Exp 3

```
SQLQuery1.sql - 30...C12\305 PC12 (68))*
CREATE Database Professionl_info;
Create table Solo_levelling (name varchar(20) NOT NULL, ROLL_NO INT PRIMARY KEY, Rank char(30),Level int);
SELECT * FROM Solo_levelling;
insert into Solo_levelling (name ,ROLL_NO,Rank,Level)values('Sung Jinwoo',01,'S',50);
insert into Solo_levelling (name ,ROLL_NO,Rank,Level)values('Mr Gunwoo',02,'S',49);
insert into Solo_levelling (name ,ROLL_NO,Rank,Level)values('Cha_hii',03,'A',46);
insert into Solo_levelling (name ,ROLL_NO,Rank,Level)values('Igris',04,'A',40);
```

120 %

Results | Messages

	name	ROLL_NO	Rank	Level
1	Sung Jinwoo	1	S	50
2	Mr Gunwoo	2	S	49
3	Cha_hii	3	A	46
4	Igris	4	A	40

```
delete from Solo_levelling where name= 'Cha_hii';
```

120 %

Results | Messages

	name	ROLL_NO	Rank	Level
1	Sung Jinwoo	1	S	50
2	Mr Gunwoo	2	S	49
3	Igris	4	A	40

```
ALTER TABLE Solo_levelling ADD Phone_no  int;
```

120 %

Results | Messages

	name	ROLL_NO	Rank	Level	Age	Phone_no
1	Sung Jinwoo	1	S	50	25	NULL
2	Igris	4	A	40	35	NULL

```
Update Solo_levelling set Age =35 where ROLL_NO = 4;
```

120 %

Results | Messages

	name	ROLL_NO	Rank	Level	Age
1	Sung Jinwoo	1	S	50	25
2	Igris	4	A	40	35

```
Select ROLL_NO , name from Solo_levelling ;
```

120 %

Results | Messages

	ROLL_NO	name
1	1	Sung Jinwoo
2	4	Igris

Exp 4

SQLQuery1.sql - 305...13.master (sa (55))*

```
create table student45(
std_id int primary key,
st_name varchar(20),
age int
);
insert into student45(std_id,st_name,age)values(1,'Mohit',20)
insert into student45(std_id,st_name,age)values(2,'Rohit',19)
insert into student45(std_id,st_name,age)values(3,'Chirag',18)
insert into student45(std_id,st_name,age)values(4,'Vishal',24)
insert into student45(std_id,st_name,age)values(5,'Sushi',25)
select*from student45;

update student45 set age= 34 where age =24;

delete from student45 where std_id=4;
```

133 %

Results | Messages

	std_id	st_name	age
1	1	Mohit	20
2	2	Rohit	19
3	3	Chirag	18
4	5	Sushi	25

Exp 5

SQLQuery1.sql - 30...C14\305 PC14 (67))*

```
use mohit;
CREATE LOGIN Mohit WITH PASSWORD = '1234';
grant select on dbo.doctor to MohitMishra;

Execute As user = 'MohitMishra';
select * from Doctor;
insert into Doctor values(2,'Harsh','mumbi',1400,25);
revert ;

revoke select on dbo.doctor to MohitMishra;
Execute As user = 'MohitMishra';
select * from Doctor;
revert;

BEGIN TRANSACTION;
insert into Doctor values(5,'Anaya','hyderabad',1700,20);
SAVE TRANSACTION Savepoint1;
select * from Doctor;
insert into Doctor values(6,'ram','hyderabad',1700,20);
select * from Doctor;
ROLLBACK TRANSACTION Savepoint1;
select * from Doctor;
```

111 %

Results | Messages

	ID	Name	Adress	Salary	age
1	1	Harsh	mumbi	1400	25
2	2	Harsh	mumbi	1400	25
3	5	Anaya	hyderabad	1700	20

Query executed successfully.

