Assignment:

Introduction to SQL Server

Problem Statement:

Consider yourself to be Sam who is a student at a prestigious university. You have enrolled for the SQL course and it is your first semester.

Tasks To Be Performed:

1. Install MS SQL Server

Step 1: Download SQL Server Installation Media

- Go to the official Microsoft SQL Server download page.
- Choose the edition and version of SQL Server that you want to install.
- Download the installation media. This may be an ISO file or an executable file, depending on the version.

Step 2: Run the SQL Server Installation Wizard

- Locate the downloaded installation media and run the setup executable.
- The SQL Server Installation Center will open. Select "New SQL Server stand-alone installation or add features to an existing installation."

Step 3: Choose Installation Type

- The wizard will check for prerequisites. If any are missing, you will be prompted to install them.
- Choose the installation type. Select "New SQL Server stand-alone installation."

Step 4: Accept License Terms

- Read and accept the license terms.

Step 5: Feature Selection

- Select the SQL Server features you want to install. Choose components like Database Engine Services, SQL Server Replication, Full-Text and Semantic Extractions, etc., based on your requirements.

Step 6: Instance Configuration

- Specify the instance name. The default instance is usually named "MSSQLSERVER."
- Choose the instance root directory where SQL Server will be installed.

Step 7: Server Configuration

- Configure the SQL Server service accounts and collation settings.
- Set authentication mode. Choose between Windows Authentication and Mixed Mode (SQL Server and Windows Authentication). If Mixed Mode is selected, set a strong password for the 'sa' account.

Step 8: Database Engine Configuration

- Configure server authentication, data directories, and other settings based on your requirements.

Step 9: Reporting Services Configuration (if applicable)

- If you're installing Reporting Services, configure the Report Server mode and specify the Report Server Database settings.

Step 10: Installation

- Review the summary and click "Install" to begin the installation process.

Step 11: Complete the Installation

- Once the installation completes, you'll see a summary. Review any messages to ensure the installation was successful.
- Click "Next" and then "Close" to complete the installation.

2. Give the difference between Char and Varchar data type.

The primary difference between CHAR and VARCHAR data types lies in how they store and handle character data:

Fixed-Length vs. Variable-Length:

- CHAR is a fixed-length character data type, meaning it always occupies a specific, fixed amount of storage regardless of the actual length of the data. If you define a CHAR(10), it will always use 10 bytes of storage, padding with spaces if the actual data is shorter.
- VARCHAR is a variable-length character data type. It only uses storage equivalent to the actual length of the data. For example, if you define a VARCHAR(10) and store "Hello," it will use only 5 bytes of storage.

Storage Efficiency:

- VARCHAR is generally more storage-efficient than CHAR for variable-length data because it doesn't waste space on padding with trailing spaces.
- CHAR may be more suitable when the data length is consistent because it avoids the overhead of storing the length information associated with variable-length data types.

Trailing Spaces:

- In CHAR, if the stored data is shorter than the defined length, it is padded with trailing spaces.
- In VARCHAR, trailing spaces are not padded. The actual data length is what is stored.

Performance Considerations:

- Retrieving data from CHAR columns can be slightly faster than from VARCHAR columns in some scenarios, especially when dealing with fixed-length data.
- However, VARCHAR may be more efficient when dealing with variable-length data due to its storage flexibility.

Use Cases:

- Use CHAR when the length of the data is consistent and fixed, such as storing codes or identifiers.
- Use VARCHAR when the length of the data varies, and you want to save storage space for variable-length strings.

Example:

```
Create Table tbl3(Name char(30))

insert into tbl3 Values('दोस्त'),('Freundin'),('朋友')

Select * From tbl3
```

	Name
1	?????
2	Freundin
3	??

```
Create Table tbl4(Name varchar(30))

insert into tbl4 Values('Friend'),('Freundin'),('朋友')

Select * From tbl4
```

	Name
1	Friend
2	Freundin
3	??

3. Explain the types of SQL Commands.

SQL (Structured Query Language) commands are categorized into several types based on their functionality. Here are the main types of SQL commands:

DDL (Data Definition Lang

- DDL commands are used for defining or altering the structure of a database.
- Key DDL commands include:
  - CREATE: Used to create database objects like tables, indexes, or views.
  - ALTER: Used to modify the structure of an existing database object.
  - DROP: Used to delete an existing database object.

DML (Data Manipulation Language):

- DML commands are used for manipulating data stored in a database.
- Key DML commands include:
  - SELECT: Used to retrieve data from one or more tables.
  - INSERT: Used to insert new records into a table.
  - UPDATE: Used to modify existing records in a table.
  - DELETE: Used to remove records from a table.

DCL (Data Control Language):

- DCL commands are used to control access to data within a database.
- Key DCL commands include:
    - GRANT: Gives specific privileges to users or roles.
    - REVOKE: Removes specific privileges from users or roles.

TCL (Transaction Control Language):

- TCL commands are used to manage transactions within a database.
- Key TCL commands include:
    - COMMIT: Saves all changes made during the current transaction.
    - ROLLBACK: Undoes changes made during the current transaction.
    - SAVEPOINT: Sets a point within a transaction to which you can later roll back.
    - SET TRANSACTION: Specifies characteristics for the transaction.

4. Explain NVarchar and Nchar

NVARCHAR and NCHAR are both Unicode character data types in SQL Server. They are used to store variable-length and fixed-length Unicode character strings, respectively. Unicode is a character encoding standard that supports a wide range of characters, including those from various languages and symbols.

- **NVARCHAR (Variable-Length Unicode Character String):**
    - NVARCHAR stands for National Variable Character.
    - It is used to store variable-length Unicode character strings.
    - Each character in an NVARCHAR column requires 2 bytes of storage.
    - It can store characters from different languages, as well as symbols and emojis.
    - The length of an NVARCHAR column is defined in terms of the number of characters it can hold, not the number of bytes.

```
Create Table tb1(Name Nvarchar(30))

insert into tb1 Values(N'दोस्त'),(N'Freundin'),(N'朋友')

Select * From tb1
```

	Name
1	दोस्त
2	Freundin
3	朋友

- **NCHAR (Fixed-Length Unicode Character String):**
    - NCHAR stands for National Character.
    - It is used to store fixed-length Unicode character strings.
    - Each character in an NCHAR column requires 2 bytes of storage, and it always reserves space for the maximum defined length.
    - Similar to NVARCHAR, it can store characters from different languages and symbols.

```
Create Table tbl2(Name Nchar(30))

insert into tbl2 Values(N'दोस्त'),(N'Freundin'),(N'朋友')

Select * From tbl2
```

	Name
1	दोस्त
2	Freundin
3	朋友

Key Differences:

Storage Size:

- Both NVARCHAR and NCHAR use 2 bytes per character, but NCHAR always reserves space for the maximum defined length, leading to potential storage wastage for shorter strings.

Length Definition:

- The length for NVARCHAR is defined in terms of the number of characters, while for NCHAR, it is defined in terms of the number of characters that can be stored.

