anmamun0
almamun20044@gmail.com

1: Query :

Here writing Different Primary Key and Composite key.

Primary KEY	Composite KEY
The primary Key is a single Column, and here record is unique	Combine two or many primary key to make Composite key,
The primary key not has null value	Value is unique to any composite key, not necessary that the value unique a single column
Roll INT PRIMARY KEY	PRIMARY KEY (Roll,Grp_Name)

2: Query :

Here writing Different JOIN and with out JOIN

JOIN	With out JOIN
Allow to join multiple to single tables query	All query is separate query for each table
Type of JOIN -> INNER, LEFT, RIGHT,CROSS	Without JOIN we can use subquery
Relationships between primary key and foreign key	No has relationship any column
Query handling simple	Query handling complex
Simple underestending	Complex understanding
If we use JOIN our code Complexity will be minimize , we can think O(M+N)	If we take query without JOIN our code complexity , we can think O(N*M); just imajin

3: Query :

```
CREATE TABLE employees
(
   No INT AUTO_INCREMENT PRIMARY KEY,
   First_name VARCHAR(30),
   Last_name VARCHAR(10),
   date_of_birth DATE,
   Department_id INT,
   Salary INT
);
INSERT INTO employees(first_name,last_name,date_of_birth,department_id, salary)
VALUES('AN Mamun','Nova','2006-04-12',113,70000),
                ('Jak','Doe','2004-04-12',206,90000);
SELECT * FROM employees;

CREATE TABLE departments
(
        department_id INT PRIMARY KEY,
        department_name VARCHAR(50)
);

INSERT INTO departments(department_id,department_name)
VALUES (113,'CSE'), (206,'EEE');

SELECT * FROM Departments;
```

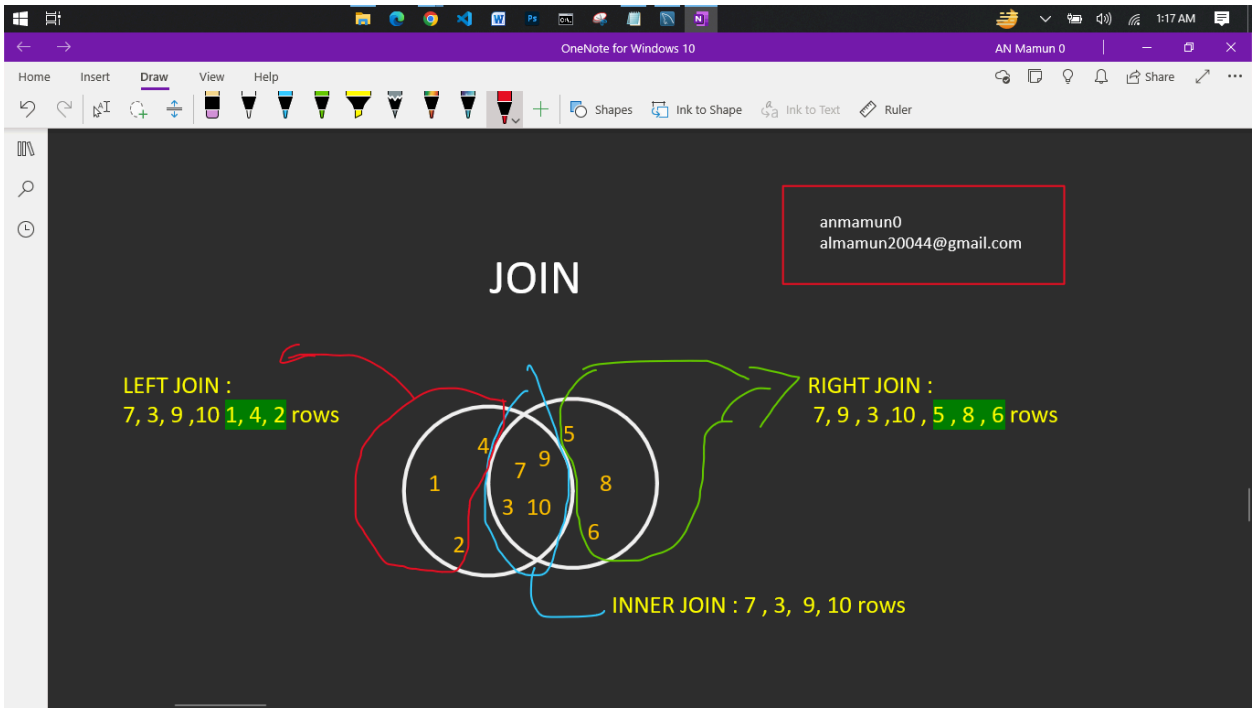
4: Query :

```
WITH MAXI AS
(
    SELECT MAX(salary) AS TK
    FROM employees
)
SELECT MAX(salary)
FROM employees
WHERE salary < (SELECT TK FROM MAXI);
```

5: Query :

```
SELECT DEP.department_name , AVG(salary)
FROM employees AS EM
RIGHT JOIN departments AS DEP
ON EM.department_id = DEP.department_id
GROUP BY DEP.department_name
ORDER BY department_name ASC;
```

6: Query :



INNER JOIN : only the rows that have matching value both tables , ,if not mach values both Tables they are not include;

LEFT JOIN: the rows that the left table match the right table, if any value not matching the Left value will added but left value will NULL;

RIGHT JOIN: the rows that the right table match the left table, if any value not matching the Right value will added but left value will be NULL;

SELF JOIN: that a table join with self, of self join we can easily convert the left join, self join is esy version then subquery,

7: Query :

SUB QUERY : SUB query we can call nested query , the query embedded inside another query, Can can slove many hiricical query with subquery , its can use any part of query

Here is some example to subquery:

```
– 2ND HIGEST SALARY
SELECT MAX(salary) AS MAXI
FROM Employees
WHERE salary < (SELECT MAX(salary)
                 FROM Employees);


– DEPARTMENT_ID   also  DEPARTMENT_NAME with subquery

SELECT EM.department_id,
        (SELECT department_name FROM departments
                WHERE EM.department_id = department_id) AS Department_Name
FROM employees AS EM;
```

8: Query :

```
WITH STV AS
(
        SELECT MIN(salary) STV_TK
        FROM employees
        WHERE first_name = 'steven'
)
SELECT first_name
FROM employees
WHERE salary < (SELECT STV_TK FROM STV);
```

9: Query :

```
SELECT EM.job_id , job_title , COUNT(employee_id) AS involve
FROM Employees AS EM
JOIN Jobs
ON EM.Job_id = Jobs.job_id
GROUP BY Job_id
ORDER BY involve DESC;
```

10: Query :

```
SELECT DEP.department_name
FROM Departments AS DEP
LEFT JOIN employees AS EM
ON DEP.department_id = EM.department_id
WHERE employee_id IS NULL;
```