Exp 6

```
create database Mohit;
create table Employee(Emp_Id int primary key, Name varchar(20)Not null,Salary int);
select * from Employee;
insert into Employee(Emp_Id,Name,Salary)values (205,'Mohit',1500);
insert into Employee(Emp_Id,Name,Salary)values (206,'Sumit',2500);
insert into Employee(Emp_Id,Name,Salary)values (207,'Harsh',3500);
insert into Employee(Emp_Id,Name,Salary)values (208,'Ronak',2800);
insert into Employee(Emp_Id,Name,Salary)values (209,'Pavan',4500);
select * from Employee;
```

	Emp_Id	Name	Salary
1	205	Mohit	1500
2	206	Sumit	2500
3	207	Harsh	3500
4	208	Ronak	2800
5	209	Pavan	4500

```
select * from Employee;

SELECT LOWER(Name) FROM Employee;
```

	(No column name)
1	mohit
2	sumit
3	harsh
4	ronak
5	pavan

```
SELECT LOWER(Name) FROM Employee;
```

	(No column name)
1	mohit
2	sumit
3	harsh
4	ronak
5	pavan

SQLQuery1.sql - L...MOHIT MISHRA (57))*

```
SELECT LOWER(Name) FROM Employee;
SELECT UPPER(Name) FROM Employee;
```

146 %

Results | Messages

	(No column name)
1	MOHIT
2	SUMIT
3	HARSH
4	RONAK
5	PAVAN

SQLQuery1.sql - L...MOHIT MISHRA (57))*

```
SELECT LOWER(Name) FROM Employee;
SELECT UPPER(Name) FROM Employee;
SELECT LEN(Name) FROM Employee;
```

146 %

Results | Messages

	(No column name)
1	5
2	5
3	5
4	5
5	5

```
SELECT LOWER(Name) FROM Employee;
SELECT UPPER(Name) FROM Employee;
SELECT LEN(Name) FROM Employee;
SELECT SUBSTRING(Name, 3, 2) FROM Employee;
```

146 %

Results | Messages

	(No column name)
1	hi
2	mi
3	rs
4	na
5	va

```
SELECT LOWER(Name) FROM Employee;
SELECT UPPER(Name) FROM Employee;
SELECT LEN(Name) FROM Employee;
SELECT SUBSTRING(Name, 3, 2) FROM Employee;
SELECT SUM(Salary) FROM Employee;
```

146 %

Results | Messages

	(No column name)
1	14800

```
SELECT LOWER(Name) FROM Employee;
SELECT UPPER(Name) FROM Employee;
SELECT LEN(Name) FROM Employee;
SELECT SUBSTRING(Name, 3, 2) FROM Employee;
SELECT SUM(Salary) FROM Employee;
SELECT AVG(Salary) FROM Employee;
```

146 %

Results | Messages

	(No column name)
1	2960

~vsB640.sql - LAP...OHIT MISHRA (76))*

```
SELECT SUM(Salary) FROM Employee;
SELECT MIN(Salary) FROM Employee;
SELECT MAX(Salary) FROM Employee;
SELECT COUNT(Emp_ID) FROM Employee;
```

146 %

Results | Messages

	(No column name)
1	5

~vsB640.sql - LAP...OHIT MISHRA (76))*

```
SELECT UPPER(Name) FROM Employee;
SELECT LEN(Name) FROM Employee;
SELECT SUBSTRING(Name, 3, 2) FROM Employee;
SELECT SUM(Salary) FROM Employee;
SELECT MIN(Salary) FROM Employee;
```