Flexibility:

- NVARCHAR is more flexible for storing variable-length strings, whereas NCHAR is used when a fixed-length string is required.

Use Cases:

- Use NVARCHAR when the length of the string can vary, and you want to optimize storage.
- Use NCHAR when a fixed-length string is needed, even if the actual data may be shorter.

In summary, choose between NVARCHAR and NCHAR based on whether you need variable-length or fixed-length Unicode character strings and consider the potential storage implications.

Assignment 2

Clauses And Wildcards

Problem Statement:

You have successfully cleared the first semester. In your second semester you will learn how to create tables, work with WHERE clause and basic operators.

Tasks To Be Performed:

1. Create a customer table which comprises of these columns: 'customer_id', 'first_name', 'last_name', 'email', 'address', 'city','state','zip'

- ```
  Create Table Customer(customer_id int identity(1,1),first_name varchar(20),
  last_name Varchar(20),email Varchar(50),address Varchar(50),city Varchar(20),
  state Varchar(20),zip Varchar(20))
  ```

2. Insert 5 new records into the table

- ```
  insert into Customer (first_name,last_name,email,address,city,state,zip)
  Values('Ayush','Lal','ayushlal@gmail.com','Delhi','Delhi','Delhi','110037'),
  ('Shweta','Gupta','shwetagupta@gmail.com','Jaipur','Jaipur','Jaipur','110035'),
  ('Vaishnav','TK','gmail','Kerala','Kerala','Kerala','364135'),
  ('George','Mathew','mathewgeorge@gmail.com','SanJose','SanJose','California','9
  40088'),
  ('Raghisha','Prasannan','gmail','Kozhikode','Kerala','Kerala','673001')
  ```

	customer_id	first_name	last_name	email	address	city	state	zip
1	1	Ayush	Lal	ayushlal@gmail.com	Delhi	Delhi	Delhi	110037
2	2	Shweta	Gupta	shwetagupta@gmail.com	Jaipur	Jaipur	Jaipur	110035
3	3	Vaishnav	TK	gmail	Kerala	Kerala	Kerala	364135
4	4	George	Mathew	mathewgeorge@gmail.com	SanJose	SanJose	California	940088
5	5	Raghisha	Prasannan	gmail	Kozhikode	Kerala	Kerala	673001

3. Select only the 'first_name' and 'last_name' columns from the customer table

- ```
  Select first_name,last_name From Customer
  ```

	first_name	last_name
1	Ayush	Lal
2	Shweta	Gupta
3	Vaishnav	TK
4	George	Mathew
5	Raghisha	Prasannan

4. Select those records where 'first_name' starts with "G" and city is 'SanJose'.

- ```
  Select * From Customer
  Where first_name Like 'G%' And City = 'SanJose'
  ```

	customer_id	first_name	last_name	email	address	city	state	zip
1	4	George	Mathew	mathewgeorge@gmail.com	SanJose	SanJose	California	940088

5. Select those records where Email has only ‘gmail’.

- ```
  Select * From Customer
  Where email = 'gmail'
  ```

	customer_id	first_name	last_name	email	address	city	state	zip
1	3	Vaishnav	TK	gmail	Kerala	Kerala	Kerala	364135
2	5	Raghisha	Prasannan	gmail	Kozhikode	Kerala	Kerala	673001

6. Select those records where the ‘last_name’ doesn't end with “A”.

- ```
  Select * From Customer
  Where last_name not like '%A'
  ```

	customer_id	first_name	last_name	email	address	city	state	zip
1	1	Ayush	Lal	ayushlal@gmail.com	Delhi	Delhi	Delhi	110037
2	3	Vaishnav	TK	gmail	Kerala	Kerala	Kerala	364135
3	4	George	Mathew	mathewgeorge@gmail.com	SanJose	SanJose	California	940088
4	5	Raghisha	Prasannan	gmail	Kozhikode	Kerala	Kerala	673001

Assignment 3

Different Types of Joins

Problem Statement:

You have successfully cleared the second semester. In your third semester youwill work with joins and update statements.

Tasks To Be Performed:

1. Create an 'Orders' table which comprises of these columns: 'order_id', 'order_date', 'amount', 'customer_id'.

- ```
  Create Table Orders(order_id int,order_date date,amount float,customer_id int)
  ```

2. Insert 5 new records.

- ```
  Insert into Orders Values(54,'2024-01-01',200,1),
  (65,'2024-01-10',500,2),(23,'2024-01-15',150,3),(15,'2024-01-20',2000,6),
  (30,'2024-01-30',550,7)
  ```

	order_id	order_date	amount	customer_id
1	54	2024-01-01	200	1
2	65	2024-01-10	500	2
3	23	2024-01-15	150	3
4	15	2024-01-20	2000	6
5	30	2024-01-30	550	7

3. Make an inner join on 'Customer' and 'Orders' tables on the 'customer_id' column.

Table Customer

	customer_id	first_name	last_name	email	address	city	state	zip
1	1	Ayush	Lal	ayushlal@gmail.com	Delhi	Delhi	Delhi	110037
2	2	Shweta	Gupta	shwetagupta@gmail.com	Jaipur	Jaipur	Jaipur	110035
3	3	Vaishnav	TK	gmail	Kerala	Kerala	Kerala	364135
4	4	George	Mathew	mathewgeorge@gmail.com	SanJose	SanJose	California	940088
5	5	Raghisha	Prasannan	gmail	Kozhikode	Kerala	Kerala	673001

Table Orders

	order_id	order_date	amount	customer_id
1	54	2024-01-01	200	1
2	65	2024-01-10	500	2
3	23	2024-01-15	150	3
4	15	2024-01-20	2000	6
5	30	2024-01-30	550	7

- ```
  Select * From Customer As C
  Join Orders As O
  On C.customer_id = O.customer_id
  ```
  ```

	customer_id	first_name	last_name	email	address	city	state	zip	order_id	order_date	amount	custom
1	1	Ayush	Lal	ayushlal@gmail.com	Delhi	Delhi	Delhi	110037	54	2024-01-01	200	1
2	2	Shweta	Gupta	shwetagupta@gmail.com	Jaipur	Jaipur	Jaipur	110035	65	2024-01-10	500	2
3	3	Vaishnav	TK	gmail	Kerala	Kerala	Kerala	364135	23	2024-01-15	150	3

4. Make left and right joins on 'Customer' and 'Orders' tables on the 'customer_id' column.

