sales
5	Ali	4000	Services
8	Aly	4500	Marketing
2	Amal	5000	Marketing
3	Mona	6000	Services
7	Zeinab	7000	Finance

```
Select Emp_ID , Emp_Name , Salary , FUN
From Employees
Order By Salary DESC , Emp_Name ASC ;
```

Emp_ID	Emp_Name	Salary	FUN
7	**Zeinab**	**7000**	Finance
3	Mona	6000	Services
2	Amal	5000	Marketing
8	Aly	4500	Marketing
5	Ali	4000	Services
4	Hamza	3500	Sales
6	**Alia**	**3000**	Sales
1	**Omar**	**3000**	Sales

```
Select St_ID , St_Name , GPA
From Students
Order By GPA DESC , St_Name ASC ;
```

Join

Customers

Cus_ID	Cus_Name
1	Ahmed
2	Omar
3	Mona
4	Hamza

Orders

Ord_ID	Product	Cus_ID
234	Printer	1
567	Camera	2
456	Mobile	2
876	Scanner	3

```
Create Database Techno_Store ;

Create Table Customers
(
Cus_ID     int     Primary Key ,
Cus_Name     varchar (50)     Not Null ,
) ;


Create Table Orders
(
Ord_ID     int     Primary Key ,
Product     varchar (50)     Not Null ,
Foreign Key (Cus_ID)   References   Customers (Cus_ID)
) ;
```

Join using where

```
Select Customers.Cus_ID , Customers.Cus_Name , Orders.Product
From Customers , Orders
Where Customers.Cus_ID = Orders.Cus_ID ;
```

Cus_ID	Cus_Name	Product
1	Ahmed	Printer
2	Omar	Camera
2	Omar	Mobile
3	Mona	Scanner

INNER Join

```
Select Customers.Cus_Name , Orders.Product
From Customers INNER Join Orders
ON Customers.Cus_ID = Orders.Cus_ID ;
```

Cus_Name	Product
Ahmed	Printer
Omar	Camera
Omar	Mobile
Mona	Scanner

Left Join

```
Select Customers.Cus_Name , Orders.Product
From Customers Left Join Orders
ON Customers.Cus_ID = Orders.Cus_ID;
```

Cus_Name	Product
Ahmed	Printer
Omar	Camera
Omar	Mobile
Mona	Scanner
Hamza	

Right Join

```
Select Customers.Cus_Name , Orders.Product
From Customers Right Join Orders
ON Customers.Cus_ID = Orders.Cus_ID;
```

Cus_Name	Product
Ahmed	Printer
Omar	Camera
Omar	Mobile
Mona	Scanner