146 %

Results | Messages

	(No column name)
1	1500

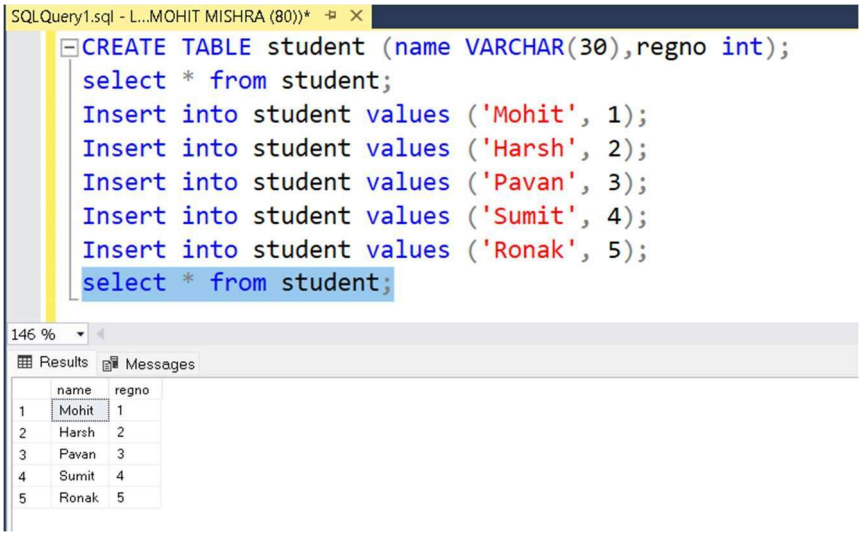
~vsB640.sql - LAP...OHIT MISHRA (76))*

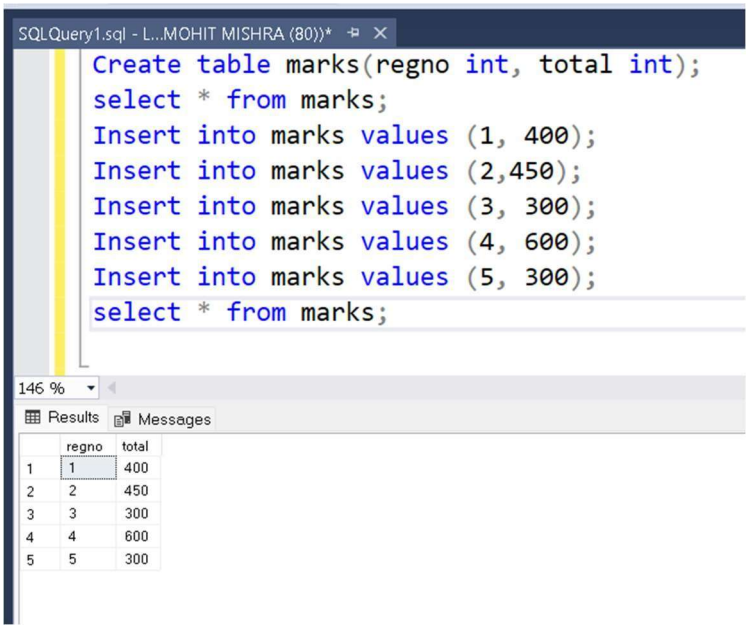
```
SELECT SUBSTRING(Name, 3, 2) FROM Employee;
SELECT SUM(Salary) FROM Employee;
SELECT MIN(Salary) FROM Employee;
SELECT MAX(Salary) FROM Employee;
```