- ```
  Select * From Customer As C
  Left Join Orders As O
  On C.customer_id = O.customer_id
  ```

	customer_id	first_name	last_name	email	address	city	state	zip	order_id	order_date	amount
1	1	Ayush	Lal	ayushlal@gmail.com	Delhi	Delhi	Delhi	110037	54	2024-01-01	200
2	2	Shweta	Gupta	shwetagupta@gmail.com	Jaipur	Jaipur	Jaipur	110035	65	2024-01-10	500
3	3	Vaishnav	TK	gmail	Kerala	Kerala	Kerala	364135	23	2024-01-15	150
4	4	George	Mathew	mathewgeorge@gmail.com	SanJose	SanJose	California	940088	NULL	NULL	NULL
5	5	Raghisha	Prasannan	gmail	Kozhikode	Kerala	Kerala	673001	NULL	NULL	NULL

- ```
  Select * From Customer As C
  Right Join Orders As O
  On C.customer_id = O.customer_id
  ```

	customer_id	first_name	last_name	email	address	city	state	zip	order_id	order_date	amount	custome
1	1	Ayush	Lal	ayushlal@gmail.com	Delhi	Delhi	Delhi	110037	54	2024-01-01	200	1
2	2	Shweta	Gupta	shwetagupta@gmail.com	Jaipur	Jaipur	Jaipur	110035	65	2024-01-10	500	2
3	3	Vaishnav	TK	gmail	Kerala	Kerala	Kerala	364135	23	2024-01-15	150	3
4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	15	2024-01-20	2000	6
5	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	30	2024-01-30	550	7

5. Make a full outer join on 'Customer' and 'Orders' table on the 'customer_id' column.

- ```
  Select * From Customer As C
  Full Outer Join Orders As O
  On C.customer_id = O.customer_id
  ```

	customer_id	first_name	last_name	email	address	city	state	zip	order_id	order_date	amount
1	1	Ayush	Lal	ayushlal@gmail.com	Delhi	Delhi	Delhi	110037	54	2024-01-01	200
2	2	Shweta	Gupta	shwetagupta@gmail.com	Jaipur	Jaipur	Jaipur	110035	65	2024-01-10	500
3	3	Vaishnav	TK	gmail	Kerala	Kerala	Kerala	364135	23	2024-01-15	150
4	4	George	Mathew	mathewgeorge@gmail.com	SanJose	SanJose	California	940088	NULL	NULL	NULL
5	5	Raghisha	Prasannan	gmail	Kozhikode	Kerala	Kerala	673001	NULL	NULL	NULL
6	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	15	2024-01-20	2000
7	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	30	2024-01-30	550

6. Update the 'Orders' table and set the amount to be 100 where 'customer_id' is 3.

- ```
  Update Orders Set amount = 100 Where customer_id = 3
  ```

  - ```
    Select * From Orders Where customer_id = 3
    ```

	order_id	order_date	amount	customer_id
1	23	2024-01-15	100	3
    ```

Assignment 4

Different Types of Functions

Problem Statement:

You have successfully cleared your third semester. In the fourth semester you will work with inbuilt functions and user-defined functions.

Tasks To Be Performed:

1. Use the inbuilt functions and find the minimum, maximum and average amount from the orders table

Order Table

- `Select * From Orders`

	order_id	order_date	amount	customer_id
1	54	2024-01-01	200	1
2	65	2024-01-10	500	2
3	23	2024-01-15	100	3
4	15	2024-01-20	2000	6
5	30	2024-01-30	550	7

- `Select Max(Amount) As [Maximum Amount] From Orders`
- `Select Min(Amount) As [Minimum Amount] From Orders`
- `Select Avg(Amount) As [Average Amount] From Orders`

	Maximum Amount
1	2000

	Minimum Amount
1	100

	Average Amount
1	670

2. Create a user-defined function which will multiply the given number with10

```
Create Function fn_mul(@input float)
Returns float
As Begin
Declare @result float
Set @result = @input * 10
Return @result
End

select dbo.fn_mul(20) As Answer
```

	Answer
1	200

3. Use the case statement to check if 100 is less than 200, greater than200or equal to 200 and print the corresponding value.

- ```
  Declare @val int,@result varchar(50)
  Set @val = 100
  Set @result = Case When @val < 200 Then '100 Is Less Than 200'
  When @val > 200 Then '100 Is Greater Than 200'
  When @val = 200 Then '100 Is Eqaul To 200'
  End
  Print @result
  ```

```
Messages
   100 Is Less Than 200

   Completion time: 2024-02-03T14:13:18.2466048+05:30
```

4. Using a case statement, find the status of the amount. Set the status of the amount as high amount, low amount or medium amount based upon the condition.

- ```
  Select *,
  Case When amount > 1000 Then 'High Amount'
  When amount >= 500 Then 'Medium Amount'
  Else 'Low Amount'
  End As [Amount Status]
  From Orders
  ```

	order_id	order_date	amount	customer_id	Amount Status
1	54	2024-01-01	200	1	Low Amount
2	65	2024-01-10	500	2	Medium Amount
3	23	2024-01-15	100	3	Low Amount
4	15	2024-01-20	2000	6	High Amount
5	30	2024-01-30	550	7	Medium Amount

5. Create a user-defined function, to fetch the amount greater than then given input.

- ```
  Create Function fn_greater(@input float)
  Returns Table
  As
  Return
  (Select order_id,amount From Orders
  Where amount > @input)

  Select * From fn_greater(500)
  ```

	order_id	amount
1	15	2000
2	30	550
  ```

Assignment 5

Different Types of Operators

Problem Statement:

You have successfully cleared your fourth semester. In the fifth semester you will work with clauses and SET operators.

Tasks To Be Performed:

1. Arrange the 'Orders' dataset in decreasing order of amount

Orders Table

	order_id	order_date	amount	customer_id
1	54	2024-01-01	200	1
2	65	2024-01-10	500	2
3	23	2024-01-15	100	3
4	15	2024-01-20	2000	6
5	30	2024-01-30	550	7

- ```
  Select * From Orders Order By amount Desc
  ```

	order_id	order_date	amount	customer_id
1	15	2024-01-20	2000	6
2	30	2024-01-30	550	7
3	65	2024-01-10	500	2
4	54	2024-01-01	200	1
5	23	2024-01-15	100	3

2. Create a table with the name 'Employee_details1' consisting of these columns: 'Emp_id', 'Emp_name', 'Emp_salary'. Create another table with the name 'Employee_details2' consisting of the same columns as the first table.

- ```
  Create Table Employee_details1(Emp_id int,Emp_name Varchar(30),Emp_Salary int)
  ```
  - ```
    Insert into Employee_details1 Values(1,'Ayush',50000),
    (2,'Shweta',100000),(3,'Samual',55000),(4,'Vaishnav',80000),(5,'Ashok',6
    5000)
    ```
  - ```
    Select * From Employee_details1
    ```

	Emp_id	Emp_name	Emp_Salary
1	1	Ayush	50000
2	2	Shweta	100000
3	3	Samual	55000
4	4	Vaishnav	80000
5	5	Ashok	65000

- ```
  Create Table Employee_details2(Emp_id int,Emp_name Varchar(30),Emp_Salary int)
  ```
    - ```
      Insert Into Employee_details2 Values(1,'Ayush',50000),
      (2,'Shweta',100000),(6,'Anshita',75000),(7,'Shalini',85000),(5,'Ashok',6
      5000)
      ```
    - ```
      Select * From Employee_details2
      ```

	Emp_id	Emp_name	Emp_Salary
1	1	Ayush	50000
2	2	Shweta	100000
3	6	Anshita	75000
4	7	Shalini	85000
5	5	Ashok	65000

3. Apply the UNION operator on these two tables

