Natural Join

- Natural Join joins two (or more) tables based on same attribute name and datatypes. The resulting table will contain all the attributes of both the table but keep only one copy of each common column.
- Lets consider an example with 2 tables (Student & Grades) given below.

Student Table

student_id	name
1	Joe
2	Alexa
3	Peter

Grades Table

student_id	grades
2	70
3	60
4	75

If you execute the following query:

```
SELECT *  FROM Student NATURAL JOIN Grades;
```

The output will be

student_id	name	grades
2	Alexa	70
3	Peter	60

Note: To perform Natural Join of more than 2 tables follow the syntax here-> https://stackoverflow.com/q/5147453

Inner Join

- Inner Join joins two (or more) tables on the basis of the column which is explicitly specified in the ON clause. The resulting table will contain all the attributes from both the tables including common column as well.
- Also keep in mind, Inner Join returns the combined tuples between two or more tables where at least one attribute is in common. If there is no attribute in common between the tables then it will return nothing.
- Lets consider the same example with 2 tables (Student & Grades) mentioned earlier.

If you execute the following query:

```
SELECT * FROM Student S INNER JOIN Grades G ON
S.student_id = G.student_id;
```

The output will be 

student_id	name	student_id	grades
2	Alex	2	70
3	Peter	3	60

Note: You don't have to write specifically INNER JOIN in the query.
You can also write 'JOIN' to get the same output as follows:

```
SELECT * FROM Student S JOIN Grades G ON S.student_id =
G.student_id;
```

Outer Join

- Outer Join returns combined tuples from the specified tables even if the join condition fails.

There are three types of outer join in SQL as follows:

- ➢ Left Outer Join
- ➢ Right Outer Join
- ➢ Full Outer Join

- Lets consider the previous example of 2 tables (Student & Grades), where we consider Student table as the Left table and Grades table as the Right table.

To execute Left Outer Join on the Student & Grades tables let's execute the following:

```
SELECT * FROM Student S LEFT OUTER JOIN Grades G ON S.student_id = G.student_id;
```

The output will be →

student_id	name	student_id	grades
1	Joe	NULL	NULL
2	Alexa	2	70
3	Peter	3	60

Note: In Left Outer Join, it joins based on all the records from the left table (Student)

Outer Join (continued…)

To execute Right Outer Join on the Student & Grades tables let's execute the following:
SELECT * FROM Student S RIGHT OUTER JOIN Grades G ON S.student_id = G.student_id;

The output will be 

Note: In Right Outer Join, it joins based on all the records from the right table (Grades)

student_id	name	student_id	grades
2	Alexa	2	70
3	Peter	3	60
NULL	NULL	4	75

Outer Join (continued…)

To execute Full Outer Join on the Student & Grades tables let's execute the following:

```
SELECT * FROM Student S FULL OUTER JOIN Grades G ON S.student_id = G.student_id;
```

The output will be

student_id	name	student_id	grades
1	Joe	NULL	NULL
2	Alexa	2	70
3	Peter	3	60
NULL	NULL	4	75

Note: In Full Outer Join, it joins based on all the records from the both the table (Student & Grades)

Currently, MySQL don't support FULL OUTER JOIN. So, the workaround to get FULL OUTER JOIN in MySQL you actually need to execute the following query:

```
SELECT * FROM Student S LEFT OUTER JOIN Grades G ON S.student_id = G.student_id
UNION
SELECT * FROM Student S RIGHT OUTER JOIN Grades G ON S.student_id = G.student_id;
```

To try out these JOIN examples yourself you could download the code for the examples using the following link and try it out on your machine:

https://github.com/somdipdey/Essex-CE153/blob/main/JOINS/example.sql

OR you can also try the queries (example) online using the following link:

https://www.mycompiler.io/new/sql?fork=2Juntdq

Note: The online query execution doesn't work for Right Outer Join / Full Outer Join as the online SQL Server doesn't support the features yet.

Thank you!

- For further questions please address to somdip.dey@essex.ac.uk