3.4.3 LEFT OUTER JOIN:

this type of join will first select all shared records from those tables, and then will bring all records from left table which are not exist in the second table.

3.4.4 RIGHT OUTER JOIN

this type of join will first select all shared records from those tables, and then will bring all records from right table which are not exist in the first table

3.4.5 FULL OUTER JOIN this type of join will first select all shared records from those tables, and then will bring all records exist in first table which are not exist in the second table and then bring all records exist in the second table which are not exist in first table.

```
select  teachers.id , teachers.name , projects.id ,
projects.name  from  teachers full OUTER join projects  on
(teachers.id = projects.id);
```

```
id |  name   | id |   name    ----+--------
+----+---------
1 | Volker |  1 | compiler
2 | Elke   |  2 | xpaint
  4 | Perl
3 | games
```

4 : Database Administrator

A **database administrator** (**DBA**) is a person responsible for :

- Design and Implementation of the database
- maintenance and repair of the database
- monitoring and improving database performance and capacity
- planning for future expansion requirements.

A basic responsibility for just about every database administrator involves the installation of new databases. As part of the database installation, the database administrator will set up login credentials to authorized persons, define the privileges associated with each authorized user, and ensure that every work station attached to the network is set up to access the new database.

5: Database Design

Database design is the process of producing a detailed data model of a database , this data model which can then be used to create a database. The term database design can be used to describe many different parts of the design , it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views.

5.1 : The Design Process

The design process consists of the following steps:

1. Determine the purpose of your database - This helps prepare you for the remaining steps.
2. Find and organize the information required - Gather all of the types of information you might want to record in the database, such as product name and order number.
3. Divide the information into tables - Divide your information items into major entities or subjects, such as Products or Orders. Each subject then becomes a table.
4. Turn information items into columns - Decide what information you want to store in each table. Each item becomes a field, and is displayed as a column in the table. For example, an Employees table might include fields such as Last Name and Hire Date.
5. Specify primary keys - Choose each table’s primary key. The primary key is a column that is used to uniquely identify each row. An example might be Product ID or Order ID.

6. Set up the table relationships - Look at each table and decide how the data in one table is related to the data in other tables. Add fields to tables or create new tables to clarify the relationships, as necessary.
7. Refine your design - Analyze your design for errors. Create the tables and add a few records of sample data. See if you can get the results you want from your tables. Make adjustments to the design, as needed.
8. Apply the normalization rules - Apply the data normalization rules to see if your tables are structured correctly. Make adjustments to the tables, as needed.

5.2:Database Cardinality

In data modeling, the cardinality of one data table with respect to another data table is a critical aspect of database design. Relationships between data tables define cardinality when explaining how each table links to another.

In the relational model, tables can be related as any of:

- **many-to-many**
- **many-to-one** (rev. **one-to-many**)
- **one-to-one**

This is said to be the **cardinality** of a given table in relation to another.

For example, considering a database designed to keep track of hospital records. Such a database could have many tables like:

- a *Doctor* table full of doctor information
- a *Patient* table with patient information
- and a *Department* table with an entry for each department of the hospital.

In that model: