Data manipulation operations (DDL, DML, TCL)

```
-- ASSIGNMENT 1:
create database CompanyDB

use CompanyDB

create table Employees( EmployeeID int not null Primary Key ,
                        FirstName varchar(50),
                        LastName VARCHAR(50) ,
                        Department VARCHAR(50),
                        Salary DECIMAL(10, 2))

select * from Employees



--ASSIGNMENT 2:
insert into Employees( EmployeeID , FirstName, LastName ,Department,Salary) values
(1,'John', 'Doe', 'HR', 50000),
(2,'Jane' ,'Smith',' IT', 60000),
(3,'David', 'Lee', 'Marketing', 55000 );

select * from Employees



--ASSIGNMENT 3:
select * from Employees

select FirstName, LastName, Salary
from Employees
where Department = 'IT' AND Salary > 5000;

select max(Salary) from Employees



--ASSIGNMENT 4:
update Employees
set Salary=55000
where FirstName ='John' and   LastName='Doe'
select * from Employees
```

```
update Employees
set Department='Marketing'
where FirstName ='Jane' and   LastName='Smith'
select * from Employees



--ASSIGNMENT 5:
delete from Employees
where  FirstName= 'David'AND  LastName= 'Lee'
select * from Employees

delete from Employees
where Department='Marketing'
select * from Employees



--ASSIGNMENT 6:
create table Products( ProductID int not null primary key, ProductName varchar(100), Category
varchar(100), Price int, Stock int)
select * from products

insert into Products(ProductID, ProductName, Category ,Price, Stock) values
(1,'Laptop ','Electronics', 1000, 50),
(2, 'Mouse' ,'Electronics', 20 ,200),
(3, 'Keyboard', 'Electronics', 30, 150),
(4, 'Printer', 'Electronics', 150, 75);
select * from products

select * from products
order by price desc

update Products
set Price=25
where ProductName='Mouse'
select * from products

delete from Products
where  ProductID =2
select * from products

select * from Products
order by ProductName asc;
```

```
--ASSIGNMENT 7:
create table Books (BookID int not null primary key, Title varchar(50), Author varchar(50),
PublishedYear int,  AvailableCopies int)
select * from Books
insert into Books(BookID, Title, Author, PublishedYear, AvailableCopies) values
(1, 'The Great Gatsby' ,'F. Scott Fitzgerald' ,1925, 3 ),
(2, '1984' ,'George Orwell', 1949, 2 ),
(3, 'To Kill a Mockingbird',' Harper Lee' ,1960, 5 ),
(4, 'Brave New World', 'Aldous Huxley' ,1932, 4 );
select * from Books

update Books
set AvailableCopies=5
where Title='1984';
select * from Books

delete from Books
where BookID=1
select * from Books

select * from Books
where PublishedYear >1950

--ASSIGNMENT 8:



create table Customers(CustomerID int not null primary key,  CustomerName varchar(50) ,Email
varchar(50) , PhoneNumber varchar(50) )
select * from Customers

insert into Customers( CustomerID ,CustomerName ,Email, PhoneNumber) values
(201, 'Alice Johnson' ,'alice@example.com', '555-1234'),
(202, 'Bob Smith', 'bob@example.com', '555-5678' ),
(203, 'Charlie Brown' ,'charlie@example.com', '555-8765');

select * from Customers
delete from Customers
insert into Customers( CustomerID ,CustomerName ,Email, PhoneNumber) values
(201, 'Alice Johnson' ,'alice@example.com', '555-1234'),
(202, 'Bob Smith', 'bob@example.com', '555-5678' ),
```

```
(203, 'Charlie Brown' ,'charlie@example.com', '555-8765');
select * from Customers

insert into Customers( CustomerID ,CustomerName ,Email, PhoneNumber) values
(204, ' David Wilson' ,'david@example.com',' 555-4321 ');
select * from Customers

update Customers
set PhoneNumber= '555-9999'
where CustomerName='Alice Johnson'
select * from Customers

delete from Customers
where  CustomerID =202
select * from Customers



--ASSIGNMENT 9:
create table Orders(OrderID int not null primary key,CustomerID int,  OrderDate varchar(50),
TotalAmount int)
select * from Orders

insert into Orders (OrderID, CustomerID ,OrderDate, TotalAmount) values
(301, 201, '2024-09-01', 250.00 ),
(302, 202, '2024-09-02', 150.00 ),
(303, 203, '2024-09-03', 200.00 );
select * from Orders

insert into Orders (OrderID, CustomerID ,OrderDate, TotalAmount) values
(304, 204, '2024-09-04', 300.00 );
select * from Orders

update Orders
set  TotalAmount=275.00
where OrderID=301;
select * from Orders

delete from Orders
where OrderID =302
select * from Orders

select * from Orders
where TotalAmount> 200
```

```
select OrderID ,TotalAmount from Orders



--ASSIGNMENT 10:
create table Students(StudentID int primary key, FirstName varchar(50), LastName varchar(50),
Email varchar(50))
select * from Students

insert into Students(StudentID , FirstName , LastName , Email)  values
(1, 'John', 'Doe', 'john.doe@example.com'),
(2, 'Jane', 'Smith', 'jane.smith@example.com'),
(3, 'Mike', 'Johnson', 'mike.johnson@example.com');
select * from Students



--ASSIGNMENT 11:
create table Course(CourseID int primary key, CourseName varchar(50))

insert into Course (CourseID , CourseName) values
(1, 'Math'),
(2, 'science'),
(3, 'Math'),
(4, 'Hindi'),
(5, 'Hindi');
select * from Course

create table Enrollments(EnrollmetID int primary key, StudentID int , CourseID int,
foreign key (StudentID) references Students(StudentID),foreign key (CourseID) references
Course(CourseID))
select * from Enrollments
insert into Enrollments( EnrollmetID, StudentID, CourseID) values
(1,1,1),
(2,6,9);



--ASSIGNMENT 12: PRIMARY KEY (OrderID, ProductID): This defines a composite key, which
means that the combination of OrderID and ProductID must
--be unique across the table. This ensures that within a single order, each product can only
appear once
```

