

Database Quality Risks and Examples.

- **Duplicate Data.**

This happens when the same information is stored in multiple places in a database. For example, if the city and country of an airport are stored both in the Airport Table and again in the Flight table, any update to the airport's city would have to be made in every related flight record.

- **Update Anomalies.**

This occurs when changes to a piece of data are not applied at all copies, leading to inconsistencies in the database. For example, if the timeArrival is stored in Flight Table and FlightPlan table, updating it in one place but not in the other one could make it seem like the timeArrival is both valid at the same time.

- **Insertion Anomalies.**

It is when new data can't be added to the database due to unnecessary dependencies between the tables. For instance, if a FlightCrew record requires a flightID, and the system does not allow a crew to be registered unless a flight already exists, you would not be able to record a crew member's availability until a flight is created.

- **Deletion Anomalies.**

Occur when deleting a record (without intentionally) removes other important data. For example, deleting a Flight entry might also delete a FlightPlan, causing the system to lose information.

¿How Normalization Helps?

Normalization is a process of structuring data into related tables to remove the redundancy, maintaining the data integrity, and minimizing anomalies. It ensures that each attribute has a single, consistent source of truth, making updates predictable and allowing new data to be inserted or deleted without affecting unrelated information. Applying normalization, our database is now organized and easy-managed.