146 %

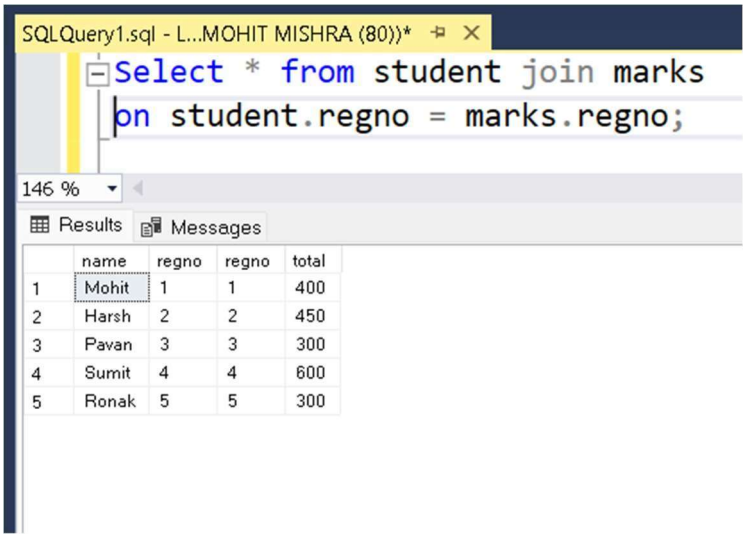
Results | Messages

	(No column name)
1	4500

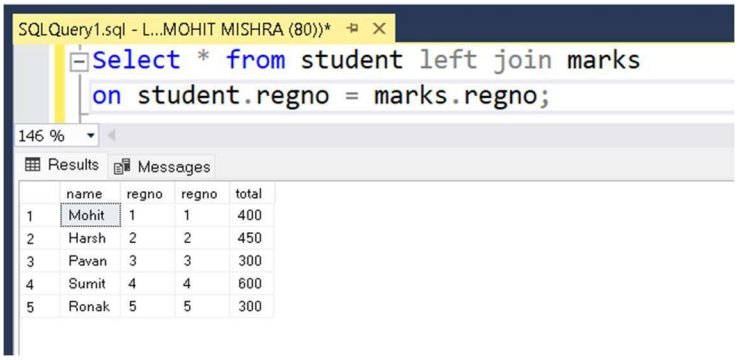




Natural Join:

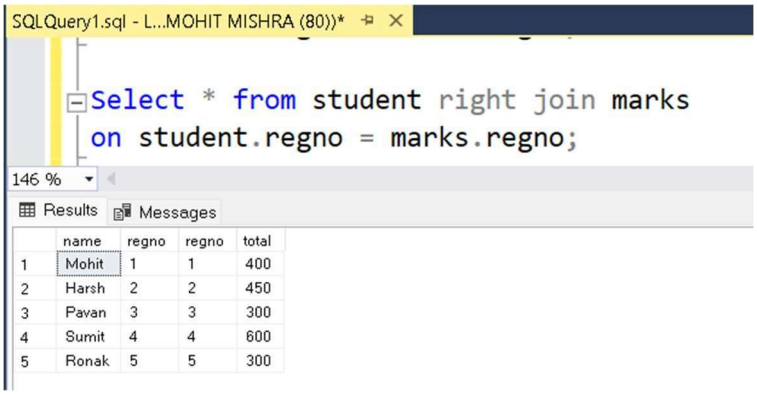


Left Join:



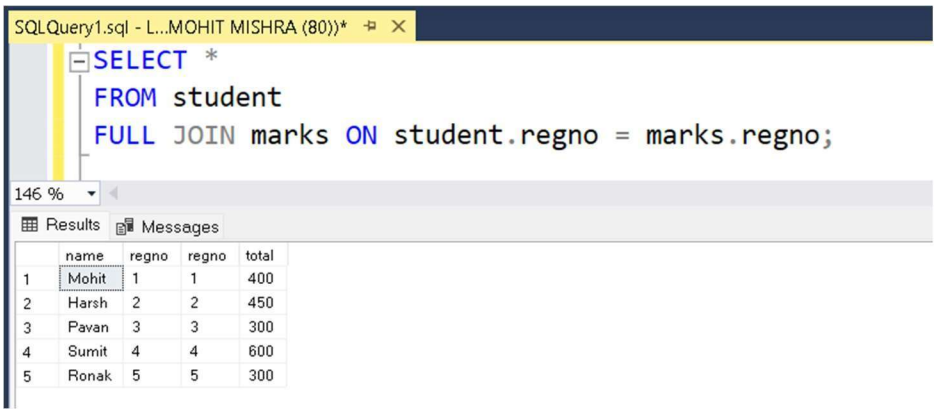
	name	regno	regno	total
1	Mohit	1	1	400
2	Harsh	2	2	450
3	Pavan	3	3	300
4	Sumit	4	4	600
5	Ronak	5	5	300

