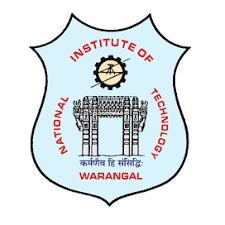
**DISTRIBUTOR–RETAILER MANAGEMENT SYSTEM**



**DATABASE MANAGEMENT SYSTEM PROJECT REPORT**

**By –**

**1)** Arindam Pattanayak

23ECB0B03

**2)** Shravan Suraj Bhangdiya

23ECB0B13

Department Of Electronics And Communication Engineering

NATIONAL INSTITUTE OF TECHNOLOGY (An Institute of National Importance)

WARANGAL

TELANGANA, 506004

July, 2025

**PROBLEM STATEMENT:**

The Distributor–Retailer Management System is designed to streamline and manage interactions between distributors and retailers, including product inventories, purchases, payments, and deliveries. The objective is to build a normalized relational database that ensures data integrity, supports financial tracking (balances, profits), and enables meaningful analytics through SQL queries. This project demonstrates core DBMS concepts such as schema design, normalization, foreign key constraints, and transactional data handling.

**ASSUMPTONS:**

* Each distributor and retailer is uniquely identified using a UUID-based primary key.
* Every product has a fixed base price, and distributors apply individual discount percentages to determine the final price.
* A distributor maintains inventory for only the products they supply.
* Retailers purchase from one distributor at a time per purchase transaction.
* A single purchase can include multiple products (purchase items).
* Retailer final balances are updated after each purchase and payment.
* Payments are made from retailers to specific distributors and recorded with timestamps.
* Deliveries are linked directly to purchases and handled by external delivery companies.
* Soft deletes (is\_deleted, deleted\_at) are used for logical deletion of distributors and retailers.
* Foreign key constraints with ON DELETE actions (CASCADE or SET NULL) ensure referential integrity on deletion.
* Final balance calculations are handled consistently using logical triggers or manual updates during transactions.
* Inserted sample data reflects realistic values, names, and financial transactions for demonstration purposes.

**TABLES:**

* 1. **distributors :-**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| distributor\_id | VARCHAR(36) | PRIMARY KEY, DEFAULT (UUID()) |
| distributor\_name | VARCHAR(255) | NOT NULL |
| distributor\_full\_name | VARCHAR(255) | NOT NULL |
| address | TEXT | NOT NULL |
| contact\_number | VARCHAR(10) | NOT NULL, UNIQUE |
| email | VARCHAR(100) | NOT NULL, UNIQUE |
| bank\_account\_number | VARCHAR(20) | NOT NULL, UNIQUE |
| is\_deleted | BOOLEAN | DEFAULT FALSE |
| deleted\_at | TIMESTAMP | NULL allowed |
| password | VARCHAR(255) | NOT NULL |

* 1. **retailers:-**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| retailer\_id | VARCHAR(36) | PRIMARY KEY, DEFAULT (UUID()) |
| retailer\_name | VARCHAR(255) | NOT NULL |
| retailer\_full\_name | VARCHAR(255) | NOT NULL |
| address | TEXT | NOT NULL |
| contact\_number | CHAR(10) | NOT NULL, UNIQUE |
| email | VARCHAR(100) | NOT NULL, UNIQUE |
| bank\_account\_number | VARCHAR(20) | NOT NULL, UNIQUE |
| opening\_balance | DECIMAL(10, 2) | NOT NULL, CHECK (opening\_balance ≥ 0) |
| final\_balance | DECIMAL(10, 2) | NOT NULL, CHECK |
| is\_deleted | BOOLEAN | DEFAULT FALSE |
| deleted\_at | TIMESTAMP | NULL allowed |
| password | VARCHAR(255) | NOT NULL |

* 1. **products :**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| product\_id | VARCHAR(36) | PRIMARY KEY, DEFAULT (UUID()) |
| product\_name | VARCHAR(255) | NOT NULL |
| company\_name | VARCHAR(255) | NOT NULL |
| base\_price | DECIMAL(10,2) | NOT NULL, CHECK (base\_price ≥ 0) |
| description | TEXT | NOT NULL |

* 1. **distributor\_product\_inventory:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| inventory\_id | VARCHAR(36) | PRIMARY KEY, DEFAULT (UUID()) |
| distributor\_id | VARCHAR(36) | NOT NULL, FOREIGN KEY → distributors(distributor\_id) ON DELETE CASCADE |
| product\_id | VARCHAR(36) | NOT NULL, FOREIGN KEY → products(product\_id) ON DELETE CASCADE |
| discount\_percentage | DECIMAL(5,2) | DEFAULT 0.00, CHECK (discount\_percentage ≥ 0 AND ≤ 100) |
| final\_price | DECIMAL(10,2) | NOT NULL, CHECK (final\_price ≥ 0) |
| stock\_quantity | INT | NOT NULL, CHECK (stock\_quantity ≥ 0) |
| purchase\_cost | DECIMAL(10,2) | NOT NULL, CHECK (purchase\_cost ≥ 0) |

* 1. **purchases :**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| purchase\_id | VARCHAR(36) | PRIMARY KEY, DEFAULT (UUID()) |
| retailer\_id | VARCHAR(36) | FOREIGN KEY → retailers(retailer\_id) ON DELETE SET NULL |
| distributor\_id | VARCHAR(36) | FOREIGN KEY → distributors(distributor\_id) ON DELETE SET NULL |
| total\_amount | DECIMAL(10,2) | NOT NULL, CHECK (total\_amount ≥ 0) |
| purchase\_date | DATE | NOT NULL |
| retailer\_name | VARCHAR(255) | Optional |
| retailer\_email | VARCHAR(100) | Optional |
| contact\_number | VARCHAR(15) | Optional |
| final\_balance | DECIMAL(10,2) | NOT NULL |

* 1. **purchase\_items:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| purchase\_item\_id | VARCHAR(36) | PRIMARY KEY, DEFAULT (UUID()) |
| purchase\_id | VARCHAR(36) | FOREIGN KEY → purchases(purchase\_id) ON DELETE SET NULL |
| inventory\_id | VARCHAR(36) | FOREIGN KEY → distributor\_product\_inventory(inventory\_id) ON DELETE SET NULL |
| quantity | INT | NOT NULL, CHECK (quantity > 0) |
| price | DECIMAL(10,2) | NOT NULL, CHECK (price ≥ 0) |

