NOVA IMS Information Management School

Storing & Retrieving Data

Project Assignment.

Presented By

Henry Tirla

ID#: 20221016

Mahmoud Ahmed

ID#: 202213976

Course Instructor: Mijail Naranjo-Zolotov

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Storing & Retrieving Data Project Report

Henry Tirla - Mahmoud Ahmed

December 2022

Introduction

In this report, we shall discuss the business model of a retail grocery store, how we implemented a database design to capture relevant processes, and how we developed the required SQL queries for this assignment.

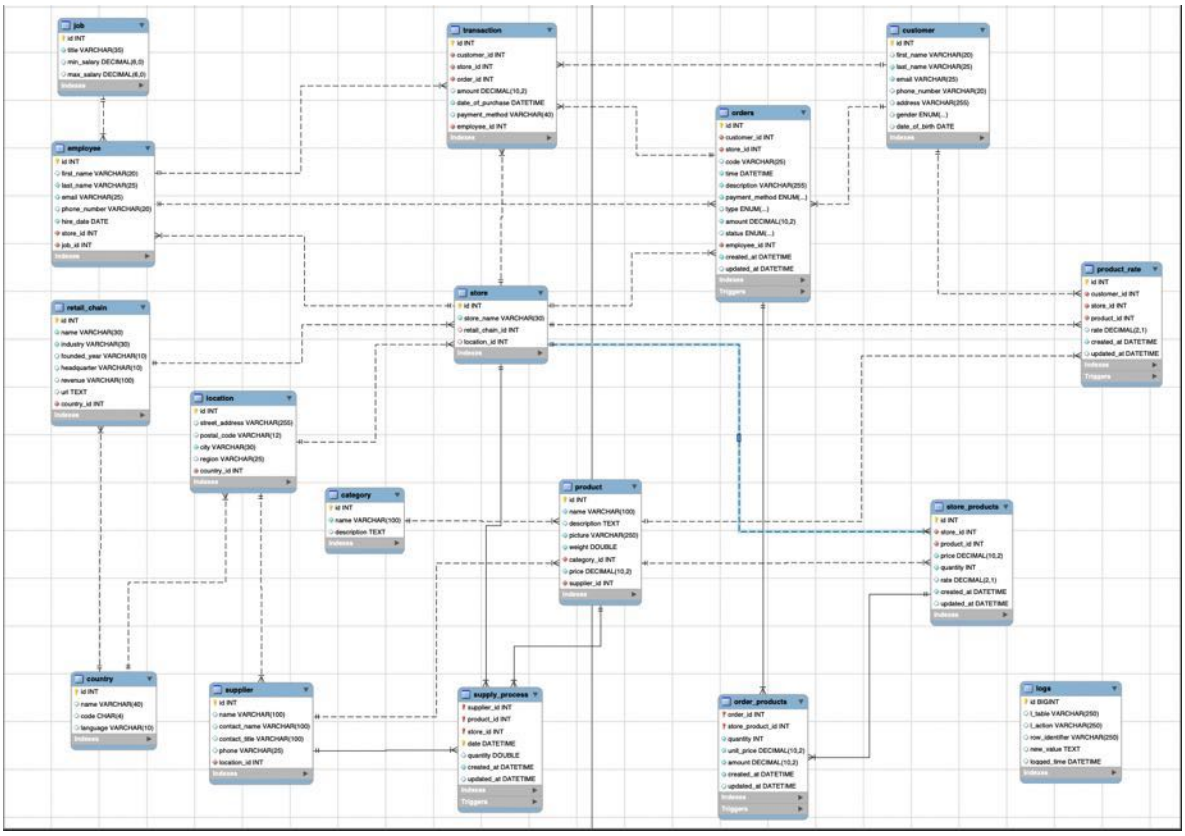
Description

We chose a commercial grocery business process that would benefit from a relational database regarding inventory management.
In a grocery store, it is essential to track and manage the inventory of various products and items to ensure that there is always a sufficient supply of products for customers. We designed a relational database that could be used to store information about each product, including its name, price, quantity in stock, and location in the store. This information is organized into different tables, each representing a different aspect of the inventory, such as the product details, the quantity in stock, and the location in the store.
Also, employees play a crucial role in the business process. They may be involved in various tasks, such as purchasing goods from suppliers, managing inventory, organizing and displaying store products, and facilitating customer transactions.
We also keep a record of the retail chain network of stores selling different products. The chain is responsible for establishing relationships with suppliers, managing inventory and finances, and providing a pleasant shopping experience for customers.

