**TASK 3**

**A real time application involving joins**

My dad works in a perfume company that manufactures perfumes. The company collects orders from different retail and wholesale stores and delivers perfumes there. They then pay for the received perfumes.

Joins can be used such that for ever order of perfumes, we can create a table for new orders and match it with a table of previous customers to know if the customer has placed an order before, to check if the person is eligible for a discount.

* TABLE 1:  This table has columns for the customer's name and ID number unique for each customer.

|  |  |
| --- | --- |
| **Customer Name** | **Customer ID** |
|  |  |

* TABLE 2: This table has columns for the order ID number, the customer ID number.

|  |  |
| --- | --- |
| **Order ID Number** | **Customer ID** |
|  |  |

To check if the customer has placed orders, the join criteria is to join the two tables on the customer ID number. This means that we only want to return rows from the table 2 where the user ID number matches a row in the table 1. If so, a discount of a fixed small amount may be given to them.

NORMALIZATION EXAMPLES

**Example 1 (1NF)**

Every attribute in the relation is **singled valued. In the given example, even though the same person got placed at 2 different companies, the tables should be designed such that each name corresponds to single company in the same row.**

|  |  |  |
| --- | --- | --- |
| Sl. No | Name | Placement |
| 1 | Sherin Xavier | Tarento |
| 1 | Sherin Xavier | IBM |
| 2 | Alona Titty | Cognizant |

**Example 2 (2NF)**

Suppose that a student can choose more than one project for final year and based on the topic of project, guides are assigned. Then it is easier to separate out the topics and guides to eliminate redundancy.

**Table 1**

|  |  |
| --- | --- |
| Student Name | Project Topic |
| Sherin Xavier | Machine Learning |
| Sherin Xavier | IOT |
| Alona Titty | Machine Learning |

**Table 2**

|  |  |
| --- | --- |
| Project Topic | Project Guide |
| Machine Learning | Guide 1 |
| IOT | Guide 2 |

**Example 3 (3NF)**

Suppose the student ID, name, project topics and department needs to be tabulated and there are a set of topics that come under the same department, then the normalization is done as follows.

**Table 1**

|  |  |  |
| --- | --- | --- |
| Student ID | Student Name | Project Topic |
| 1 | Sherin Xavier | Machine Learning |
| 2 | Julien David | IOT |
| 3 | Alona Titty | Automation |

**Table 2**

|  |  |
| --- | --- |
| Project Topic | Department |
| Machine Learning | ECE |
| IOT | ECE |
| Automation | ME |

**Example 4 (BCNF)**

Table should be in 3NF and functional dependency X−>Y, X must be in Super Key.

**Table 1**

|  |  |
| --- | --- |
| Student Name | Project Topic |
| Sherin Xavier | Machine Learning |
| Sherin Xavier | IOT |
| Alona Titty | Automation |
| Alona Titty | IOT |

**Table 2**

|  |  |
| --- | --- |
| Project Topic | Project Guide |
| Machine Learning | Guide 1 |
| IOT | Guide 2 |
| Automation | Guide 3 |

**Example 5 (4NF)**

A relation can contain a functional dependency along with a multi-valued dependency.

**Table 1**

|  |  |
| --- | --- |
| Student ID | Course |
| 100 | Machine Learning |
| 100 | IOT |
| 102 | Automation |
| 102 | Cloud computing |

**Table 2**

|  |  |
| --- | --- |
| Student ID | Organization |
| 100 | IEEE |
| 100 | IEDC |
| 102 | IEEE |
| 102 | IEDC |

**Example 6 (5NF)**

5NF is satisfied when all the tables are broken into as many tables as possible in order to avoid redundancy.

**Table 1**

|  |  |
| --- | --- |
| Student ID | Project Topic |
| 100 | Machine Learning |
| 100 | IOT |
| 102 | Automation |

**Table 2**

|  |  |
| --- | --- |
| Student ID | Project Guide |
| 100 | Guide 1 |
| 100 | Guide 2 |
| 102 | Guide 3 |

**Table 3**

|  |  |
| --- | --- |
| Project Topic | Project Guide |
| Machine Learning | Guide 1 |
| IOT | Guide 2 |
| Automation | Guide 3 |