- ```
  Select * From Employee_details1
  Union
  Select * From Employee_details2
  ```

	Emp_id	Emp_name	Emp_Salary
1	1	Ayush	50000
2	2	Shweta	100000
3	3	Samual	55000
4	4	Vaishnav	80000
5	5	Ashok	65000
6	6	Anshita	75000
7	7	Shalini	85000

4. Apply the INTERSECT operator on these two tables

- ```
  Select * From Employee_details1
  Intersect
  Select * From Employee_details2
  ```

	Emp_id	Emp_name	Emp_Salary
1	1	Ayush	50000
2	2	Shweta	100000
3	5	Ashok	65000

5. Apply the EXCEPT operator on these two tables

- ```
  Select * From Employee_details1
  Except
  Select * From Employee_details2
  ```

	Emp_id	Emp_name	Emp_Salary
1	3	Samual	55000
2	4	Vaishnav	80000

Assignment 6

Transaction And Exception Handling In SQL

Problem Statement:

You have successfully cleared your fifth semester. In the final semester you will work with views, transactions and exception handling.

Table Customer

	Customer_id	First_name	Last_Name	Email	Address	City	State	Zip
1	1	Sana	B	sana@gmail.com	Jayanagar	Bangalore	Karnataka	5877
2	2	Apurva	Wankade	apurva@yahoo.com	5th Cross	Pune	Mumbai	6894
3	3	Gautham	Sinha	gautham@yahoo.com	New City	San Jose	CA	12868
4	4	Vishal	V	vishal@gmail.com	4th Cross	Chennai	TamilNadu	6958
5	5	Bob	Barly	bob@hotmail.com	3rd Street	Texas	CA	84985

Table Orders

	order_id	order_date	amount	customer_id
1	101	2021-07-04	2450	1
2	201	2018-09-13	5670	3
3	301	2020-02-02	2000	5
4	401	2019-01-05	3500	6
5	501	2021-06-03	300	7

Tasks To Be Performed:

1. Create a view named 'customer_san_jose' which comprises of only those customers who are from San Jose

- ```
  Create View vw_customer_san_jose
  As
  Select * From Customer Where City = 'San Jose'

  Select * From vw_customer_san_jose
  ```

	Customer_id	First_name	Last_Name	Email	Address	City	State	Zip
1	3	Gautham	Sinha	gautham@yahoo.com	New City	San Jose	CA	12868

2. Inside a transaction, update the first name of the customer to Francis where the last name is Jordan:

a. Rollback the transaction

- ```
  Begin Transaction

  Update Customer Set First_name = 'Francis' Where Last_Name =
  'Jordan'

  Rollback
  ```

b. Set the first name of customer to Alex, where the last name is Jordan

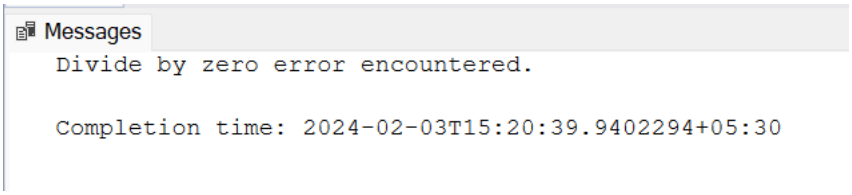
-
```
Begin Transaction

Update Customer Set First_name = 'Alex' Where Last_Name = 'Jordan'

Rollback
```

3. Inside a TRY... CATCH block, divide 100 with 0, print the default error message.

-
```
Begin Try
Print 100/0
End Try
Begin Catch
Print Error_message()
End Catch
```



4. Create a transaction to insert a new record to Orders table and save it.

-
```
Begin Transaction

Insert into Orders Values(502,'2021-07-04',600,8)

Commit

Select * From Orders
```

	order_id	order_date	amount	customer_id
1	101	2021-07-04	2450	1
2	201	2018-09-13	5670	3
3	301	2020-02-02	2000	5
4	401	2019-01-05	3500	6
5	501	2021-06-03	300	7
6	502	2021-07-04	600	8

SQL Assignment

1. CHAR(n), TEXT(n), VARCHAR(n).

Alphanumeric data either fixed at n symbols or up to n symbols. It's not possible to do arithmetic on this data.

2. REAL, FLOAT, NUMBER, NUMERIC, DECIMAL

These are numbers with decimal places

3. INTEGER, LONG, INT, SMALLINT

These are the whole numbers. They vary in how big a number they can hold

4. MONEY, CURRENCY

These are numeric types with decimal places for holding monetary values

5. BINARY, LONGBINARY, GENERAL, IMAGE, OLEOBJECT

These can hold complete files, such as pictures or media. They are not fixed in size and the upper limit on their size is large.

6. DATE, TIME, DATETIME

These can hold date and time values. These are hybrid types. Although you think of them as text type fields, it is possible to do arithmetic and numeric comparisons on them

7. Write an SQL command that will create a table named Friend with the following fields and types: id no CHAR(3), name CHAR(24), address CHAR(24), birthday DATE, waist size INTEGER, gift value CURRENCY.

- Create Table Friend(Id_no char(3),Name Char(24),Address Char(24),Birthday Date, Waist_Size Integer,Gift_Value Money)

8. Write an SQL command that will create a table named Friend with the following fields: idno, name, address, age, bank balance. Each of these fields has characteristics that would make a particular type appropriate for it. You need to choose the type and size for the fields.

- Create Table Friend(Idno char(3),Name Char(24),Address Char(24),Age int,Bank_Balance Money)

9. Write an SQL command that will put this name into a record in the Friend table: 'Road Runner'.

- insert into Friend(Name) Values('Road Runner')
- Select * From Friend

Results Messages

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	NULL	NULL	NULL

10. For the record created by the previous question, what values would be in the fields other than the name field?

Null

11. Write an SQL command that will insert a complete record into the Friend table with these values for the respective fields: '123', 'Tasmanian Devil', 'Tasmania', 07/07/57, 32, 29.99.

- Insert into Friend Values(123,'Tasmanian Devil','Tasmania','07-07-57',32,29.99)
- Select * From Friend

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	NULL	NULL	NULL
2	123	Tasmanian Devil	Tasmania	1957-07-07	32	29.99

12. Write an SQL command that will insert a complete record into the Friend table with these values for the respective fields: '456', 'Felix the Cat','Hollywood', NULL, NULL, NULL.

- Insert into Friend Values(456,'Felix the Cat','Hollywood',Null,Null,Null)
- Select * From Friend

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	NULL	NULL	NULL
2	123	Tasmanian Devil	Tasmania	1957-07-07	32	29.99
3	456	Felix the Cat	Hollywood	NULL	NULL	NULL

13. Write an SQL command (with a query included) that will insert into Friend all of the spno's, names, and addresses from the salesperson table.

Table Salesperson

- Create Table Salesperson(spnos int,Name Varchar(20),Address Varchar(30))
- insert into Salesperson Values(200,'Arshin','Goa')
- Select * From Salesperson

	spnos	Name	Address
1	200	Arshin	Goa

- Insert into Friend(Id_no,Name,Address)
- Select * From Salesperson
- Select * From Friend

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	NULL	NULL	NULL
2	123	Tasmanian Devil	Tasmania	1957-07-07	32	29.99
3	456	Felix the Cat	Hollywood	NULL	NULL	NULL
4	200	Arshin	Goa	NULL	NULL	NULL

14. Write an SQL command (with a query included) that will insert into Friend all of the custno's, names, and addresses of customers where the state is equal to 'WA'.

Customer Table

	Customer_id	First_name	Last_Name	Email	Address	City	State	Zip
1	1	Sana	B	sana@gmail.com	Jayanagar	Bangalore	Karnataka	5877
2	2	Apurva	Wankade	apurva@yahoo.com	5th Cross	Pune	Mumbai	6894
3	3	Gautham	Sinha	gautham@yahoo.com	New City	San Jose	CA	12868
4	4	Vishal	V	vishal@gmail.com	4th Cross	Chennai	TamilNadu	6958
5	5	Bob	Barly	bob@hotmail.com	3rd Street	Texas	CA	84985
6	6	Charlie	MA	ma@gmail.com	4th Street	Washington	WA	20001
7	7	Prithvi	Raj	rajprithvi@gmail.com	7th Street	Washington	WA	20001