* 1. **payments :**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| payment\_id | VARCHAR(36) | PRIMARY KEY, DEFAULT (UUID()) |
| retailer\_id | VARCHAR(36) | FOREIGN KEY → retailers(retailer\_id) ON DELETE SET NULL |
| distributor\_id | VARCHAR(36) | FOREIGN KEY → distributors(distributor\_id) ON DELETE SET NULL |
| amount\_given | DECIMAL(10,2) | NOT NULL, CHECK (amount\_given ≥ 0) |
| final\_balance | DECIMAL(10,2) | NOT NULL |
| payment\_method | VARCHAR(50) | NOT NULL |
| payment\_status | TEXT | NOT NULL |
| payment\_date | DATE | NOT NULL |
| created\_at | TIMESTAMP | DEFAULT CURRENT\_TIMESTAMP |
| retailer\_name | VARCHAR(255) | NOT NULL |
| retailer\_email | VARCHAR(100) | NOT NULL |
| contact\_number | VARCHAR(15) | NOT NULL |
| address | TEXT | NOT NULL |
| distributor\_name | VARCHAR(255) | NOT NULL |

* 1. **deliveries :**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| delivery\_id | VARCHAR(36) | PRIMARY KEY, DEFAULT (UUID()) |
| purchase\_id | VARCHAR(36) | NOT NULL, FOREIGN KEY → purchases(purchase\_id) ON DELETE CASCADE |
| delivery\_date | DATE | NOT NULL |
| status | VARCHAR(50) | NOT NULL |
| delivery\_company | VARCHAR(100) | NOT NULL |

**FUNCTIONAL DEPENDENCIES AND PRIMARY KEY:**

* **Distributors :-**

**Functional Dependencies:**

distributor\_id → distributor\_name, distributor\_full\_name, address, contact\_number, email, bank\_account\_number, is\_deleted, deleted\_at, password

* Since , *distributor\_id* is a unique identifier (e.g., UUID or manually generated unique ID) assigned to each distributor. This is the Primary Key.
* **Retailers:-**

**Functional Dependencies:**  
retailer\_id → retailer\_name, retailer\_full\_name, address, contact\_number, email, bank\_account\_number, opening\_balance, final\_balance, is\_deleted, deleted\_at, password

* Since *retailer\_id* uniquely identifies each retailer.So This is the Primary Key.
* **Products:-**

**Functional Dependencies:**  
product\_id → product\_name, company\_name, base\_price, description

* Since , *product\_id* uniquely distinguishes each product, even with similar names. This is the Primary Key.
* **Distributor\_product\_inventory :-**

**Functional Dependencies:**  
inventory\_id → distributor\_id, product\_id, discount\_percentage, final\_price, stock\_quantity, purchase\_cost

* One distributor may sell the same product at different prices over time or maintain separate inventory batches*. inventory\_id* ensures each such entry is uniquely tracked.so this is the Primary Key.
* **Purchases :-**

**Functional Dependencies:**purchase\_id → retailer\_id, distributor\_id, total\_amount, purchase\_date, retailer\_name, retailer\_email, contact\_number, final\_balance

* Each transaction between a retailer and distributor is uniquely identified using *purchase\_id*. A retailer may make multiple purchases from the same distributor purchase\_id distinguishes them. so this is the Primary Key.
* **Purchase\_items :**

**Functional Dependencies:**  
purchase\_item\_id → purchase\_id, inventory\_id, quantity, price

* A single purchase (e.g., purchase\_id = P1) may have multiple items (inventory\_id), possibly even repeated with different quantities. purchase\_item\_id uniquely identifies each line item. A composite key (purchase\_id, inventory\_id) might cause issues with multiple entries for the same inventory. so *purchase\_item\_id* is the Primary Key.
* **Payments:**

**Functional Dependencies:**  
payment\_id → retailer\_id, distributor\_id, amount\_given, final\_balance, payment\_method, payment\_status, payment\_date, created\_at, retailer\_name, retailer\_email, contact\_number, address, distributor\_name

* One retailer can make multiple payments to a distributor over time. *payment\_id* ensures each payment entry is uniquely identified and traceable. so this is the Primary Key.
* **Deliveries:**

**Functional Dependencies:**  
delivery\_id → purchase\_id, delivery\_date, status, delivery\_company

* Each delivery may be tied to a single purchase\_id but can occur in different batches or dates. *delivery\_id* provides a unique record for tracking delivery history and issues. . so this is the Primary Key.

**NORMAISATION:**

* 1. **distributors:**
* **1NF**: All fields contain atomic values (no multivalued attributes like contact\_number or address lists).
* **2NF**: All non-key attributes depend fully on the primary key distributor\_id (no partial dependency).
* **3NF**: There are no transitive dependencies (e.g., email is not dependent on contact\_number, etc.).
* **BCNF**: distributor\_id is the only candidate key and determines all other fields — thus satisfies BCNF.

**2. retailers:**

* **1NF**: All attributes are atomic.
* **2NF**: Since there's only one attribute (retailer\_id) as primary key, all non-key fields depend fully on it.
* **3NF**: No transitive dependencies (e.g., final\_balance isn't derived from opening\_balance; it is updated independently).
* **BCNF**: retailer\_id is the only candidate key and determines all attributes.

**3. products :**

* **1NF**: All attribute values are atomic.
* **2NF:** Single-attribute primary key, so all non-key attributes depend fully on product\_id.
* **3NF**: No derived/transitive attributes.
* **BCNF**: Only candidate key is product\_id; thus satisfies BCNF.

4**. distributor\_product\_inventory :**

* **1NF**: All attributes are scalar and atomic.
* **2NF**: Every non-key attribute is fully dependent on the primary key (inventory\_id).
* **3NF**: No transitive dependencies (e.g., final\_price is not derived from stock\_quantity, but discount is applied directly to base\_price).
* **BCNF**: inventory\_id is the only candidate key.

