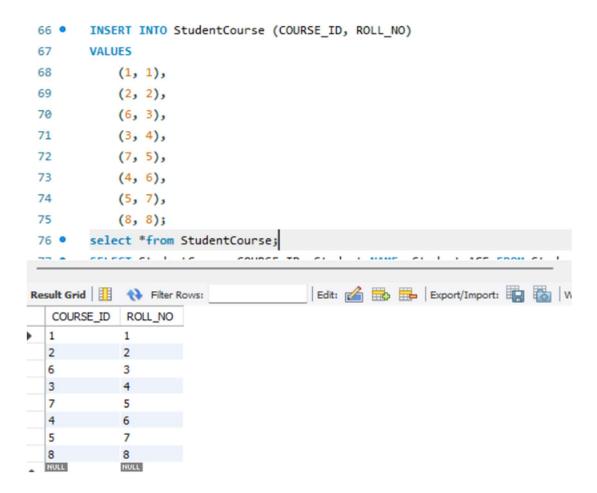
Name: Krushnakumar Patle

Email: krishnapatle128@gmail.com

Batch: Data Engineering Batch-1

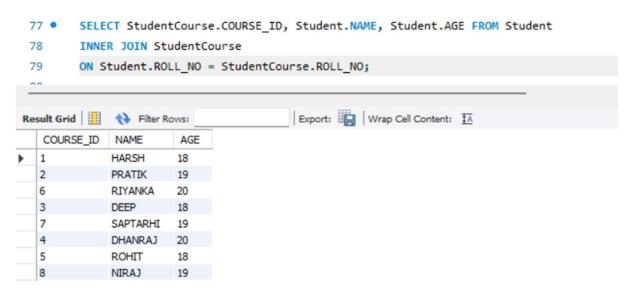
Topic Join and SubQueries

```
INSERT INTO Student (ROLL NO, NAME, ADDRESS, PHONE, Age) VALUES
 49 •
            (1, 'HARSH', 'DELHI', '947646383', 18),
 50
            (2, 'PRATIK', 'BIHAR', '894465483', 19),
 51
            (3, 'RIYANKA', 'SILIGURI', '9474383332', 20),
 52
            (4, 'DEEP', 'RAMNAGAR', '9037364632', 18),
 53
            (5, 'SAPTARHI', 'KOLKATA', '87463283743', 19),
 54
            (6, 'DHANRAJ', 'BARABAJAR', '789046532', 20),
 55
 56
            (7, 'ROHIT', 'BALURGHAT', 'ALIPUR', 18),
            (8, 'NIRAJ', 'ALIPUR', '7836463722', 19);
 57
 58
        select *from Student;
                                        Edit: 🚄 🖶 Export/Import: 🙀 🦝 W
ROLL_NO
           NAME
                     ADDRESS
                                PHONE
                                            Age
           HARSH
                     DELHI
                                947646383
                                            18
  1
  2
           PRATIK
                    BIHAR
                               894465483
                                           19
  3
           RIYANKA
                     SILIGURI
                                9474383332
                                            20
  4
           DEEP
                    RAMNAGAR 9037364632
                                           18
  5
           SAPTARHI
                    KOLKATA
                                87463283743
                                           19
  6
           DHANRAJ
                    BARABAJAR
                               789046532
                                           20
  7
           ROHIT
                     BALURGHAT
                               ALIPUR
                                            18
           NIRAJ
  8
                     ALIPUR
                               7836463722
                                           19
           NULL
                    NULL
                               NULL
                                           NULL
```



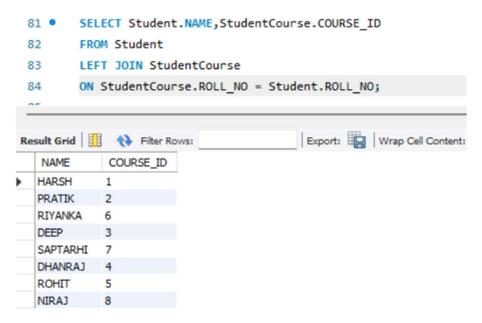
1. Inner JOIN:

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.