- ```
  Insert into Friend(Id_no,Name,Address)
  Select Customer_id,First_name,Address From Customer
  Where State = 'WA'
  ```

- ```
  Select * From Friend
  ```

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	NULL	NULL	NULL
2	123	Tasmanian Devil	Tasmania	1957-07-07	32	29.99
3	456	Felix the Cat	Hollywood	NULL	NULL	NULL
4	200	Arshin	Goa	NULL	NULL	NULL
5	6	Charlie	4th Street	NULL	NULL	NULL
6	7	Prithvi	7th Street	NULL	NULL	NULL

15. Write an SQL command (with a query included) that will insert into Friend just the custno's of all of the customers where the state is equal to 'WA'.

- ```
  Insert into Friend(Id_no)
  Select Customer_id From Customer
  Where State = 'WA'
  ```

16. Write an SQL command (with a query included) that will insert intoFriend just the spno's and names of salespeople.

- ```
  Insert into Friend(Id_no,Name)
  Select spnos,Name From Salesperson
  ```

17. What's wrong with this query?

INSERT INTO Friend(idno, name, giftvalue)SELECT spno, name, commrate

FROM salesperson

It doesn't make sense to put commrate into giftvalue.

18. What's wrong with this query?

INSERT INTO Friend SELECT *

FROM Salesperson

- Select * From Salesperson

	spnos	Name	Address
1	200	Arshin	Goa

- Select * From Friend

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	NULL	NULL	NULL
2	123	Tasmanian Devil	Tasmania	1957-07-07	32	29.99
3	456	Felix the Cat	Hollywood	NULL	NULL	NULL
4	200	Arshin	Goa	NULL	NULL	NULL
5	6	Charlie	4th Street	NULL	NULL	NULL
6	7	Prithvi	7th Street	NULL	NULL	NULL

There aren't as many fields in the Friend table as there are in the Salesperson table, so it's not possible to select all from Salesperson to insert into Friend. Plus,the respective fields don't match anyway.

19. Write an SQL command that will update the giftvalue to 49.99 for all people in the Friend table who are 21 and older

Friend Table

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	1988-05-07	NULL	NULL
2	123	Tasmanian Devil	Tasmania	1957-07-07	32	29.99
3	456	Felix the Cat	Hollywood	2005-05-07	NULL	NULL
4	200	Arshin	Goa	2007-05-07	NULL	NULL
5	6	Charlie	4th Street	1970-05-07	NULL	NULL
6	7	Prithvi	7th Street	1990-05-07	NULL	NULL

- Update Friend Set Gift_Value = 49.99
  Where (Datediff(Year,Birthday,GETDATE()) -
  Case When Month(Birthday)>=Month(Getdate()) And
  Day(Birthday)>Day(GETDATE()) Then 1 Else 0 End) >=21

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	1988-05-07	NULL	49.99
2	123	Tasmanian Devil	Tasmania	1957-07-07	32	49.99
3	456	Felix the Cat	Hollywood	2005-05-07	NULL	NULL
4	200	Arshin	Goa	2007-05-07	NULL	NULL
5	6	Charlie	4th Street	1970-05-07	NULL	49.99
6	7	Prithvi	7th Street	1990-05-07	NULL	49.99

---

20. Write an SQL command that will update the Friend table so that all waist sizes will be 0.

- Update Friend Set Waist_Size = 0

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	1988-05-07	0	49.99
2	123	Tasmanian Devil	Tasmania	1957-07-07	0	49.99
3	456	Felix the Cat	Hollywood	2005-05-07	0	NULL
4	200	Arshin	Goa	2007-05-07	0	NULL
5	6	Charlie	4th Street	1970-05-07	0	49.99
6	7	Prithvi	7th Street	1990-05-07	0	49.99

21. Write an SQL command that will add a city field to the Friend table. Youcan choose its type and size.

- Alter Table Friend Add City Varchar(100)

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value	City
1	NULL	Road Runner	NULL	1988-05-07	0	49.99	NULL
2	123	Tasmanian Devil	Tasmania	1957-07-07	0	49.99	NULL
3	456	Felix the Cat	Hollywood	2005-05-07	0	NULL	NULL
4	200	Arshin	Goa	2007-05-07	0	NULL	NULL
5	6	Charlie	4th Street	1970-05-07	0	49.99	NULL
6	7	Prithvi	7th Street	1990-05-07	0	49.99	NULL

22. Write an SQL command that will change the specifications of the name field in the Friend table to CHAR(36).

- Alter Table Friend Alter Column Name varchar(36)
- sp_help 'Friend'

2	Name	varchar	no	36

23. Write an SQL command that will get rid of the city field from the Friendtable.

- Alter Table Friend Drop Column City

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	1988-05-07	0	49.99
2	123	Tasmanian Devil	Tasmania	1957-07-07	0	49.99
3	456	Felix the Cat	Hollywood	2005-05-07	0	NULL
4	200	Arshin	Goa	2007-05-07	0	NULL
5	6	Charlie	4th Street	1970-05-07	0	49.99
6	7	Prithvi	7th Street	1990-05-07	0	49.99

24. Write an SQL command that will delete all records from the Friend table where the giftvalue is over 100.

Friend Table

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	1988-05-07	0	49.99
2	123	Tasmanian Devil	Tasmania	1957-07-07	0	49.99
3	456	Felix the Cat	Hollywood	2005-05-07	0	NULL
4	200	Arshin	Goa	2007-05-07	0	NULL
5	6	Charlie	4th Street	1970-05-07	0	49.99
6	7	Prithvi	7th Street	1990-05-07	0	120.00

- Delete From Friend Where Gift_Value > 100

	Id_no	Name	Address	Birthday	Waist_Size	Gift_Value
1	NULL	Road Runner	NULL	1988-05-07	0	49.99
2	123	Tasmanian Devil	Tasmania	1957-07-07	0	49.99
3	456	Felix the Cat	Hollywood	2005-05-07	0	NULL
4	200	Arshin	Goa	2007-05-07	0	NULL
5	6	Charlie	4th Street	1970-05-07	0	49.99

---

25. Write an SQL command that will drop the table Friend from a database,eliminating it and all of its records

- Drop Table Friend

26.

A) What distinctive punctuation marks should be used to enclose date values in queries?

B) What different kinds of punctuation marks can be used inside date values in order to separate month, day, and year?

A) single quotes (').

B) Dashes and slashes are the correct separators for dates when you want to enter values into a table or give a specific value for comparisonin a WHERE clause, for example. It is possible to print out dates with various punctuation marks when doing formatting.

27. Write a query that will select all fields from the Carsale table where the salesdate is after January 1st, 2006.

Carsale Table

	Id	Model	Salesdate	Purchase_Amount
1	1	Lamborghini	2001-01-14	30000000.00
2	2	Ferrari	2010-02-10	35000000.00
3	3	Ferrari Porsche	2011-10-17	59000000.00
4	4	Lamborghini Huracan	2015-05-10	79000000.00
5	5	Maruti Alto K10	2024-03-19	600000.00
6	6	Renault KWID	2023-03-19	700000.00
7	7	Maruti Alto	2017-01-23	600000.00
8	8	Maruti S-Presso	2016-07-10	700000.00
9	9	Hyundai Creta	2011-01-14	2100000.00
10	10	Mahindra Thar	2012-07-23	1700000.00
11	11	Tata Punch	2020-07-17	1000000.00
12	12	Maruti Brezza	2021-01-14	1500000.00
13	13	Mercedes-Benz GLA	2019-05-10	6000000.00
14	14	Land Rover Range Rover Evoque	2007-01-17	7000000.00
15	15	Land Rover Range Rover	2018-05-14	50000000.00