**5. Purchases :**

* **1NF**: Atomic fields (purchase\_date, total\_amount, etc.).
* **2NF**: All non-key attributes depend entirely on purchase\_id.
* **3NF**: retailer\_name, email, contact\_number, etc. are denormalized intentionally — but not derived from other fields in this table (we assume historical snapshot values).
* **BCNF**: purchase\_id is the only candidate key and all attributes depend on it.

**6. purchase\_items:**

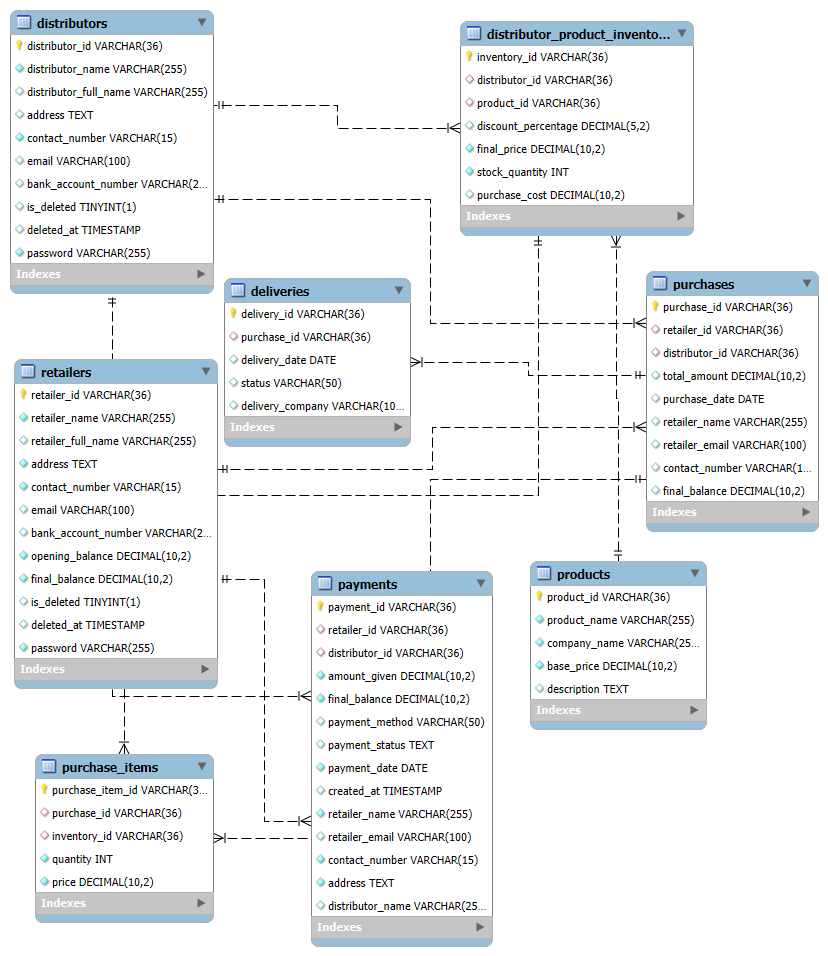
* **1NF**: No multivalued or nested fields.
* **2NF**: purchase\_item\_id is the PK and all other fields depend on it fully.
* **3NF**: No transitive dependency — price is stored per item and not computed from another column in the same table.
* **BCNF**: All functional dependencies have a superkey (purchase\_item\_id), so it satisfies BCNF.

**7. payments :**

* **1NF**: Atomic attributes (email, address, amount\_given, etc.)
* **2NF**: All attributes are fully dependent on the PK (payment\_id)
* **3NF**: Though retailer\_name and distributor\_name are derivable from foreign tables, they are stored as point-in-time snapshots and not transitively dependent within this table.
* **BCNF**: payment\_id is the only candidate key and determines all attributes.

**8.deliveries:**

* **1NF**: All fields contain atomic values
* **2NF**: All non-key attributes depend fully on delivery\_id
* **3NF**: No attribute is transitively dependent
* **BCNF**: delivery\_id is the only candidate key and determines all fields
* **RELATIONAL SCHEMA:**



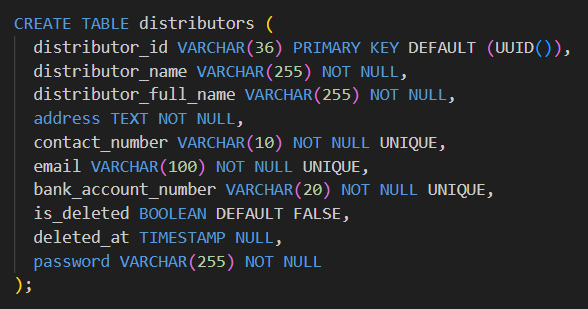
**MYSQL CODES:**

* **Creating Tables:**

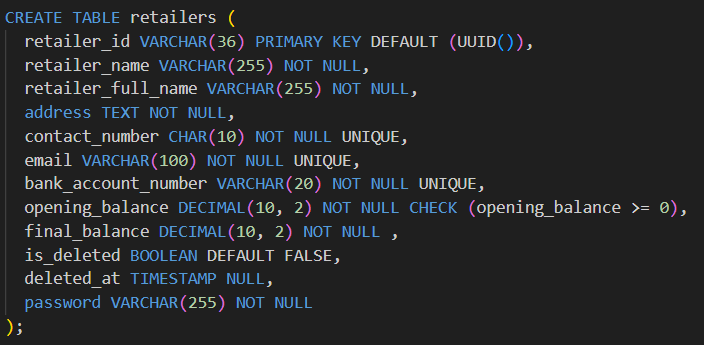
**CREATE DATABASE IF NOT EXISTS distributor\_retailer\_management\_system;**