Right Join:



	name	regno	regno	total
1	Mohit	1	1	400
2	Harsh	2	2	450
3	Pavan	3	3	300
4	Sumit	4	4	600
5	Ronak	5	5	300

Full Join:



	name	regno	regno	total
1	Mohit	1	1	400
2	Harsh	2	2	450
3	Pavan	3	3	300
4	Sumit	4	4	600
5	Ronak	5	5	300

Exp 8

```
USE Mohit;
CREATE TABLE students ( student_id INT PRIMARY KEY,
student_name VARCHAR(50), major VARCHAR(50));
INSERT INTO students (student_id, student_name, major )
VALUES
(1, 'Mohit', 'Cybersecurity'),
(2, 'Harsh', 'Database'),
(3, 'Pavan ', 'Os'),
(4,'Sumit', 'maths'),
(5,'Aryan', 'graphics')
Select *from students
```

146 %

Results | Messages

	student_id	student_name	major
1	1	Mohit	Cybersecurity
2	2	Harsh	Database
3	3	Pavan	Os
4	4	Sumit	maths
5	5	Aryan	graphics

```
CREATE TABLE grades ( grade_id INT PRIMARY KEY,
student_id INT, course_name VARCHAR(50),
grade DECIMAL(3, 2));

INSERT INTO grades (grade_id, student_id, course_name, grade)
VALUES
(1, 1, 'Integration', 4.0), (2, 1, 'Statistics', 3.5),
(3, 2, 'Algorithms', 3.9), (4, 4, 'Laplace', 3.7),
(5, 3, 'Inverse', 3.2);
Select *from grades
```

146 %

Results | Messages

	grade_id	student_id	course_name	grade
1	1	1	Integration	4.00
2	2	1	Statistics	3.50
3	3	2	Algorithms	3.90
4	4	4	Laplace	3.70
5	5	3	Inverse	3.20

```
SELECT *FROM students
WHERE major IN ('Cybersecurity', 'Os');
```

146 %

Results | Messages

	student_id	student_name	major
1	1	Mohit	Cybersecurity
2	3	Pavan	Os

SQLQuery1.sql - L...MOHIT MISHRA (58))*

```
SELECT *FROM students
WHERE major NOT IN ('Os');
```

146 %

Results | Messages

	student_id	student_name	major
1	1	Mohit	Cybersecurity
2	2	Harsh	Database
3	4	Sumit	maths
4	5	Aryan	graphics

SQLQuery1.sql - L...MOHIT MISHRA (58))*

```
SELECT *FROM students
WHERE student_id = ANY (SELECT student_id FROM grades WHERE grade
>= 3.7);
```

146 %

Results | Messages

	student_id	student_name	major
1	1	Mohit	Cybersecurity
2	2	Harsh	Database
3	4	Sumit	maths

SQLQuery1.sql - L...MOHIT MISHRA (58))*

```
SELECT *FROM students
WHERE student_id = ALL (SELECT student_id FROM grades WHERE grade
>= 4.0);
```

146 %

Results | Messages

	student_id	student_name	major
1	1	Mohit	Cybersecurity

SQLQuery1.sql - L...MOHIT MISHRA (58))*

```
SELECT student_id, student_name, major,
(SELECT AVG(grade) FROM grades WHERE grades.student_id =
students.student_id) AS avg_grade
FROM students;
```

146 %

Results | Messages

	student_id	student_name	major	avg_grade
1	1	Mohit	Cybersecurity	3.750000
2	2	Harsh	Database	3.900000
3	3	Pavan	Os	3.200000
4	4	Sumit	maths	3.700000
5	5	Aryan	graphics	NULL

Exp 9

SQLQuery1.sql - L...MOHIT MISHRA (72))*

```
CREATE PROCEDURE factor @number INT
AS
BEGIN
    DECLARE @i INT = 1, @result INT = 1
    WHILE (@i <= @number)
    BEGIN
        SET @result = @result * @i
        SET @i += 1
    END
    SELECT @result AS Factorial
END
EXEC factor 5
```