Suppliers are the companies or individuals that provide the retail chain's products. They may include farmers, food manufacturers, and distributors. The retail chain will negotiate prices and delivery schedules with the suppliers to ensure a steady supply of goods.

Transactions refer to the process of a customer purchasing goods from the store. This may involve the customer selecting items, paying at the checkout lane, and receiving a receipt. The store will manage the financial aspect of the transaction, including processing the payment and recording the sale. Transactions are an essential part of the retail chain's business, as they generate revenue for the company.

Entity Relational Diagram
NB: Please check last page for clear diagram



B.

This task was completed.

C.

Though only two triggers were requested, we created 6 for this report only two will be discussed.

The TR_supply_process_AINSERT trigger is executed after a new row is inserted into the supply_process table. It updates the quantity of a product in the store_products table if the product already exists or inserts a new row into the store_products table if the product does not exist. After the operation, it also inserts a log entry into the logs table.

```
CREATE TRIGGER `TR_order_products_BINSERT` BEFORE INSERT ON `order_products` FOR EACH ROW BEGIN
    DECLARE storeID INT DEFAULT 0;
    DECLARE pQuantity INT DEFAULT 0;
    DECLARE pPrice INT DEFAULT 0;
    SELECT `orders`.store_id INTO storeID FROM `orders` WHERE `orders`.id = NEW.order_id;
    IF storeID > 0 THEN
        SELECT store_products.quantity, store_products.price INTO pQuantity, pPrice
                    FROM store_products WHERE store_products.id= NEW.store_product_id;
        IF pQuantity > NEW.quantity THEN
            SET NEW.unit_price = pPrice;
            SET NEW.amount = pPrice * NEW.quantity;
        ELSE
            SIGNAL SQLSTATE '12345'
            SET MESSAGE_TEXT = 'Store Dose not have This Product / Product Out Of Stock / Requested Quantity Exceeded';
        END IF;
    ELSE
        SIGNAL SQLSTATE '40001'
        SET MESSAGE_TEXT = 'Invalid Store ID / Check Order ID If Exists';
    END IF;
END
```

The TR_order_products_BINSERT trigger is executed before a new row is inserted into the order_products table. It retrieves the store ID associated with the order ID and then retrieves the quantity and price of the product from the store_products table. If the store has enough product in stock to fulfill the order, the trigger calculates the unit price and total amount of the order and sets these values in the order_products table. If the store does not have enough of the product in stock or if the order is invalid for some other reason, the trigger raises an error and does not insert the row into the order_products table.

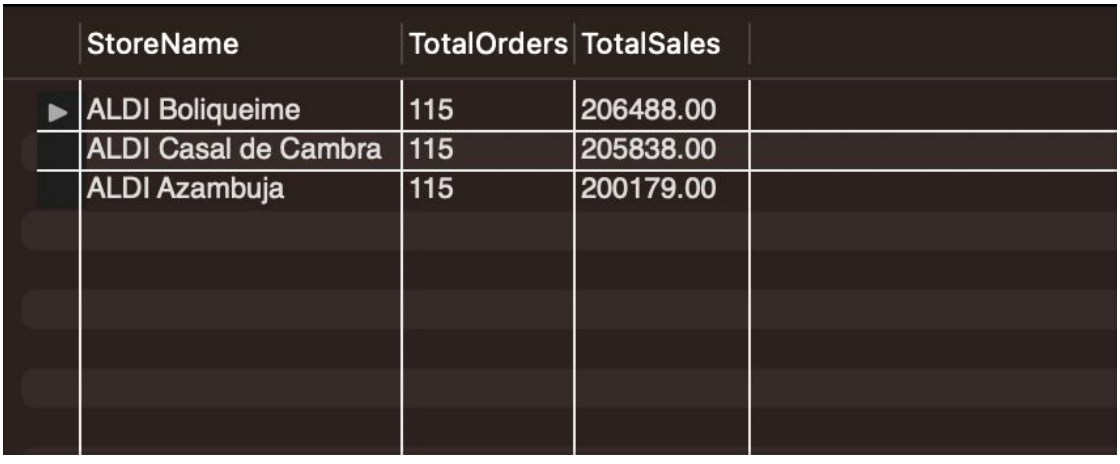
F.

```
SELECT
    CONCAT(customer.first_name, ' ', customer.last_name)
AS FullName,
    orders.time AS OrderTime,
    store.store_name AS StoreName,
    product.name AS ProductName,
    order_product.quantity AS ProductQuantity,
    order_product.unit_price AS ProductPrice
FROM `orders`
    JOIN customer ON customer.id = orders.id
    JOIN store ON store.id = orders.store_id
    JOIN order_products order_product ON
order_product.order_id = orders.id
    JOIN store_products store_product ON
store_product.id = order_product.store_product_id
    JOIN product ON product.id =
```

```
store_product.product_id
WHERE orders.time BETWEEN '2021-01-01' AND NOW() ORDER
BY customer_id;
```