**USE distributor\_retailer\_management\_system;**

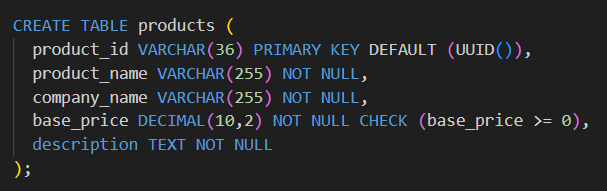
**1.distributors:**



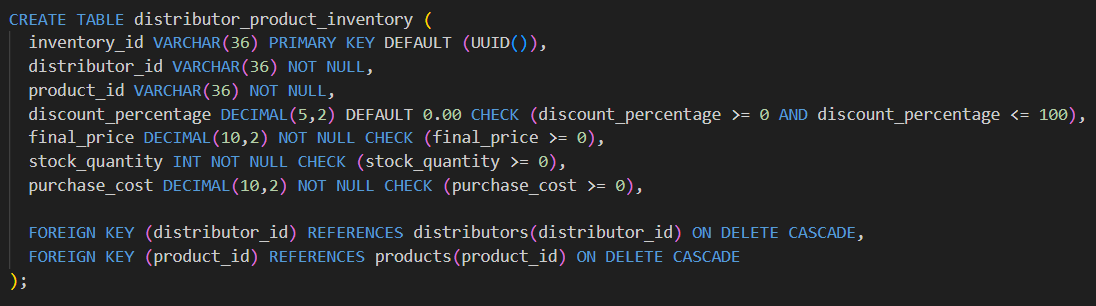
**2.retailers:**



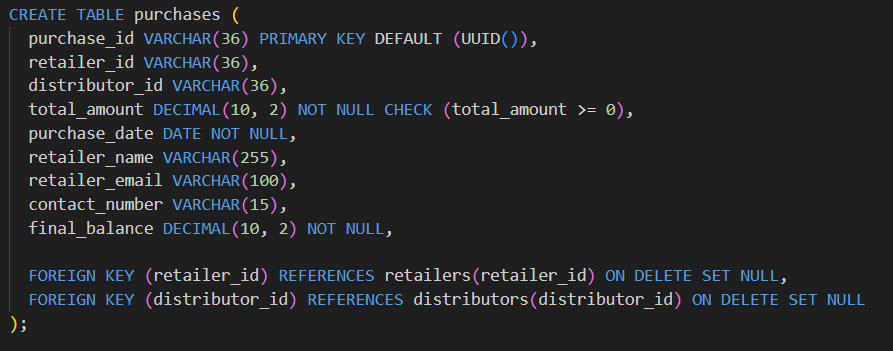
**3.products:**



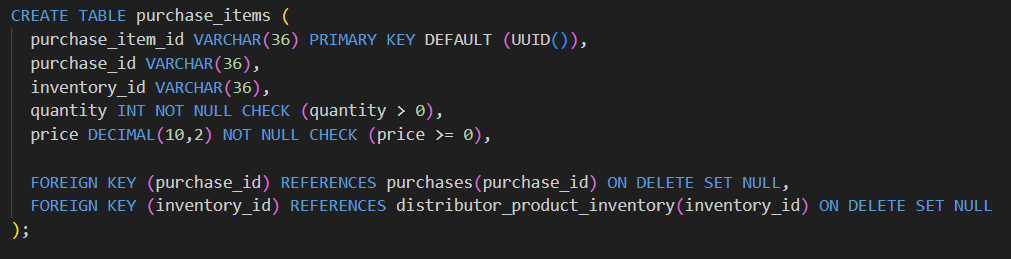
**4. distributor\_product\_inventory :**



