

Anomalies in DBMS

A database anomaly is a data inconsistency caused by an operation such as an insertion, deletion, or update. When a record is held in multiple locations, and not all copies are updated, inconsistencies can occur.

Anomalies in DBMS Definition

Anomalies in DBMS occur when the data in the database contains too much redundancy and the tables that comprise the database are poorly constructed.

What Causes Anomalies in DBMS?

Poorly constructed tables in a database are commonly the main cause behind anomalies. Poor construction means when a designer creates a database, he fails to identify interdependent entities, such as rooms in a hostel and the hostel, and then minimizes the chances of one entity being independent of another.

Database anomalies in DBMS are a weakness within a database that can result from poor planning or when all data is stored in a flat database. Typically, a normalization procedure that combines and splits tables is sufficient to remove this. We reduce the likelihood of creating tables that generate anomalies by normalizing the database.

Types of Anomalies in DBMS

If a database design isn't perfect, it may contain anomalies, a database administrator's major headache. It is nearly impossible to manage a database that contains anomalies. There are various types of anomalies in DBMS that are as follows:

- Insert Anomalies in DBMS
- Update Anomalies in DBMS
- Delete Anomalies in DBMS

Insert Anomalies in DBMS

When certain attributes cannot be inserted into the database without the presence of other attributes, this is referred to as an Insert Anomalies in DBMS. For example, suppose there is a student database with four columns: Rollno for the student's id, Name for the student's name, Address for the student's address, and Club for the student's club. Eventually, the table will appear as follows:

Rollno	Name	Address	Club
220	Annamalai	Kerala	yoga

Rollno	Name	Address	Club
220	Muthu	Kerala	Music
231	Mukesh	Mumbai	Crypto
232	Muni	Karnataka	Public Speaking
232	Muni	Karnataka	Arts

If a new student named Nanda joins the college and has no department affiliation because the club only accepts students in their second year, he will appear in the above table. Then we can't insert Nanda's data into the table because the Club field doesn't accept null values.

Update Anomalies in DBMS

An Update Anomalies in DBMS occur when one or more instances of duplicated data are updated but not all. For example, suppose there is a database of rooms that contains four columns: Courseno for the course type, Name for the tutor's name, Rollno for the room allocated, and Room_size for the room space. Eventually, the table will appear as follows:

Courseno	Name	Roomno	Room_size
220	Annamalai	10	50
220	Muthu	11	150
231	Mukesh	15	250
232	Muni	15	250

Roomno 15 has been improved; it is now Room_size = 500. For updating a single entity, we have to update all other columns where Roomno 15.

Delete Anomalies in DBMS

When certain attributes are lost due to the deletion of other attributes, this is referred to as Delete Anomalies in DBMS. For example, suppose there is a database of rooms that contains four columns: Courseno for the course type, Name for the tutor's name, Rollno for the room allocated, and Room_size for the room space. Eventually, the table will appear as follows:

Courseno	Name	Roomno	Room_size
220	Annamalai	10	50

Courseno	Name	Roomno	Room_size
220	Muthu	11	150
231	Mukesh	15	250
232	Muni	15	250

If we remove the entity courseno: 231 from the above table, the details of roomno 15 get deleted. This implies the corresponding course will also get deleted.

How to Remove Anomalies in DBMS?

To avoid anomalies in DBMS, normalize the database by efficiently organizing data in a database.

The following are the goals of normalization, according to Edgar F Codd, the inventor of relational databases:

- removing all duplicate (or redundant) data from the database
- removing unnecessary insertions, updates, and deletions anomalies in DBMS
- reducing the need to restructure the entire database whenever new fields are added
- making table relationships more useful and understandable