- ```
  Select * From Carsale
  Where Salesdate > '01-01-2016'
  ```

	Id	Model	Salesdate	Purchase_Amount
1	5	Maruti Alto K10	2024-03-19	600000.00
2	6	Renault KWID	2023-03-19	700000.00
3	7	Maruti Alto	2017-01-23	600000.00
4	8	Maruti S-Presso	2016-07-10	700000.00
5	11	Tata Punch	2020-07-17	1000000.00
6	12	Maruti Brezza	2021-01-14	1500000.00
7	13	Mercedes-Benz GLA	2019-05-10	6000000.00
8	15	Land Rover Range Rover	2018-05-14	50000000.00

28. Write the same query as for question 27, but use a different punctuationmark to separate the parts of the date value.

- ```
  Select * From Carsale
  Where Salesdate > '01/01/2016'
  ```

29. Write an SQL query that will select the salesdate field from the Carsale table formatted as follows: month, day, and year in that order, numeric, separated by slashes. You may assume that your computer is using U.S. default settings.

- ```
  Select Format(Salesdate,'MM/dd/yyyy') As [New Date Format]
  From Carsale
  ```
  ```

	New Date Format
1	01/14/2001
2	02/10/2010
3	10/17/2011
4	05/10/2015
5	03/19/2024
6	03/19/2023
7	01/23/2017
8	07/10/2016
9	01/14/2011
10	07/23/2012
11	07/17/2020
12	01/14/2021
13	05/10/2019
14	01/17/2007
15	05/14/2018

30. Write an SQL query that will select the salesdate field from the Carsale table formatted as follows: the full name of the day, a comma, the full nameof the month, the number of the day in the month, a comma, and a 4 digit year.

- Select FORMAT(Salesdate,'dddd,MMMM,dd,yyyy') As [New Date Format]
  From Carsale

	New Date Format
1	Sunday,January,14,2001
2	Wednesday,February,10,2010
3	Monday,October,17,2011
4	Sunday,May,10,2015
5	Tuesday,March,19,2024
6	Sunday,March,19,2023
7	Monday,January,23,2017
8	Sunday,July,10,2016
9	Friday,January,14,2011
10	Monday,July,23,2012
11	Friday,July,17,2020
12	Thursday,January,14,2021
13	Friday,May,10,2019
14	Wednesday,January,17,2007
15	Monday,May,14,2018

31. Write an SQL query that will select the salesdate field from the Carsale table formatted as follows: 2 digit number of day in the month, 3 letter abbreviation of month, 2 digit year. Spaces should be used as separators.

- Select FORMAT(Salesdate,'dd MMM yy') As [New Date Format]
  From Carsale

	New Date Format
1	14 Jan 01
2	10 Feb 10
3	17 Oct 11
4	10 May 15
5	19 Mar 24
6	19 Mar 23
7	23 Jan 17
8	10 Jul 16
9	14 Jan 11
10	23 Jul 12
11	17 Jul 20
12	14 Jan 21
13	10 May 19
14	17 Jan 07
15	14 May 18

32. Write an SQL query that will select the salesdate field from the Carsale table formatted as follows: 4 digit year, comma, 2 digit week of year, comma, 1 digit number of day of the week.

- ```
  Select FORMAT(Salesdate,'yyyy,Iyyy') + ',' +
  SUBSTRING(FORMAT(Salesdate,'d'),1,1) As [New Date Format]
  From Carsale
  ```

	New Date Format
1	2001,I2001,1
2	2010,I2010,2
3	2011,I2011,1
4	2015,I2015,5
5	2024,I2024,3
6	2023,I2023,3
7	2017,I2017,1
8	2016,I2016,7
9	2011,I2011,1
10	2012,I2012,7
11	2020,I2020,7
12	2021,I2021,1
13	2019,I2019,5
14	2007,I2007,1
15	2018,I2018,5
  ```

SQL Assignment

Table Employee

```
Create Table Employee(Employee_Id int Primary Key,First_name Varchar(50),
Last_name Varchar(50),Salary Money,Joining_date datetime,Department Varchar(50))

Insert Into Employee Values(1,'Anika','Arora',100000,'2020-02-14 09:00:00','Hr')
,(2,'Veena','Varma',80000,'2011-06-15 09:00:00','Admin')
,(3,'Vishal','Singhal',300000,'2020-02-16 09:00:00','Hr')
,(4,'Sushant','Sing',500000,'2020-02-17 09:00:00','Admin')
,(5,'Bhupal','Bhati',500000,'2011-06-18 09:00:00','Admin')
,(6,'Dheeraj','Diwan',200000,'2011-06-19 09:00:00','Account')
,(7,'Karan','Kumar',75000,'2020-01-14 09:00:00','Account')
,(8,'Chandrika','Chauhan',90000,'2011-04-15 09:00:00','Admin')
```

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	1	Anika	Arora	100000.00	2020-02-14 09:00:00.000	Hr
2	2	Veena	Varma	80000.00	2011-06-15 09:00:00.000	Admin
3	3	Vishal	Singhal	300000.00	2020-02-16 09:00:00.000	Hr
4	4	Sushant	Sing	500000.00	2020-02-17 09:00:00.000	Admin
5	5	Bhupal	Bhati	500000.00	2011-06-18 09:00:00.000	Admin
6	6	Dheeraj	Diwan	200000.00	2011-06-19 09:00:00.000	Account
7	7	Karan	Kumar	75000.00	2020-01-14 09:00:00.000	Account
8	8	Chandrika	Chauhan	90000.00	2011-04-15 09:00:00.000	Admin

Table Employee_Bonus