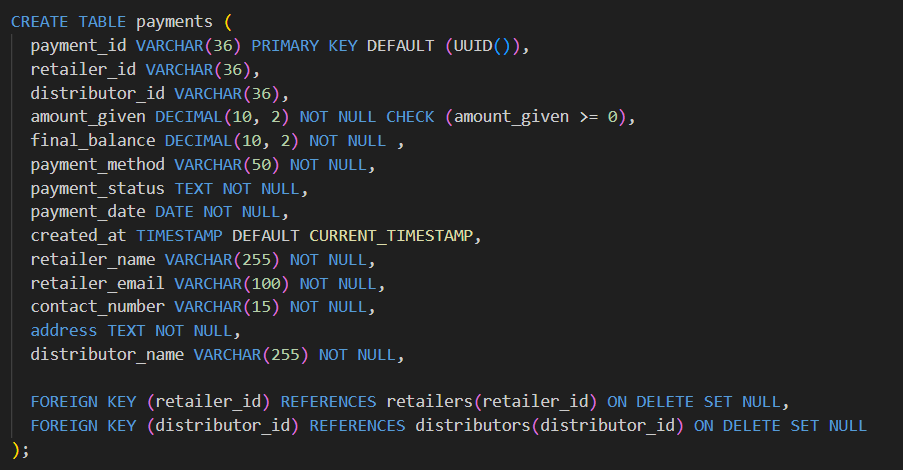
**5. purchases:**



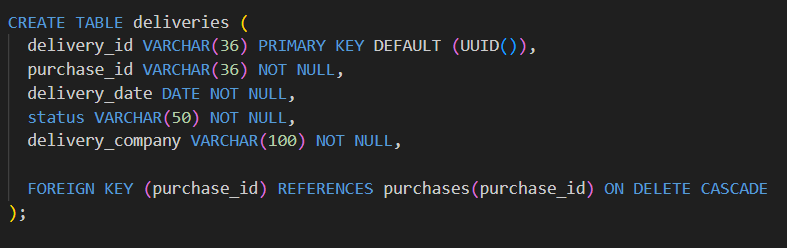
**6. purchase\_items :**



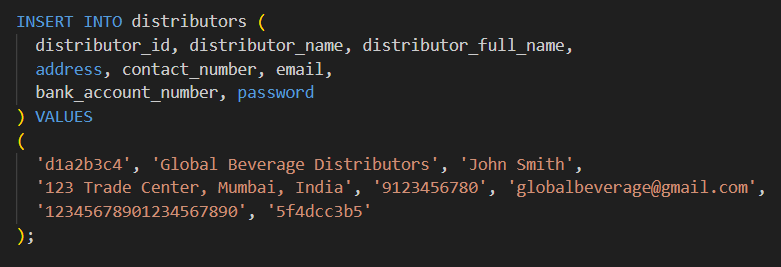
**7. payments :**

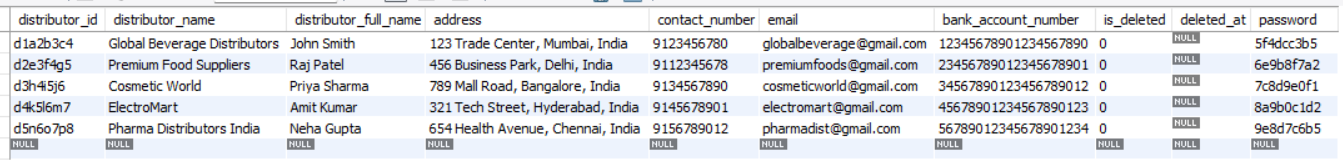


**8.deliveries :**

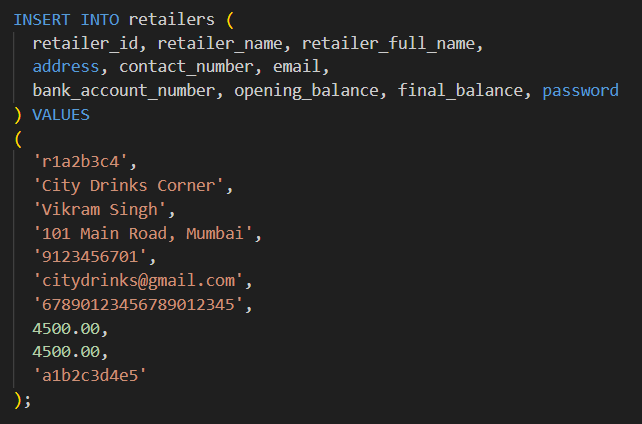
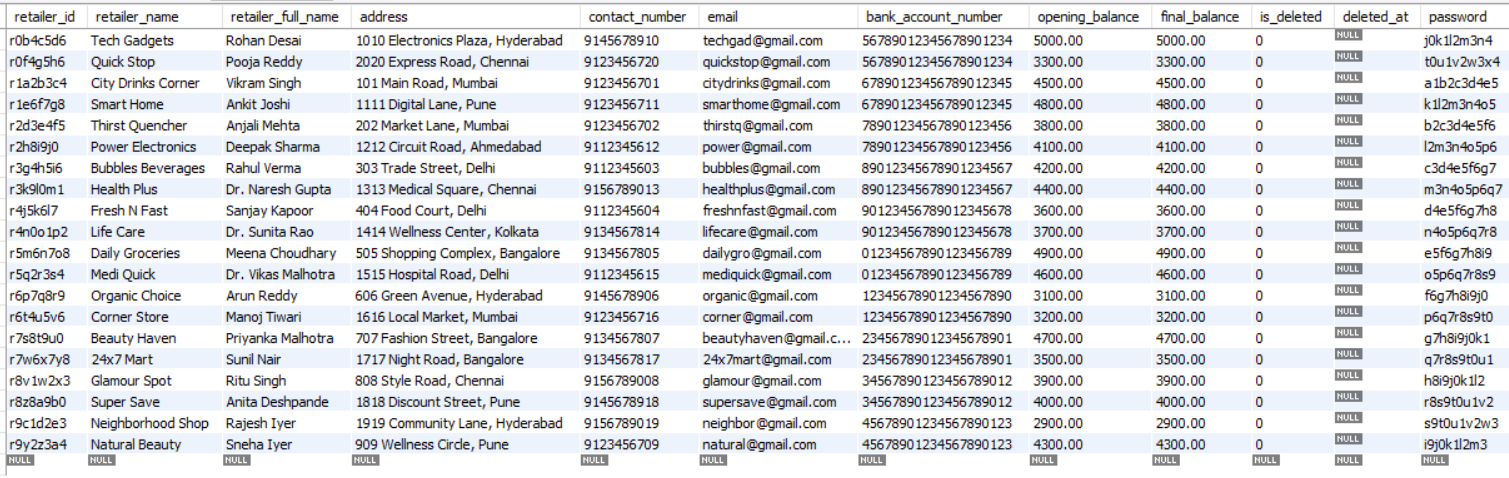


* **INSERTING DATA:**
  1. **Distributors :**

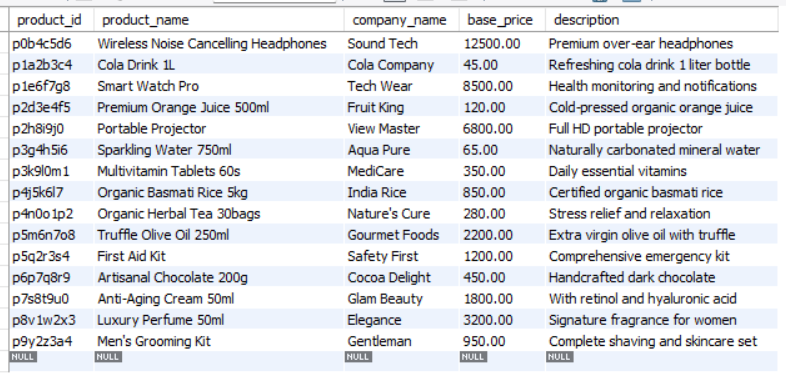
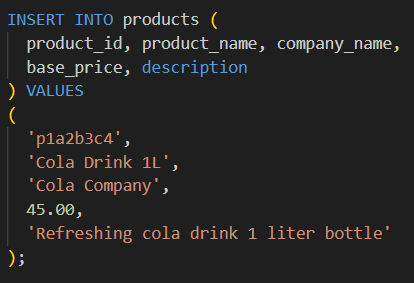