146 %

Results | Messages

	Factorial
1	120

```sql
CREATE TABLE Student_details (
    id INTEGER,
    addmission_no INTEGER,
    first_name VARCHAR(10),
    last_name VARCHAR(20),
    age INTEGER,
    city VARCHAR(20)
);

INSERT INTO Student_details (id, addmission_no, first_name, last_name, age, city)
VALUES
(1, 1001, 'ayush', 'Chaubey', 20, 'New York'),
(2, 1002, 'Prashant', 'phatak', 22, 'Los Angeles'),
(3, 1003, 'naman', 'bhalani', 19, 'Chicago'),
(4, 1004, 'Nilesh', 'Yadav', 21, 'Houston');

SELECT * FROM Student_details;
```

id	addmission_no	first_name	last_name	age	city
1	1001	ayush	Chaubey	20	New York
2	1002	Prashant	phatak	22	Los Angeles
3	1003	naman	bhalani	19	Chicago
4	1004	Nilesh	Yadav	21	Houston

```sql
CREATE TABLE fees (
    addmission_no INTEGER,
    course VARCHAR(20),
    amount_paid INTEGER
);

INSERT INTO fees (addmission_no, course, amount_paid)
VALUES
(1001, 'Math', 2000),
(1001, 'English', 1500),
(1002, 'History', 1800),
(1003, 'Physics', 2500),
(1003, 'Chemistry', 2200),
(1004, 'Biology', 1900);

SELECT * FROM fees;
```

addmission_no	course	amount_paid
1001	Math	2000
1001	English	1500
1002	History	1800
1003	Physics	2500
1003	Chemistry	2200
1004	Biology	1900

```
CREATE VIEW course_enrolled AS
SELECT first_name, last_name, course, amount_paid
FROM Student_details AS S
INNER JOIN fees AS F ON S.addmission_no = F.addmission_no;

SELECT * FROM course_enrolled;
```

first_name	last_name	course	amount_paid
ayush	Chaubey	Math	2000
ayush	Chaubey	English	1500
Prashant	phatak	History	1800
naman	bhalani	Physics	2500
naman	bhalani	Chemistry	2200
Nilesh	Yadav	Biology	1900

```sql
CREATE TABLE Employee (
    Id INT PRIMARY KEY,
    Name VARCHAR(45),
    Salary INT,
    Gender VARCHAR(12),
    DepartmentId INT
);

INSERT INTO Employee VALUES
(1,'Steffan', 82000, 'Male', 3),
(2,'Amelie', 52000, 'Female', 2),
(3,'Antonio', 25000, 'male', 1),
(4,'Marco', 47000, 'Male', 2),
(5,'Eliana', 46000, 'Female', 3);

SELECT * FROM Employee;
```

Id	Name	Salary	Gender	DepartmentId
1	Steffan	82000	Male	3
2	Amelie	52000	Female	2
3	Antonio	25000	male	1
4	Marco	47000	Male	2
5	Eliana	46000	Female	3

```
CREATE TABLE Employee_Audit_Test (
    Id INT IDENTITY,
    Audit_Action TEXT
);

CREATE TRIGGER trInsertEmployee
ON Employee
FOR INSERT
AS
BEGIN
    DECLARE @Id INT
    SELECT @Id = Id FROM inserted
    INSERT INTO Employee_Audit_Test
    VALUES ('New employee with Id = ' + CAST(@Id AS VARCHAR(10)) +
            ' is added at ' + CAST(GETDATE() AS VARCHAR(22)))
END;
```

```
INSERT INTO Employee VALUES (6, 'Peter', 62000, 'Male', 3);

SELECT * FROM Employee_Audit_Test;
```

Id	Audit_Action
1	New employee with Id = 6 is added at Apr 14 2025 ...