FullName	OrderTime	StoreName	ProductName	ProductQuant...	ProductPri...
Maria Anders	2022-06-05 18:04:38	ALDI Agualva Cacém	Dettol Original Soap	7	26.00
Maria Anders	2022-06-05 18:04:38	ALDI Agualva Cacém	Rasmalai	9	240.00
Christina Berglund	2021-10-28 16:34:53	ALDI Almancil	Dant Kanti	8	60.00
Hanna Moos	2022-07-25 06:25:15	ALDI Alta de Lisboa	Alovera shampoo	10	95.00
Hanna Moos	2022-07-25 06:25:15	ALDI Alta de Lisboa	coconut biscuit	4	10.00
Hanna Moos	2022-07-25 06:25:15	ALDI Alta de Lisboa	Cream Milk bikis	5	25.00
Hanna Moos	2022-07-25 06:25:15	ALDI Alta de Lisboa	Tata lite	2	18.00
Frédérique Citeaux	2021-08-22 17:01:53	ALDI Alverca do Ribatejo	Alovera kranti soap	10	30.00
Frédérique Citeaux	2021-08-22 17:01:53	ALDI Alverca do Ribatejo	Good day chocolate	4	30.00
Frédérique Citeaux	2021-08-22 17:01:53	ALDI Alverca do Ribatejo	swarn night-queen agarbatti	9	65.00
Laurence Lebihan	2021-05-28 11:08:22	ALDI Boliqueime	Marrie gold	1	10.00
Elizabeth Lincoln	2021-09-16 18:28:30	ALDI Cacém	Milk bikis	6	10.00
Elizabeth Lincoln	2021-09-16 18:28:30	ALDI Cacém	coriander	8	80.00
Elizabeth Lincoln	2021-09-16 18:28:30	ALDI Cacém	Chandan soap	6	20.00
Elizabeth Lincoln	2021-09-16 18:28:30	ALDI Cacém	maggi masala noodle	3	25.00
Victoria Ashworth	2021-07-16 14:06:16	ALDI Caldas da Rainha	Cream Milk bikis	3	25.00
Patricio Simpson	2021-05-15 14:09:19	ALDI Carregado	coconut biscuit	7	10.00
Patricio Simpson	2021-05-15 14:09:19	ALDI Carregado	Laung	9	16.00
Patricio Simpson	2021-05-15 14:09:19	ALDI Carregado	Daily Shine	2	110.00
Patricio Simpson	2021-05-15 14:09:19	ALDI Carregado	cadbury oreo	2	15.00

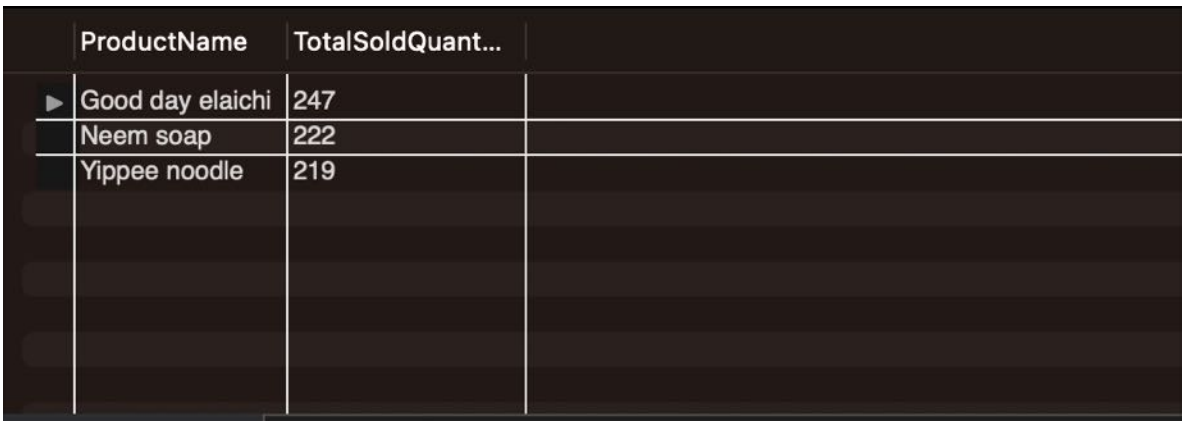
```
SELECT store.store_name AS StoreName, count(*) AS
TotalOrders, SUM(amount) AS TotalSales
FROM orders
    JOIN store ON orders.store_id = store.id
GROUP BY store_id ORDER BY TotalSales DESC , TotalOrders
DESC LIMIT 3;
```