--(1,101,5), (2,102,9),(1,102,15),(3,103,2),(1,101,5): 1,101 combination repeated which is an error

```
 create table  OrderDetails (OrderID int ,ProductID int , Quantity int, primary key (OrderID, ProductID))
 select * from  OrderDetails
 Insert into OrderDetails (OrderID ,ProductID , Quantity ) values
 (1,1,5)
 Insert into OrderDetails (OrderID ,ProductID , Quantity ) values
 (2,2,6)
 Insert into OrderDetails (OrderID ,ProductID , Quantity ) values
 (3,3,10)
 Insert into OrderDetails (OrderID ,ProductID , Quantity ) values
 (4,4,9)
 Insert into OrderDetails (OrderID ,ProductID , Quantity ) values
 (5,5,4)

  select * from  OrderDetails
```

--ASSIGNMENT 13: Both Username and Email are candidate keys because each can potentially be used to uniquely identify a user

--The Email column is chosen as the alternate key because it is marked with the UNIQUE constraint.

--Even though its not the primary key, it still ensures uniqueness.

```
  create table Users( OrderID int primary key, email varchar(100) Unique, Username varchar(100));
  insert into  Users(OrderID, email, Username) values
  (1,'123@gmail.com', 'John'),
  (2,'127@gmail.com', 'Jacob'),
  (3,'923@gmail.com', 'Riya'),
  (4,'743@gmail.com', 'Raj')
   select * from  Users

    insert into  Users(OrderID, email, Username) values
  (8,'123@gmail.com', 'Jenna') : error because email field is set to unique. Here this mail id is duplicate.
```