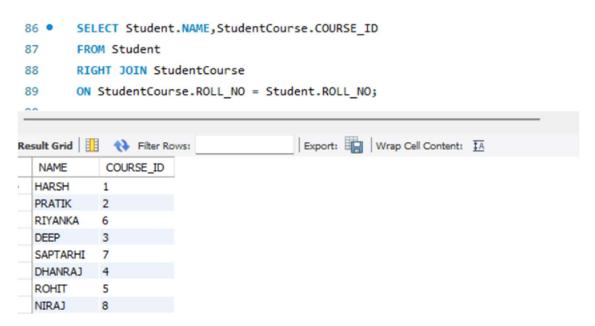
2. LEFT JOIN:

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain null. LEFT JOIN is also known as LEFT OUTER JOIN.



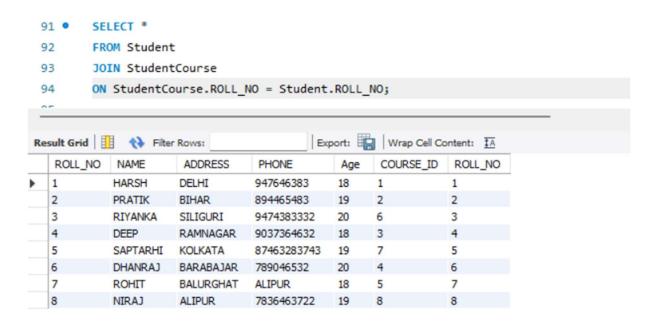
3. RIGHT JOIN:

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.



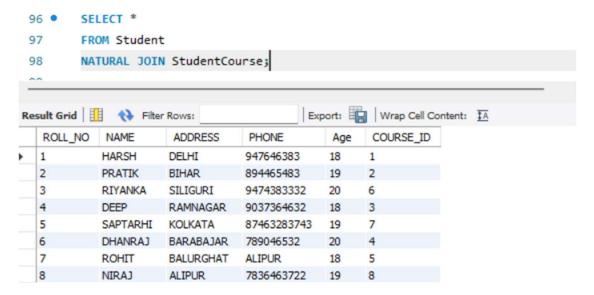
4. FULL JOIN:

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain NULL values.



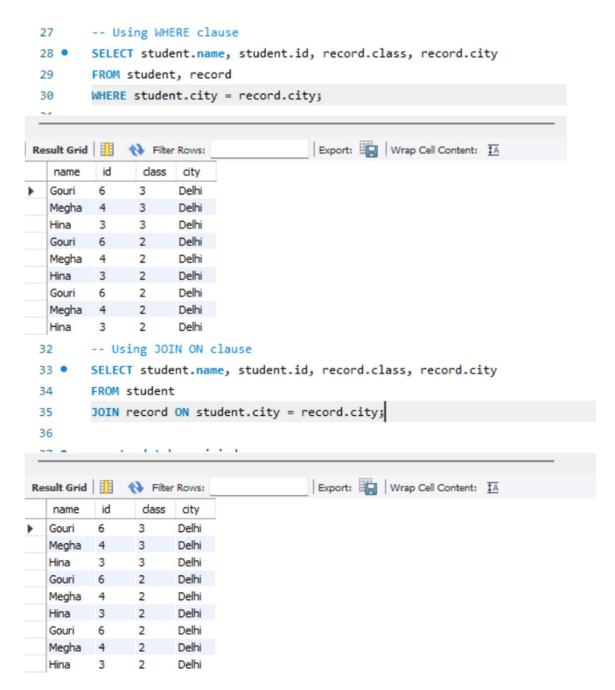
NATURAL JOIN:

A NATURAL JOIN is a type of join in SQL that automatically matches columns with the same name in both tables. It's based on the idea that if two tables have columns with the same name, they are likely to be related. The NATURAL JOIN clause eliminates the need to explicitly specify the join condition.



6. EQUI JOIN:

EQUI JOIN creates a JOIN for equality or matching column(s) values of the relative tables. EQUI JOIN also create JOIN by using JOIN with ON and then providing the names of the columns with their relative tables to check equality using equal sign (=).



7. NON EQUI JOIN:

NON EQUI JOIN performs a JOIN using comparison operator other than equal(=) sign like >, <, >=, <= with conditions.