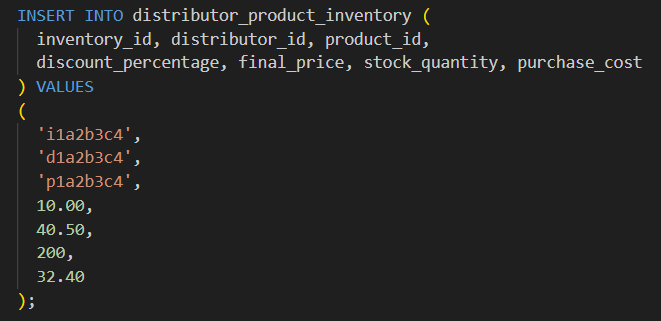
* 1. **Retailers:**

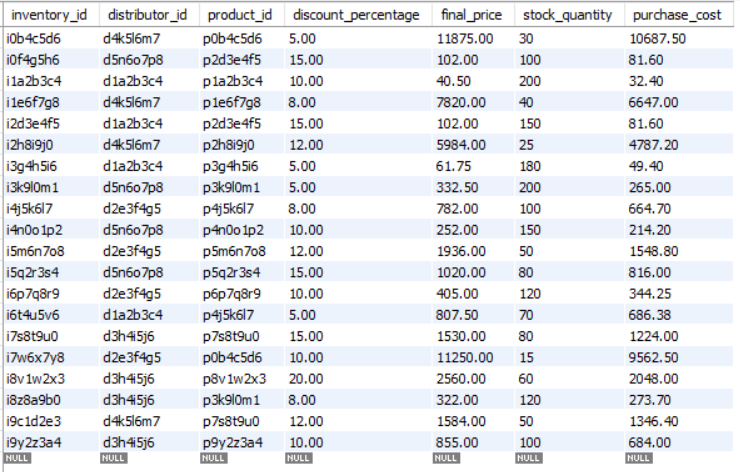


* 1. **Products :**

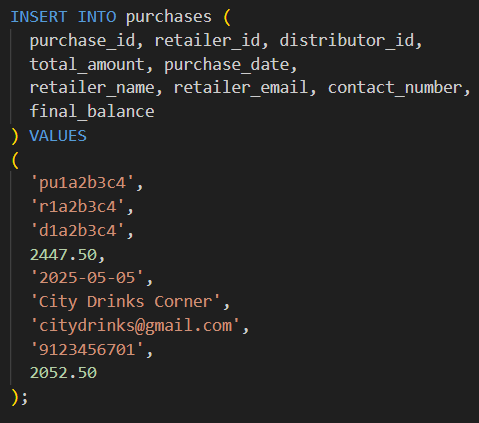


* 1. **distributor\_product\_inventory :**

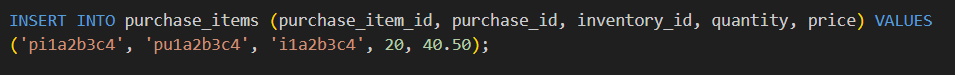


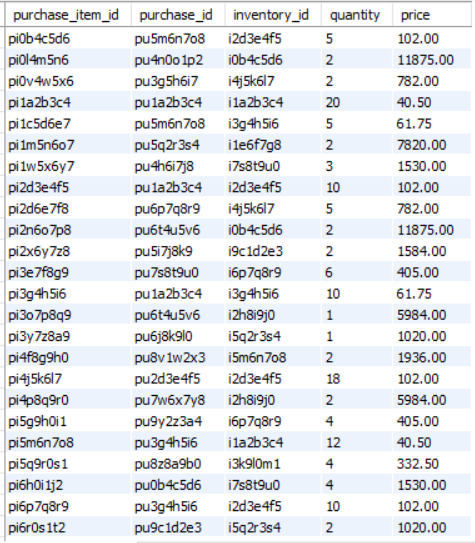


* 1. **purchases :**

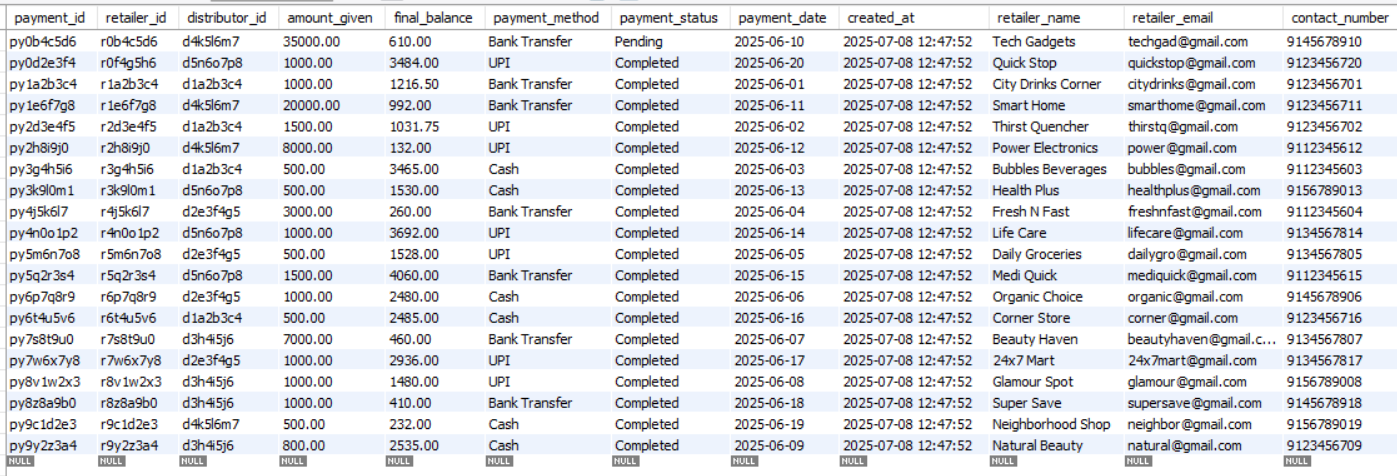


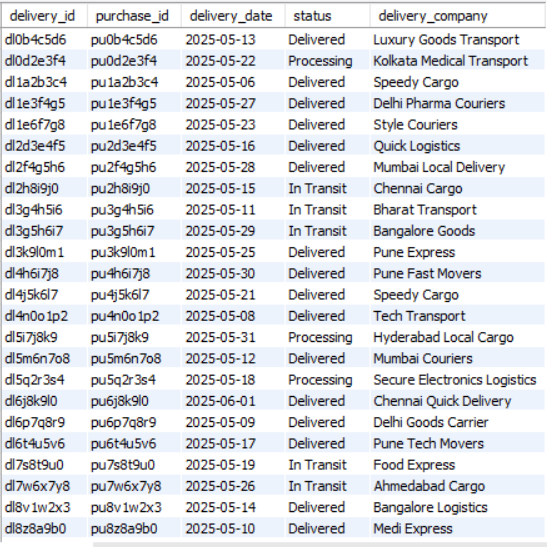
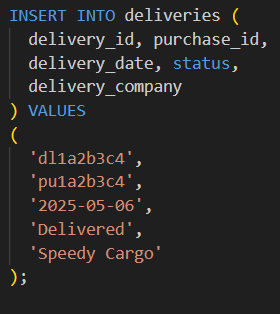


