**Database anomalies and Normalization**

**If there is data redundancy in a relation , it can lead to various types of data update problems.**

**These problems can be classified into following three categories-**

1. **Insertion Anomaly**
2. **Deletion Anomaly**
3. **Modification Anomaly**

**Company**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Emp\_Id** | **Emp\_Name** | **Dept\_Id** | **Dept\_Name** | **Dept\_Location** |
| E1001 |  | D001 | Purchase | Floor 1 |
| E1002 |  | D002 | Sales | Floor 2 |
| E1003 |  | D001 | Purchase | Floor 1 |
| E1004 |  | D002 | Sales | Floor 2 |
| E1005 |  | D003 | Manufacturing | Floor 3 |

**Insertion Anomaly**

In case some data cannot be inserted in a tuple without the pre existence of some other data in the tuple, then this type of problem is called incursion anomaly. In this example relation we can not insert the data for a new department till a particular employee joins the department. The reason is that , in such a case the primary key value for Emp\_Id will have a NULL. Which is not allowed as per entity integrity rule.

**Deletion Anomaly**

This is the reverse problem of incursion anomaly. In case the deletion of some data leads to the permanent deletion of some other data in the database , then this type of problem is called deletion anomaly. In the example relation , if there is a single employee in a department and he leaves the company , then by deleting the tuple of that employee , all the data related to the particular department will be deleted permanently.

**Modification Anomaly**

In case modification of some data in the database leads to inconsistent data , then this type of problem is called modification anomaly. In the example relation , if we want to change location of a particular department, then we have to modify all tuples that has that department location. If we miss to modify some tuples then the relation will store two different location of same department. This will lead to inconsistent data in the database.

**Normalization**

To make a relation with consistent data and without unnecessary data redundancy , it need to be tested for certain requirements on primary key and other functional dependencies that exist in the relation. This testing lead to a process called normalization.

**Normalization is the process of taking a relation through a series of tests to ensure data integrity and eliminate insertion , deletion and modification anomalies and other problems related to data redundancy. The tests are done based on the primary key and other functional dependencies in the relation.**

**The result of normalization usually leads to breaking up a single table into multiple smaller tables with commonly related information. Each breaking process leads to a particular normal form. which basically indicates how much redundancy is left in the table. These normal forms are expressed as 1NF , 2NF , 3NF , BCNF, 4NF and 5NF.**

**The First Normal Form ( 1NF )**

**For a relation to be in the first normal form , every attribute value in it should be atomic (non decomposable ) and repeating groups should not be present in the relation.**

**The steps required to convert a table to 1NF are :**

1. Decomposing all Composite attributes .
2. In case of multi valued attributes , split each tuples into multiple tuples .
3. Combine the original primary key of the relation with the multi valued attributes to form a unique primary key for the new relation

**The Second Normal Form ( 2NF )**

**A relation is said to be in the Second normal form , if it is in 1NF and non key attribute(s) in the relation are fully functionally dependent on the primary key of the relation.**

**The steps required to convert a table to 2NF are :**

1. Find and remove the attributes from the original relation that are related to only one part of the primary key . Group the removed attributes in a new relation .
2. If more than one partial functional dependency prevents the relation from being in 2NF , then repeat the previous process for each FD.
3. Copy the composite primary key component on which the non key attributes depended partially . Make this attribute the primary key of the new relation .

**The Third Normal Form ( 3NF )**

**A relation is said to be in the Third normal form , if it is in 2NF and each non key attribute is non – transitively dependent on the primary key .**

**The steps required to convert a table to 3NF are :**

1. Find and remove the non-key attributes that functionally depend on another non key attribute(s) from the original relation , and group the removed attributes in a new relation .
2. Copy the determining non key attribute (s) from the original relation and make it the primary key of the new relation.