NORMALIZATION

Normalization is a process used in database design to eliminate redundancy and ensure data integrity. It involves organizing data into separate tables and establishing relationships between them

Why do we need Normalization?

The main reason for normalizing the relations is removing these anomalies. Failure to eliminate anomalies leads to data redundancy and can cause data integrity and other problems as the database grows. Normalization consists of a series of guidelines that helps to guide you in creating a good database structure.

Data modification anomalies can be categorized into three types:

• **Insertion Anomaly:** Insertion Anomaly refers to when one cannot insert a new tuple into a relationship due to lack of data.

• **Deletion Anomaly:** The delete anomaly refers to the situation where the deletion of data results in the unintended loss of some other important data.

• **Updation Anomaly:** The update anomaly is when an update of a single data value requires multiple rows of data to be updated

First Normal Form (1NF)

1. First Normal Form requires that each column in a table contains only atomic (indivisible) values.
2. It eliminates duplicate rows by ensuring that each row has a unique identifier, typically a primary key.

1. **Entity:** Branch

    Branch(Branch_id,Name,Address)

    Functional Dependencies:

    branch_id → name,address

    Normal Form: This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.

2. **Entity**: Customer

    customer(customer_id,branch_id,loan_id,first_name,last_name,date_of_birth,gender)

Functional Dependencies:

    customer_id→branch_id,loan_id,first_name,last_name,date_of_birth,gender

Normal Form Violation:This violates 1NF

    This Entity is divided into 1.customer

                             2.customer_loan

➤ **Entity**: customer

   customer(customer_id,branch_id,first_name,last_name,date_of_birth,gender)

  Functional_Dependencies:

    Customer_id→branch_id,first_name,last_name,date_of_birth,gender

Normal Form:This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.

➤ **Entity**: customer_loan

    customer_loan(customer_id,loan_id)

  Functional Dependencies:

    customer_id → loan_id, loan_id →customer_id

  Normal Form:This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.

3. **Entity**: Account

    Account(account_id,balance,customer_id)

Functional Dependencies:

    account_id → balance,customer_id

Normal Form Violation:This violates 1NF

    This Entity is divided into 1.account

                             2.account_customer

- ➢ **Entity**: account

  account(account_id,balance)

  Functional Dependencies:

  account_id →balance

  Normal Form:This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.

- ➢ **Entity**: account_customer

  account(account_id,customer_id)

  Functional Dependencies:

  account_id →customer_id,customer_id→account_id

  Normal Form:This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.

4. **Entity**: Loan

   loan(loan_id,account_id,amount_paid,start_date,end_date)

   Normal Form Violation:This violates 1NF

   This Entity is divided into 1.loan

   2.loan_account

- ➢ **Entity**: loan

  loan(loan_id,amount_paid,start_date,end_date)

  Functional Dependencies:

  loan_id →amount_paid,start_date,end_date

  Normal Form:This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.

- ➢ **Entity**: loan_account

  Loan_account(loan_id,account_id)

  Functional Dependencies:

  loan_id→ account_id, account_id→ loan_id

  Normal Form:This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.

5. **Entity**: loan_type

loan_type(loan_type_id,loan_id,type,description,base_amount,base_interest_rate)

Functional Dependencies:

Loan_type_id,loan_id → type,description,base_amount,base_interest_rate

Normal Form: This is in 1NF because there are no repeating groups present in this table as each attribute or column contains atomic values.

6. **Entity**: card

card(card_id,account_id,card_number,expiration_date,is_blocked)

Functional Dependencies:

card_id → account_id,card_number,expiration_date,is_blocked

Normal Form: This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.

7. **Entity**: Transaction

Transaction(transaction_id,account_id,description,amount,tdate)

Functional Dependencies:

transaction_id → account_id,description,amount,tdate

Normal Form: This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.

8. **Entity**: account_loantype

account_loantype(account_id,loan_type_id)

Functional Dependencies:

account_id → loan_type_id, loan_type_id →account_id

Normal Form: This is in 1NF because there are no repeating groups present in this table and each attribute or column contains atomic values.