```
Create Table Employee_Bonus(Employee_ref_id int,Bonus_Amount Money,
Bonus_Date datetime,
Foreign Key(Employee_ref_id) References Employee(Employee_Id))

Insert into Employee_Bonus Values(1,5000,'2020-02-16 0:00:00')
,(2,3000,'2011-06-16 0:00:00')
,(3,4000,'2020-02-16 0:00:00')
,(1,4500,'2020-02-16 0:00:00')
,(2,3500,'2011-06-16 0:00:00')

Select * From Employee_Bonus
```

	Employee_ref_id	Bonus_Amount	Bonus_Date
1	1	5000.00	2020-02-16 00:00:00.000
2	2	3000.00	2011-06-16 00:00:00.000
3	3	4000.00	2020-02-16 00:00:00.000
4	1	4500.00	2020-02-16 00:00:00.000
5	2	3500.00	2011-06-16 00:00:00.000

Table Employee_Title

```
Create Table Employee_Title(Employee_ref_id int,Employee_title Varchar(50),
Affective_Date datetime, Foreign Key(Employee_ref_id) References
Employee(Employee_Id))

Insert Into Employee_Title Values(1,'Manager','2016-02-20 0:00:00')
,(2,'Executive','2016-06-11 0:00:00')
,(8,'Executive','2016-06-11 0:00:00')
```

```
,(5,'Manager','2016-06-11 0:00:00')
,(4,'Asst. Manager','2016-06-11 0:00:00')
,(7,'Executive','2016-06-11 0:00:00')
,(6,'Lead','2016-06-11 0:00:00')
,(3,'Lead','2016-06-11 0:00:00')

Select * From Employee_Title
```

	Employee_ref_id	Employee_title	Affective_Date
1	1	Manager	2016-02-20 00:00:00.000
2	2	Executive	2016-06-11 00:00:00.000
3	8	Executive	2016-06-11 00:00:00.000
4	5	Manager	2016-06-11 00:00:00.000
5	4	Asst. Manager	2016-06-11 00:00:00.000
6	7	Executive	2016-06-11 00:00:00.000
7	6	Lead	2016-06-11 00:00:00.000
8	3	Lead	2016-06-11 00:00:00.000

Tasks To Be Performed:

1. Display the FIRST_NAME from the Employee Table using the alias name as Employee_name.

- ```
  Select First_name As Employee_name From Employee
  ```

	Employee_name
1	Anika
2	Veena
3	Vishal
4	Sushant
5	Bhupal
6	Dheeraj
7	Karan
8	Chandrika

2. Display LAST_NAME from the Employee Table in upper case.

- ```
  Select Upper(Last_name) As [Employee Last Name] From Employee
  ```

	Employee Last Name
1	ARORA
2	VARMA
3	SINGHAL
4	SING
5	BHATI
6	DIWAN
7	KUMAR
8	CHAUHAN

3. Display unique values of DEPARTMENT from the Employee Table.

- `Select Distinct(Department) From Employee`

	Department
1	Account
2	Admin
3	Hr

4. Display the first three characters of LAST_NAME from the Employee Table.

- `Select SUBSTRING(Last_name,1,3) As [First 3 Characters] From Employee`

	First 3 Characters
1	Aro
2	Var
3	Sin
4	Sin
5	Bha
6	Diw
7	Kum
8	Cha

5. Display the unique values of DEPARTMENT from the Employee Table and print its length.

- `Select Distinct(Department),Len(Department) As [Character Length] From Employee`

	Department	Character Length
1	Account	7
2	Admin	5
3	Hr	2

6. Display the FIRST_NAME and LAST_NAME from the Employee Table into a single column as FULL_NAME. A space char should separate them.

- `Select First_name +' '+Last_name As Full_Name From Employee`

	Full_Name
1	Anika Arora
2	Veena Varma
3	Vishal Singhal
4	Sushant Sing
5	Bhupal Bhati
6	Dheeraj Diwan
7	Karan Kumar
8	Chandrika Chauhan

7. Display all employee details from the Employee Table order by FIRST_NAME ascending.

- `Select * From Employee Order By First_name Asc`

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	1	Anika	Arora	100000.00	2020-02-14 09:00:00.000	Hr
2	5	Bhupal	Bhati	500000.00	2011-06-18 09:00:00.000	Admin
3	8	Chandrika	Chauhan	90000.00	2011-04-15 09:00:00.000	Admin
4	6	Dheeraj	Diwan	200000.00	2011-06-19 09:00:00.000	Account
5	7	Karan	Kumar	75000.00	2020-01-14 09:00:00.000	Account
6	4	Sushant	Sing	500000.00	2020-02-17 09:00:00.000	Admin
7	2	Veena	Varma	80000.00	2011-06-15 09:00:00.000	Admin
8	3	Vishal	Singhal	300000.00	2020-02-16 09:00:00.000	Hr

8. Display all employee details ordered by FIRST_NAME ascending and DEPARTMENT descending.

- `Select * From Employee Order By First_name Asc,Department Desc`

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	1	Anika	Arora	100000.00	2020-02-14 09:00:00.000	Hr
2	5	Bhupal	Bhati	500000.00	2011-06-18 09:00:00.000	Admin
3	8	Chandrika	Chauhan	90000.00	2011-04-15 09:00:00.000	Admin
4	6	Dheeraj	Diwan	200000.00	2011-06-19 09:00:00.000	Account
5	7	Karan	Kumar	75000.00	2020-01-14 09:00:00.000	Account
6	4	Sushant	Sing	500000.00	2020-02-17 09:00:00.000	Admin
7	2	Veena	Varma	80000.00	2011-06-15 09:00:00.000	Admin
8	3	Vishal	Singhal	300000.00	2020-02-16 09:00:00.000	Hr

9. Display details of employees with the first name as “VEENA '' and “KARAN” from the Employee Table.

- `Select * From Employee Where First_name in ('Veena','Karan')`

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	2	Veena	Varma	80000.00	2011-06-15 09:00:00.000	Admin
2	7	Karan	Kumar	75000.00	2020-01-14 09:00:00.000	Account

10. Display details of employees with DEPARTMENT name as “Admin”.

- `Select * From Employee Where Department = 'Admin'`

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	2	Veena	Varma	80000.00	2011-06-15 09:00:00.000	Admin
2	4	Sushant	Sing	500000.00	2020-02-17 09:00:00.000	Admin
3	5	Bhupal	Bhati	500000.00	2011-06-18 09:00:00.000	Admin
4	8	Chandrika	Chauhan	90000.00	2011-04-15 09:00:00.000	Admin

11. Display details of the employees whose FIRST_NAME contains ‘V’.

- `Select * From Employee Where First_name Like '%V%'`

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	2	Veena	Varma	80000.00	2011-06-15 09:00:00.000	Admin
2	3	Vishal	Singhal	300000.00	2020-02-16 09:00:00.000	Hr

12. Display details of the employees whose salary lies between 100000 and 500000.

