

Database Documentation

Contents

1	General Information	1
2	Database Structure	2
2.1	Region	2
2.2	Country	2
2.3	City	2
2.4	Address	2
2.5	Client	3
2.6	Supplier	3
2.7	Warehouse	3
2.8	Employee	4
2.9	Category	4
2.10	Product	4
2.11	ProductInventory	5
2.12	Transaction	5
2.13	TransactionProduct	6
3	Triggers	6
4	Procedures	6

1 General Information

The database was implemented in MySQL and operates in the Aiven cloud. It represents the structure of a warehouse database and contains mechanisms ensuring data integrity, along with procedures and triggers supporting warehouse inventory operations and transactions.

2 Database Structure

2.1 Region

Stores information about geographical regions.

- **RegionID (INT)** - Primary key
- **Name (VARCHAR(50))** - Region name

2.2 Country

Contains data about countries and assigned regions.

- **CountryID (INT)** - Primary key
- **Name (VARCHAR(50))** - Country name
- **CountryCode (VARCHAR(5))** - Country code
- **RegionID (INT)** - Foreign key to Region

2.3 City

Describes cities and their relationships with countries.

- **CityID (INT)** - Primary key
- **Name (VARCHAR(50))** - City name
- **PostalCode (VARCHAR(20))** - Postal code
- **CountryID (INT)** - Foreign key to Country

2.4 Address

Represents addresses of clients, suppliers, warehouses, etc.

- **AddressID (INT)** - Primary key
- **Street (VARCHAR(50))** - Street name
- **StreetNumber (INT)** - Building number
- **CityID (INT)** - Foreign key to City

2.5 Client

End customer data.

- **ClientID (INT)** - Primary key
- **Name (VARCHAR(50))** - Full name or company name
- **Email (VARCHAR(50))** - Email address
- **PhoneNumber (VARCHAR(20))** - Phone number
- **AddressID (INT)** - Foreign key to Address

2.6 Supplier

Product suppliers.

- **SupplierID (INT)** - Primary key
- **Name (VARCHAR(50))** - Supplier name
- **Email (VARCHAR(50))** - Email address
- **PhoneNumber (VARCHAR(20))** - Phone number
- **AddressID (INT)** - Foreign key to Address

2.7 Warehouse

Warehouse descriptions.

- **WarehouseID (INT)** - Primary key
- **Name (VARCHAR(100))** - Warehouse name
- **Capacity (DECIMAL(6,2))** - Total capacity
- **OccupiedCapacity (DECIMAL(6,2))** - Current usage
- **AddressID (INT)** - Foreign key to Address

2.8 Employee

Employees assigned to warehouses.

- **EmployeeID (INT)** - Primary key
- **Name (VARCHAR(100))** - First name
- **Surname (VARCHAR(100))** - Last name
- **Position (VARCHAR(100))** - Job position
- **Email (VARCHAR(100))** - Email
- **PhoneNumber (VARCHAR(20))** - Phone
- **AddressID (INT)** - Foreign key to Address
- **WarehouseID (INT)** - Foreign key to Warehouse

2.9 Category

Product categories.

- **CategoryID (INT)** - Primary key
- **Name (VARCHAR(100))** - Category name
- **Description (VARCHAR(200))** - Description

2.10 Product

Warehoused or sold products.

- **ProductID (INT)** - Primary key
- **Name (VARCHAR(50))** - Product name
- **Description (VARCHAR(200))** - Description
- **UnitPrice (DECIMAL(6,2))** - Unit price
- **UnitSize (DECIMAL(4,2))** - Unit size
- **CategoryID (INT)** - Foreign key to Category

2.11 ProductInventory

Warehouse contents.

- **ProductInventoryID (INT)** - Primary key
- **ProductID (INT)** - Foreign key to Product
- **WarehouseID (INT)** - Foreign key to Warehouse
- **Quantity (INT)** - Product quantity
- **Price (DECIMAL(10,2))** - Product price

2.12 Transaction

Representation of goods transactions.

- **TransactionID (INT)** - Primary key
- **TransactionType (ENUM)** - Transaction type
- **Date (DATE)** - Transaction date
- **Description (VARCHAR(200))** - Description
- **EmployeeID (INT)** - Foreign key to Employee
- **FromWarehouseID (INT)** - Source warehouse
- **ToWarehouseID (INT)** - Target warehouse
- **ClientID (INT)** - Target client
- **SupplierID (INT)** - Supplier
- **SourceWarehouseCapacityAfterTransaction (DECIMAL(6,2))** - Source capacity after transaction
- **TargetWarehouseCapacityAfterTransaction (DECIMAL(6,2))** - Target capacity after transaction

Three transaction types are supported:

- **WAREHOUSE_TO_WAREHOUSE** - Goods transfer between warehouses
- **SUPPLIER_TO_WAREHOUSE** - Goods delivery from supplier to warehouse
- **WAREHOUSE_TO_CUSTOMER** - Goods issue from warehouse to customer

2.13 TransactionProduct

Details about products transferred in transactions.

- **TransactionProductID (INT)** - Primary key
- **TransactionID (INT)** - Foreign key to Transaction
- **ProductID (INT)** - Foreign key to Product
- **Quantity (INT)** - Product quantity
- **TransactionPrice (DECIMAL(10,2))** - Unit price of transferred product

3 Triggers

Triggers implement automatic actions related to data integrity and business logic:

- **Warehouse capacity verification before state update:** Before each product quantity change, verification checks if planned update would exceed total warehouse capacity. If yes - operation is aborted with error.
- **Warehouse occupancy update:** After each add/change/delete operation in ProductInventory table, corresponding OccupiedCapacity value is updated according to product unit size and quantity change.
- **Transaction type validation:** Before insert/update in Transaction table, logical validation checks transaction type against required fields (e.g., SUPPLIER_TO_WAREHOUSE requires SupplierID and FromWarehouseID). Otherwise transaction is rejected.
- **Automatic product inventory initialization:** After adding new product to Product table, entries are automatically generated in ProductInventory table for each existing warehouse with initial quantity zero and price equal to new product's unit price.

4 Procedures

Stored procedures implement complex business logic operating on multiple tables. They form operational layer enabling safe warehouse transactions:

- **add_transaction** - Central procedure handling all transaction types. Based on JSON with products:

- Inserts transaction record
 - Validates and updates product quantities in relevant warehouses
 - Saves products in TransactionProduct table
 - Updates warehouse occupancy after transaction
- **receive_delivery** - Handles delivery from supplier to warehouse. Before delegating to **add_transaction**, verifies target warehouse has sufficient capacity.
- **sell_to_client** - Handles goods sale to client. Checks product availability in source warehouse, then delegates to **add_transaction**.
- **exchange_between_warehouses** - Handles product transfer between warehouses. Verifies product availability in source warehouse and capacity in target warehouse before delegating to **add_transaction**.