* 1. **purchase\_items:**

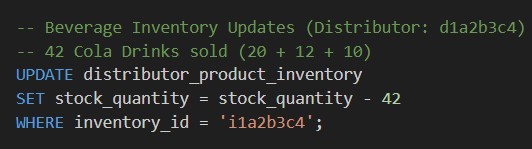


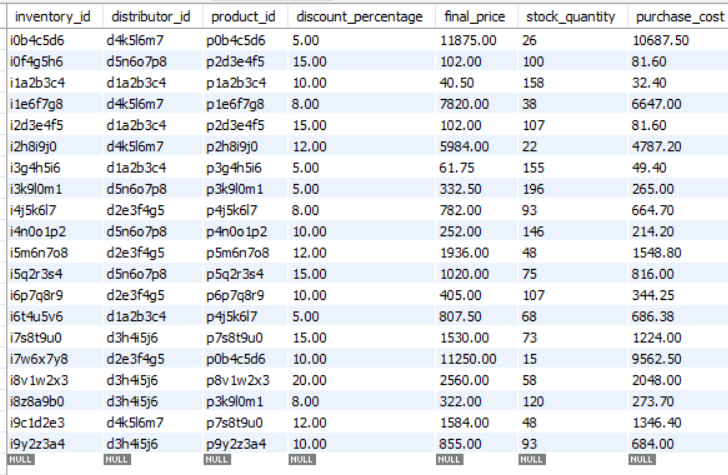
* 1. **payments :**



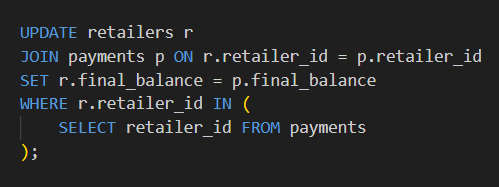
* 1. **Deliveries :**

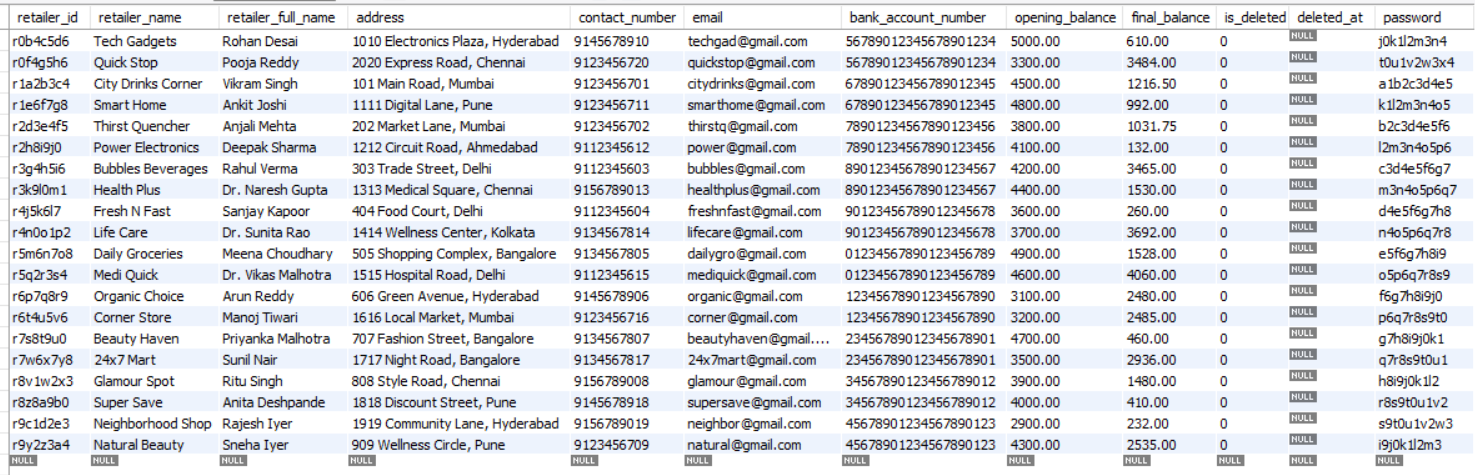
* **UPDATE OPERATIONS :**
  1. **Update the stock quantity after purchase :**





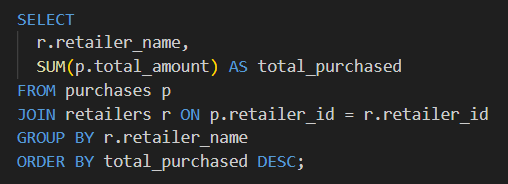
* 1. **Update Final Balance of Retailers after Payments:**