StoreName	TotalOrders	TotalSales
ALDI Boliqueime	115	206488.00
ALDI Casal de Cambra	115	205838.00
ALDI Azambuja	115	200179.00

```
SELECT product.name AS ProductName,
SUM(order_product.quantity) TotalSoldQuantity
FROM order_products order_product
    JOIN store_products store_product ON
order_product.store_product_id = store_product.id
    JOIN product ON store_product.product_id =
```

```
product.id
GROUP BY product.id
ORDER BY TotalSoldQuantity desc LIMIT 3;
```



ProductName	TotalSoldQuant...
Good day elaichi	247
Neem soap	222
Yippee noodle	219

```
CALL GET_AvgSales(5, 2021, 2022);
```

PeriodOfSales	TotalSales	avgYearlySales	avgMonthlySales
2021-2022	1701399.00	40509	3858

```
SELECT country.name AS CountryName, location.city AS
cityName, store.store_name AS StoreName, SUM(amount) AS
TotalSales
FROM orders
    JOIN store ON orders.store_id = store.id
    JOIN location ON store.location_id = location.id
    JOIN country ON country.id = location.country_id
GROUP BY country.id, location.city, store.id;
```

CountryName	cityName	StoreName	TotalSales
Portugal	Agualva-Cacém	ALDI Agualva Cacém	183715.00
Portugal	Albufeira	ALDI Albufeira	195984.00
Portugal	Alfragide	ALDI Alcantarilha	190906.00
Portugal	Alfragide	ALDI Alfragide	164001.00
Portugal	Almancil	ALDI Almancil	160248.00
Portugal	Lisboa	ALDI Alta de Lisboa	171194.00
Portugal	Alverca do Ribatejo	ALDI Alverca do Ribatejo	169090.00
Portugal	Azambuja	ALDI Azambuja	200179.00
Portugal	Boliqueime	ALDI Boliqueime	206488.00

