Assignment 1 –

Creating Table –

CREATE TABLE Studentinfo ( Roll INT, Name VARCHAR(50), Branch VARCHAR(50), Mobile VARCHAR(50), Total INT)

Inserting Data –

INSERT INTO Studentinfo VALUES

(101, 'Ayush', 'CSE', '1234567890', 78),

(102, 'Vrushal', 'ETC', '2345678901', 89),

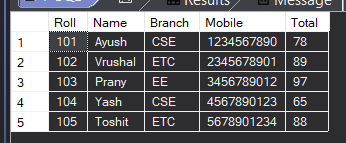
(103, 'Prany', 'EE', '3456789012', 97),

(104, 'Yash', 'CSE', '4567890123', 65),

(105, 'Toshit', 'ETC', '5678901234', 88)

Display data –

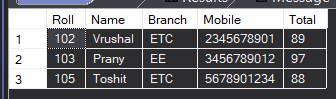
SELECT \* FROM Studentinfo;



Que 1. Display record having total > 80 using View?

CREATE VIEW total\_above\_80 AS SELECT \* FROM Studentinfo WHERE total > 80;

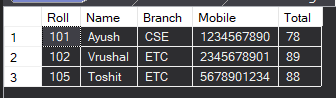
SELECT \* FROM total\_above\_80;



Que 2. Display Record where total is between 70 to 95

CREATE VIEW Bet\_total AS SELECT \* FROM Studentinfo WHERE total BETWEEN 70 AND 95

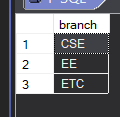
SELECT \* FROM Bet\_total



Que 3. Create view for group by branch

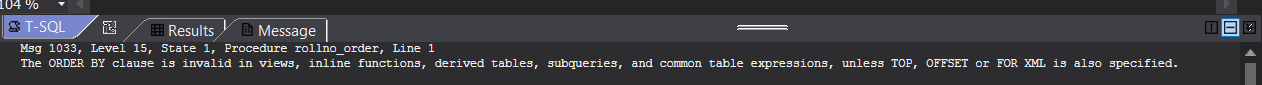
CREATE VIEW Group\_branch AS SELECT branch FROM studentinfo GROUP BY branch;

SELECT \* FROM Group\_branch;



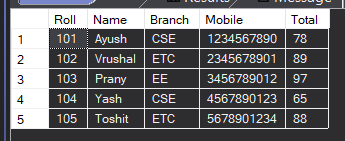
Que 4. Create view for descending Total

CREATE VIEW rollno\_order AS SELECT \* FROM studentinfo ORDER BY roll DESC;

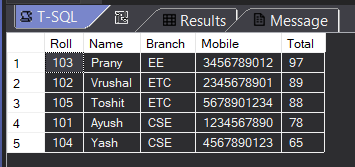


CREATE VIEW D\_studentinfo AS SELECT \* FROM studentinfo;

SELECT \* FROM D\_studentinfo;



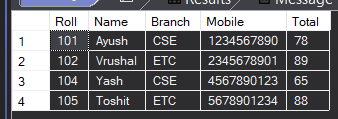
SELECT \* FROM D\_studentinfo ORDER BY total DESC;



Que 5. Create view CSE and ETC branch

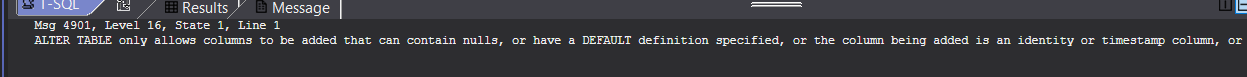
CREATE VIEW branch\_in AS SELECT \* FROM studentinfo WHERE branch IN ('CSE', 'ETC');

SELECT \* FROM branch\_in;