- ```
  Select * From Employee Where Salary Between 100000 And 500000
  ```

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	1	Anika	Arora	100000.00	2020-02-14 09:00:00.000	Hr
2	3	Vishal	Singhal	300000.00	2020-02-16 09:00:00.000	Hr
3	4	Sushant	Sing	500000.00	2020-02-17 09:00:00.000	Admin
4	5	Bhupal	Bhati	500000.00	2011-06-18 09:00:00.000	Admin
5	6	Dheeraj	Diwan	200000.00	2011-06-19 09:00:00.000	Account

13. Display details of the employees who have joined in February, 2020.

- ```
  Select * From Employee Where MONTH(Joining_date) = 2 And YEAR(Joining_date) = 2020
  ```

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	1	Anika	Arora	100000.00	2020-02-14 09:00:00.000	Hr
2	3	Vishal	Singhal	300000.00	2020-02-16 09:00:00.000	Hr
3	4	Sushant	Sing	500000.00	2020-02-17 09:00:00.000	Admin

14. Display employee names with salaries >= 50000 and <= 100000.

- ```
  Select First_name+' '+Last_name As [Full Name],Salary From Employee
  Where Salary >=50000 And Salary<=100000
  ```

	Full Name	Salary
1	Anika Arora	100000.00
2	Veena Varma	80000.00
3	Karan Kumar	75000.00
4	Chandrika Chauhan	90000.00

15. Display the number of employees for each department in descending order.

- ```
  Select Department,Count(Employee_Id)As [Number Of Employees] From Employee
  Group By Department
  Order By Department Desc
  ```

	Department	Number Of Employees
1	Hr	2
2	Admin	4
3	Account	2

16. Display details of employees who are also managers.

- ```
  Select * From Employee
  Where Employee_Id in
  (Select Employee_ref_id From Employee_Title
  Where Employee_title = 'Manager')
  ```

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	1	Anika	Arora	100000.00	2020-02-14 09:00:00.000	Hr
2	5	Bhupal	Bhati	500000.00	2011-06-18 09:00:00.000	Admin
  ```

17. Display duplicate records having matching data in some fields of a table.

- ```
  Select Department,Salary,COUNT(*) As Count From Employee
  Group By Department,Salary
  Having COUNT(*)>1
  ```

18. Display only odd rows from a table.

- ```
  With CTE As(
  Select *,ROW_NUMBER()Over(Order By Employee_Id Asc) As Sub From Employee)

  Select * From CTE Where Sub % 2 =1
  ```

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department	Sub
1	1	Anika	Arora	100000.00	2020-02-14 09:00:00.000	Hr	1
2	3	Vishal	Singhal	300000.00	2020-02-16 09:00:00.000	Hr	3
3	5	Bhupal	Bhati	500000.00	2011-06-18 09:00:00.000	Admin	5
4	7	Karan	Kumar	75000.00	2020-01-14 09:00:00.000	Account	7

19. Clone a new table from the Employee Table.

- ```
  Select * Into Employee2
  From Employee
  ```

20. Display the top 2 highest salaries from the table.

- ```
  Select Top 2 Salary From Employee Order By Salary Desc
  ```

	Salary
1	500000.00
2	500000.00

21. Display the list of employees with the same salary.

- ```
  Select First_name+' '+Last_name As [Full Name],Salary From Employee
  Where Salary In
  (Select Salary From Employee
  Group By Salary
  Having Count(*)>1)
  ```

	Full Name	Salary
1	Sushant Sing	500000.00
2	Bhupal Bhati	500000.00

22. Display the second highest salary from the table.

- ```
  Select Salary From
  (Select Salary,DENSE_RANK()Over(Order By Salary Desc)As Sub From Employee) As
  Sub2
  Where Sub = 2
  ```

	Salary
1	300000.00

23. Display the first 50% records from a table.

- ```
  Select Top 50 Percent * From Employee
  ```
  ```

	Employee_Id	First_name	Last_name	Salary	Joining_date	Department
1	1	Anika	Arora	100000.00	2020-02-14 09:00:00.000	Hr
2	2	Veena	Varma	80000.00	2011-06-15 09:00:00.000	Admin
3	3	Vishal	Singhal	300000.00	2020-02-16 09:00:00.000	Hr
4	4	Sushant	Sing	500000.00	2020-02-17 09:00:00.000	Admin

24. Display the departments that have less than 4 people in it.

- ```
  Select Department From Employee
  Group By Department
  Having Count(Employee_Id)<4
  ```

	Department
1	Account
2	Hr

25. Display all departments along with the number of people in there.

- ```
  Select Department,Count(Employee_Id) As [Number Of People] From Employee
  Group By Department
  ```

	Department	Number Of People
1	Account	2
2	Admin	4
3	Hr	2

26. Display the name of employees having the highest salary in each department.

- ```
  With MaxSalary As (
  Select First_name,Department,Salary,
  ROW_NUMBER()Over(Partition By Department Order By Salary Desc) As Sub From
  Employee)

  Select First_name,Department From MaxSalary
  Where Sub = 1
  ```

	First_name	Department
1	Dheeraj	Account
2	Sushant	Admin
3	Vishal	Hr

27. Display the names of employees who earn the highest salary.

- ```
  Select First_name From Employee
  Where Salary = (Select Max(Salary) From Employee)
  ```

	First_name
1	Sushant
2	Bhupal

28. Display the average salaries for each department.

- ```
  Select Department,Avg(Salary) As [Average Salary] From Employee
  Group By Department
  ```
  ```

	Department	Average Salary
1	Account	137500.00
2	Admin	292500.00
3	Hr	200000.00

29. Display the name of the employee who got the maximum bonus.

- ```
  Select E.First_name From Employee As E
  Join Employee_Bonus As EB
  ON E.Employee_Id = EB.Employee_ref_id
  Where EB.Bonus_Amount = (Select Max(Bonus_Amount) From Employee_Bonus)
  ```

	First_name
1	Anika

30. Display the first name and title of all the employees.

- ```
  Select E.First_name,T.Employee_title From Employee As E
  Join Employee_Title As T
  On E.Employee_Id = T.Employee_ref_id
  ```

	First_name	Employee_title
1	Anika	Manager
2	Veena	Executive
3	Chandrika	Executive
4	Bhupal	Manager
5	Sushant	Asst. Manager
6	Karan	Executive
7	Dheeraj	Lead
8	Vishal	Lead