# Mini project 4

1. What is normalization?

Normalization is a systematic and structured process in database design that aims to reduce data redundancy and improve data integrity by organizing data into well-structured tables and defining relationships between them, ensuring that each piece of data is stored in only one place and that dependencies between data are accurately represented. This process minimizes the likelihood of data anomalies and inconsistencies, leading to more efficient and reliable databases.

1. When is a table in 1NF?

A table is considered to be in First Normal Form (1NF) when it meets the following criteria:Atomic Values: Each column in the table contains only atomic (indivisible) values. Atomic values are values that cannot be further subdivided. This means that each cell in the table should hold a single value, not a list, array, or composite data type. Unique Column Names: Each column in the table has a unique name. There should be no duplicate column names within the same table. Order-Independent: The order of rows and columns should not affect the interpretation of data. In other words, the data in the table should be unordered, and the values in each column should be unrelated to the values in other columns. Meeting these criteria ensures that the table is in 1NF, which is the fundamental requirement for proper database organization.

1. When is a table in 2NF?

To be in 2NF, a table must: Already be in 1NF: This means that the table should have atomic values (indivisible values) in each column, unique column names, and data that is order-independent. Have a Primary Key: The table must have a primary key, which is a unique identifier for each row in the table. Not Contain Partial Dependencies: Partial dependencies occur when a non-prime attribute (an attribute not part of the primary key) depends on only a part of the primary key. In a table in 2NF, there should be no partial dependencies. All non-prime attributes should be fully functionally dependent on the entire primary key. To eliminate partial dependencies and achieve 2NF, you may need to split the table into multiple related tables, where non-prime attributes depend on the entire primary key rather than just a part of it.

1. When is a table in 3NF?

To be in 3NF, a table must: Already be in 1NF and 2NF: This means that the table should have atomic values in each column, unique column names, a primary key, and no partial dependencies. Not Contain Transitive Dependencies: Transitive dependencies occur when a non-prime attribute (an attribute not part of the primary key) depends on another non-prime attribute, rather than directly on the primary key. In a table in 3NF, there should be no transitive dependencies. To achieve 3NF, you need to ensure that all non-prime attributes are functionally dependent on the primary key and not on each other. This typically involves breaking down the table into multiple related tables, where the relationships between the attributes are clearly defined and follow the rules of 3NF.

v.) When is a table in BCNF?

To be in BCNF, a table must: Not Contain Non-Trivial Functional Dependencies: The table should not have any non-trivial functional dependencies of non-prime attributes on the primary key. In other words, for any non-prime attribute (an attribute not part of the primary key), it should be functionally dependent on the whole primary key and not on a subset of the primary key. Achieving BCNF often involves further decomposition of tables to remove these non-trivial functional dependencies. This process ensures that all attributes in the table are functionally dependent on the primary key and not on other non-key attributes. The result is a highly normalized database structure that minimizes data redundancy and anomalies. BCNF is considered a more advanced level of normalization than Third Normal Form (3NF) and is appropriate for databases where maintaining data integrity and minimizing anomalies are of utmost importance. However, achieving BCNF may lead to a more complex schema with more tables and relationships, which can have implications for query performance. Therefore, the decision to go beyond 3NF to BCNF should be made based on the specific needs and constraints of the database system.

1. When is a table in the fourth Normal Form?

It does not contain any multi-valued dependencies.

Multi-valued dependencies occur when an attribute is dependent on a combination of values in another attribute, and this dependency is not fully explained by the primary key. In other words, a table is in 4NF when it ensures that no non-key attribute depends on a subset of the primary key. However, it can still have attributes that depend on multiple values within another non-key attribute without causing redundancy or anomalies. Achieving 4NF often requires further normalization and decomposition of tables to eliminate multi-valued dependencies. While 4NF helps to minimize data anomalies, it is a level of normalization that is not always necessary for all database designs. Whether to pursue 4NF depends on the specific requirements of the application and the trade-offs between maintaining data integrity and query performance, as achieving 4NF can sometimes lead to more complex schemas and potentially slower query execution.

1. Convert the above students table to the first Normal form (1NF). Explain your answer

A screenshot of a computer

Description automatically generated

The original table is not in 1NF because the Pet column is multi-valued. This means that a single cell in the Pet column can hold multiple values. For example, Jennifer has both a dog and a cat, so her cell in the Pet column contains the values "Dog, Cat".

A table is in 1NF if it meets the following criteria:

Each cell in the table contains a single value.

Each column in the table has a unique name.

There is no partial dependency between any non-key attribute and the primary key.

8.) Convert the above student table to the second Normal form (2NF). Explain your answer.

A screenshot of a cell phone

Description automatically generated

To convert the table in the example to 2NF, we need to remove all partial dependencies. A partial dependency occurs when a non-key attribute is dependent on only a part of the primary key.

In the example table, the Pet Name column is partially dependent on the primary key (Name). This is because a student's pet name can vary depending on the pet. For example, Jennifer's dog is named Rex, but her cat is named Bushy.

To remove the partial dependency, we need to create a new table for the pet names. This new table will have two columns: Pet Name and Pet ID. The Pet ID column will be a foreign key that references the primary key (Name) of the student table.

9.) . Convert the above student table to the third Normal form (3NF). Explain your answer

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

This new table is in 3NF because it meets all of the following criteria:

It is in 2NF.

There are no transitive dependencies between any non-key attribute and the primary key.

10.) Convert the above student table to the Fourth Normal Form (4NF). Explain your answer.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

This new table is in 4NF because it meets all of the following criteria:

It is in 3NF.

There are no multi-valued dependencies.

11.) Create a trigger for Bank transactions (withdrawals and deposits). The bank account tables are given below.

A screenshot of a computer

Description automatically generated

A white background with black and orange text

Description automatically generated