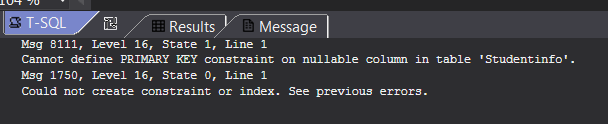


Que 6. Add new column as a primary key

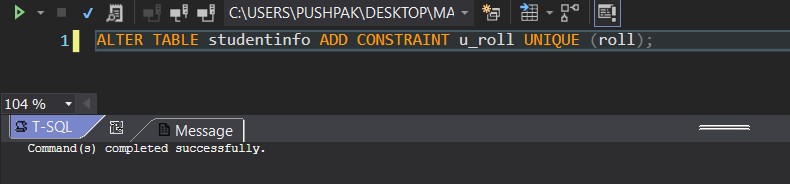
ALTER TABLE Studentinfo ADD u\_roll INT PRIMARY KEY;



ALTER TABLE Studentinfo ADD CONSTRAINT u\_roll PRIMARY KEY (roll);



Que 7. Make unique key

ALTER TABLE studentinfo ADD CONSTRAINT u\_roll UNIQUE (roll); 

Assignment 3 –

Que 1. Insert data using store procedure

Insert Data –

CREATE PROCEDURE insertdata(@id1 AS INT,

@Name1 AS VARCHAR(50),

@Email1 AS VARCHAR(50),

@Mobile1 AS VARCHAR(50),

@Salary1 AS VARCHAR(50)

)

AS

BEGIN

INSERT INTO Employee (Id, Name, Email, Mobile, salary) VALUES (@id1, @Name1, @Email1, @Mobile1, @Salary1);

END;

SQL Query –

EXEC insertdata @Id1 = 101, @Name1 = 'Ayush', @Email1 = 'ayush@gmail.com', @Mobile1 = '1234567890', @Salary1 = 40000

EXEC insertdata @Id1 = 102, @Name1 = 'Vrushal', @Email1 = 'vrushal@gmail.com', @Mobile1 = '2345678901', @Salary1 = 50000

EXEC insertdata @Id1 = 103, @Name1 = 'Yash', @Email1 = 'vrushal@gmail.com', @Mobile1 = '3456789012', @Salary1 = 20000

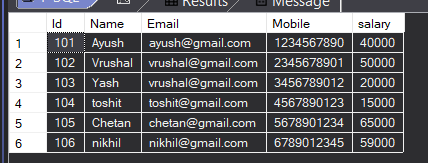
EXEC insertdata @Id1 = 104, @Name1 = 'toshit', @Email1 = 'toshit@gmail.com', @Mobile1 = '4567890123', @Salary1 = 15000

EXEC insertdata @Id1 = 105, @Name1 = 'Chetan', @Email1 = 'chetan@gmail.com', @Mobile1 = '5678901234', @Salary1 = 65000

EXEC insertdata @Id1 = 106, @Name1 = 'nikhil', @Email1 = 'nikhil@gmail.com', @Mobile1 = '6789012345', @Salary1 = 59000

Output –

SELECT \* FROM employee



Que 2. Update using store procedure

Store procedure –

CREATE PROCEDURE update\_data(

@id1 AS INT,

@Name1 AS VARCHAR(50),

@Email1 AS VARCHAR(50),

@Mobile1 AS VARCHAR(50),

@Salary1 AS int

)

AS

BEGIN

UPDATE Employee SET Name = @Name1, Email = @Email1, Mobile = @Mobile1 , salary = @Salary1 WHERE Id = @id1;

END;

SQL query –

EXEC update\_data @id1 = 101, @Name1 = 'Mohan', @Email1 = 'Mohan@gmail.com', @Mobile1 = '9988776655', @Salary1 = 99999;

Output –

SELECT \* FROM Employee



Update only Name

Store Procedure –

CREATE PROCEDURE update\_name(

@id1 AS INT,

@Name1 AS VARCHAR(50)

)

AS

BEGIN

UPDATE employee SET Name = @Name1 WHERE Id = @id1;

END;

SQL Query –

EXEC update\_name @id1 = 103, @Name1 = 'Pranay'

Output –

SELECT \* FROM Employee