--ASSIGNMENT 14:A surrogate key violation occurs when you try to manually insert or modify a value in an auto-incremented column (like AutoID)

```
  --that is meant to be automatically generated by the database.
  --Since the surrogate key is managed by the database, any attempt to interfere with this
process will cause an error

 CREATE TABLE Product (
    AutoID INT IDENTITY(1,1) PRIMARY KEY,  -- Surrogate Key with auto-increment
    ProductName VARCHAR(100) NOT NULL,     -- Name of the product
    Price DECIMAL(10, 2) NOT NULL          -- Price of the product
);
 select * from  Product
 INSERT INTO Product (ProductName, Price) values
              ('Laptop', 1000.00),
       ('Mouse', 20.50),
       ('Keyboard', 30.75),
       ('Printer', 150.00);
select * from  Product



--ASSIGNMENT 15: SSN must be unique for every employee, enforcing the integrity of the SSN
as a natural key.

create table Employee (
    SSN CHAR(11) PRIMARY KEY,      -- Natural key: SSN is unique for each employee
    FirstName VARCHAR(50) NOT NULL,  -- First name of the employee
    LastName VARCHAR(50) NOT NULL,   -- Last name of the employee
    Position VARCHAR(100) NOT NULL   -- Employee's position or job title
);
select * from Employee

insert into Employee (SSN, FirstName, LastName, Position) values
              ('123-45-6789', 'John', 'Doe', 'Software Engineer'),
       ('987-65-4321', 'Jane', 'Smith', 'Data Analyst');
  select * from Employee

 insert into Employee (SSN, FirstName, LastName, Position) values
            ('123-45-6789', 'Alice', 'Johnson', 'Manager');



--ASSIGNMENT 16:The primary key (CustomerID) ensures that each customer is uniquely
identified.
--The unique key (PhoneNumber) ensures that no two customers can have the same phone
number.
```

```
--There is a duplicate value for phone number.

create table Customer (
   CustomerID INT PRIMARY KEY,        -- Primary key: unique identifier for each customer
   CustomerName VARCHAR(100) NOT NULL, -- Customer's name
   PhoneNumber VARCHAR(15) UNIQUE      -- Unique phone number for each customer
);
 select * from Customer
 insert into Customer (CustomerID, CustomerName, PhoneNumber) values
            (1, 'Alice Johnson', '123-456-7890'),
      (2, 'Bob Smith', '555-123-4567');
            select * from Customer

insert into Customer (CustomerID, CustomerName, PhoneNumber) values
            (3, 'Charlie Brown', '123-456-7890');



 --ASSIGNMENT 17:
create table Employees( EmployeeID int not null Primary Key ,
                                    FirstName varchar(50),
                                    LastName VARCHAR(50) ,
                                    Department VARCHAR(50),
                                    Salary DECIMAL(10, 2))
select * from Employees

insert into Employees( EmployeeID , FirstName, LastName ,Department,Salary) values
(2,'Jane' ,'Smith',' IT', 60000),
(3,'David', 'Lee', 'Marketing', 55000 );
select * from Employees

 create view Employeeview as
 select FirstName, LastName ,Department
 from Employees
 select * from Employeeview




-- ASSIGNMENT 18:
 create view EditableEmployeeview as
 select EmployeeID, FirstName, LastName ,Department, Salary
 from Employees
 select * from EditableEmployeeview
```

```
update EditableEmployeeview
set salary=600000
where EmployeeID=1;
select * from EditableEmployeeview



-- ASSIGNMENT 19:
create view EmployeeSalaryBonusView as
select EmployeeID, FirstName, LastName ,Department, Salary*0.10 as bonus
from Employees

select *from EmployeeSalaryBonusView



 -- ASSIGNMENT 20:
alter view Employeeview as
select FirstName, LastName ,Department, Salary
from Employees
select * from Employeeview



 -- ASSIGNMENT 21:
  Create view  FilteredEmployeeView as
  select FirstName, LastName ,Department
  from Employees
  select * from  FilteredEmployeeView

  select * from  FilteredEmployeeView
  where Department='HR'



  -- ASSIGNMENT 22:
  drop view Employeeview
  select * from Employeeview
```