Inventory Management System

FUNCTIONAL DEPENDENCIES

To determine the functional dependencies in your project, we need to analyze the relationships between attributes within each table. These concepts help ensure data integrity, minimize redundancy, and optimize the structure of the database by establishing relationships between different attributes within the tables. Here’s a comprehensive look at the functional dependencies identified and the normal forms applied across the various tables in this project.

Functional Dependencies Identified Functional dependencies are vital for understanding how data in one attribute or a set of attributes uniquely determines data in another attribute or set of attributes. Here are key dependencies in the database:

**1.Items table**

**-Item\_name->supplier\_name**

**2.suppliers**

**-Supplier\_id->SupplierContactPhone**

**Unnormalised items tables**

A screenshot of a computer

Description automatically generated

**1. First Normal Form (1NF):**

**- Description: In 1NF, each attribute of a relation/table contains only atomic (indivisible) values, and there are no repeating groups or arrays of values.**

**- Application: Ensure that each column in the inventory table contains only single values, without any repeating groups or arrays.**

**2. Second Normal Form (2NF):**

**- Description: In 2NF, the table is in 1NF, and all non-key attributes are fully functional dependent on the entire primary key.**

**- Application: Decompose the inventory table into separate tables to eliminate partial dependencies and ensure that each non-key attribute is dependent on the entire primary key.**

**3. Third Normal Form (3NF):**

**- Description: In 3NF, the table is in 2NF, and all transitive dependencies are removed. Each non-key attribute is dependent only on the primary key, not on any other non-key attributes.**

**- Application: Further decompose tables to eliminate transitive dependencies and ensure data integrity by storing data in smaller, related tables.**

**4. Boyce-Codd Normal Form (BCNF):**

**- Description: BCNF is a stronger version of 3NF, where every determinant is a candidate key. It eliminates non-trivial functional dependencies where the determinant is not a superkey.**

**- Application: Analyze dependencies in the inventory tables and ensure that every determinant is a candidate key to remove anomalies and redundancy.**

**5. Fourth Normal Form (4NF):**

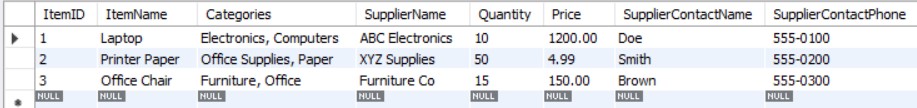
**- Description: In 4NF, a table is in BCNF, and multi-valued dependencies are removed. It ensures that there are no more than one multi-valued dependency in a table.**

**- Application: Identify and remove multi-valued dependencies in the inventory tables to maintain data integrity and eliminate redundancy.**

**Implementation of the normalization**

**1st normalization**

**Each atomic group contains atomic values. So we separated the supplier info that was conaining contact and name ,to 2 columns of supplier contact name and supplier contact phone.**



**2nd normalization**

**It requires that the table be in 1st Normal Form and that all non-key attributes be fully functionally dependent on the entire primary key. For a table that has only a simple (single column) primary key, like our items table, achieving 2NF primarily involves removing partial dependencies of non-key attributes on a part of any primary key.**

A screenshot of a computer

Description automatically generated