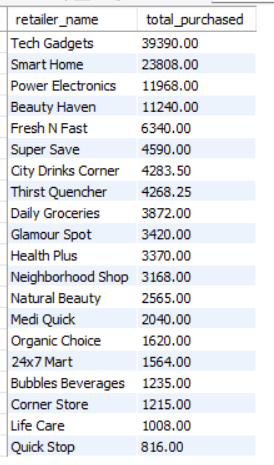




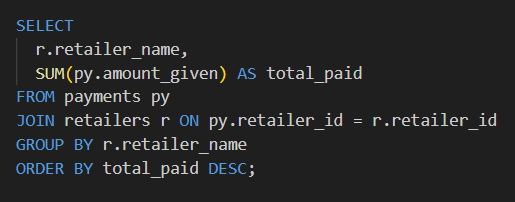
* **SQL QUERIES:**

**1) Total Purchase Amount per Retailer :**



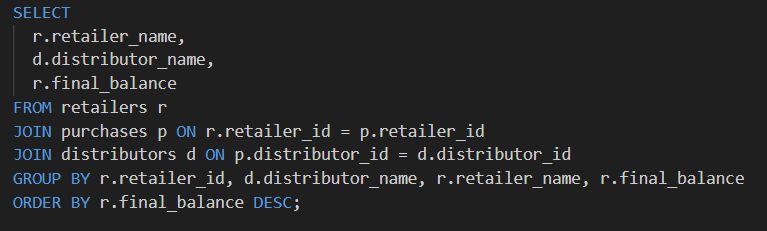


**2) Total Payments Made by Each Retailer :**



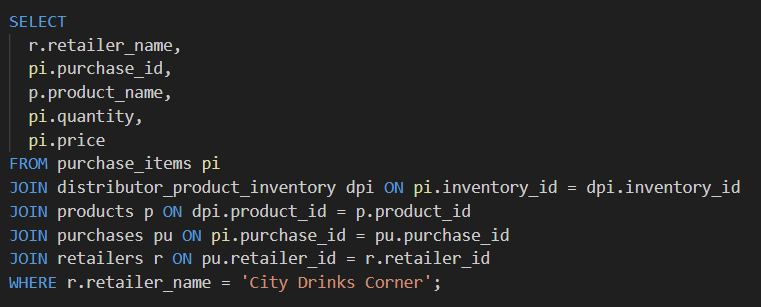


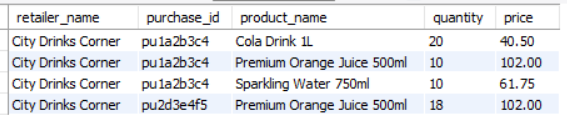
**3) Outstanding Balance for Each Retailer along with corresponding distributor :**



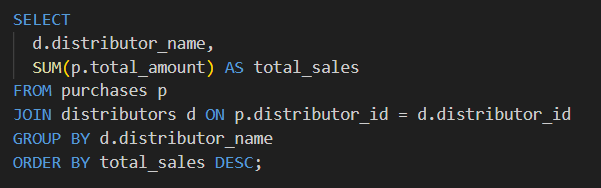


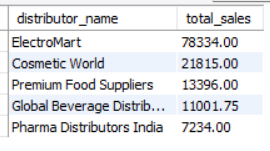
**4) Purchase Items Breakdown for a Retailer :**





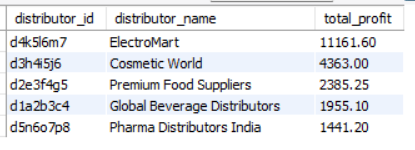
**5) Total Sales (Revenue) per Distributor :**



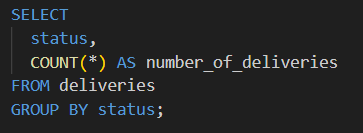


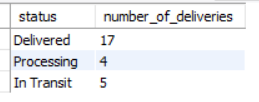
**6) Distributor-wise Profit:**



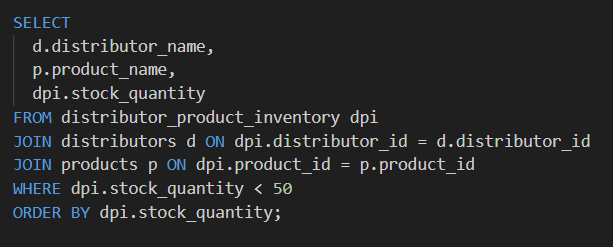


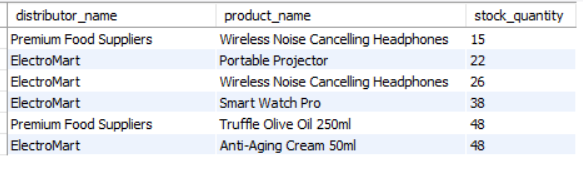
**7) Delivery Summary by Status :**



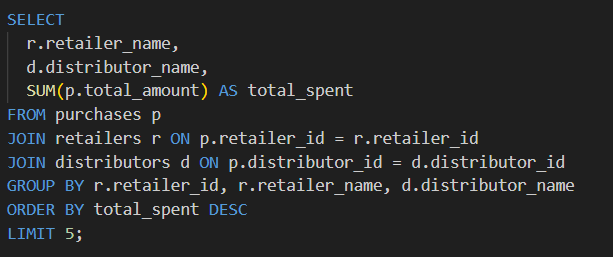


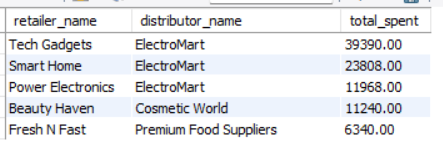
**8) Low Stock Alert :**

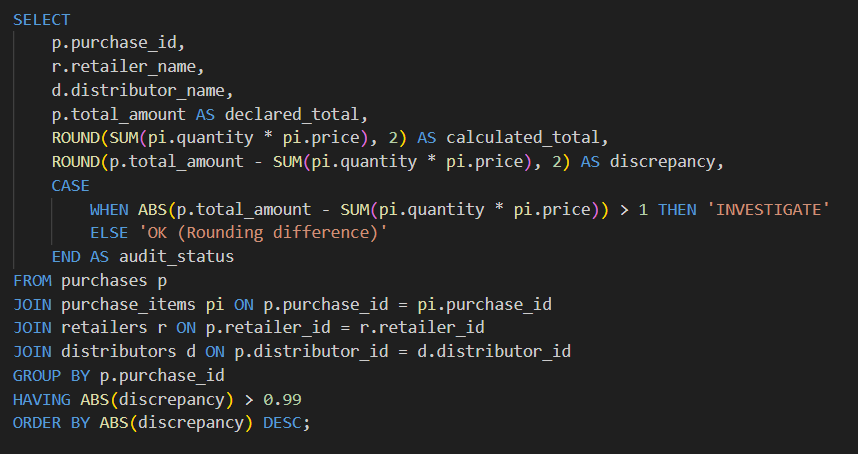




**9) Top 5 Retailers by Purchase Volume (with corresponding distributor) :**





**10) Purchase Audit Report for Discrepancy Detection :**