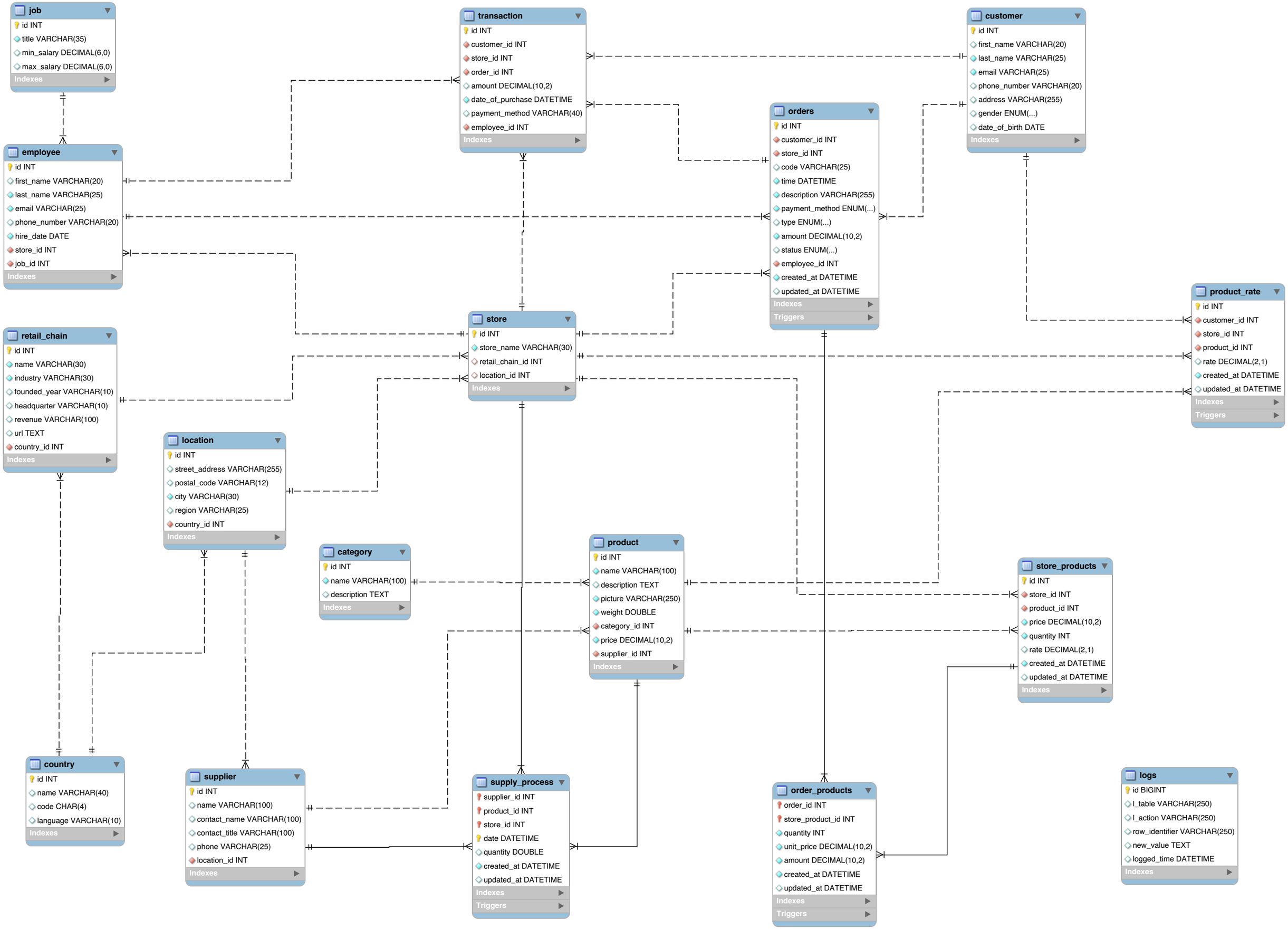
```
SELECT location.city AS City, store.store_name AS
StoreName, product.name AS productName,
store_product.rate AS avgRate
FROM orders
    JOIN order_products order_product ON orders.id =
order_product.order_id
    JOIN store_products store_product ON
store_product.id = order_product.store_product_id
    JOIN product ON product.id =
store_product.product_id
    JOIN store ON store.id = orders.store_id
    JOIN location ON store.location_id = location.id
GROUP BY location.city, product.id, store.id, avgRate
ORDER BY avgRate desc ;
```

City	StoreName	productName	avgRate
Agualva-Cacém	ALDI Agualva Cacém	Dettol Original Soap	0.0
Agualva-Cacém	ALDI Cacém	Dettol Original Soap	0.0
Lisboa	ALDI Alta de Lisboa	Dettol Original Soap	0.0
Azambuja	ALDI Azambuja	Dettol Original Soap	0.0
Alfragide	ALDI Alcantarilha	Dettol Original Soap	0.0
Carregado	ALDI Carregado	Dettol Original Soap	0.0
Almancil	ALDI Almancil	Dettol Original Soap	0.0
Caldas da Rainha	ALDI Caldas da Rainha	Dettol Original Soap	0.0
Alfragide	ALDI Alfragide	Dettol Original Soap	0.0

```
SELECT * FROM vw_invoice_details WHERE invoiceId = 452;
```

invoiceId	ProductName	unitCost	QTY	AMOUNT
452	Dant Kanti	60.00	2	120.00
452	Turmeric	40.00	1	40.00
452	Gold sohanpapdi	330.00	3	990.00
452	Rasgulla	250.00	7	1750.00
452	Classic rock salt	30.00	8	240.00
452	yippee magic masala	25.00	5	125.00
452	Close-up everfresh	40.00	7	280.00
452	Beauty Bar	60.00	8	480.00
452	Sohan papdi	140.00	9	1260.00

```
SELECT * FROM vw_invoice_summary WHERE invoiceId = 452;
```


job
id INT
title VARCHAR(35)
min_salary DECIMAL(6,0)
max_salary DECIMAL(6,0)
Indexes
transaction
id INT
customer_id INT
store_id INT
order_id INT
amount DECIMAL(10,2)
date_of_purchase DATETIME
payment_method VARCHAR(40)
employee_id INT
Indexes
customer
id INT
first_name VARCHAR(20)
last_name VARCHAR(25)
email VARCHAR(25)
phone_number VARCHAR(20)
address VARCHAR(255)
gender ENUM(...)
date_of_birth DATE
Indexes
employee
id INT
first_name VARCHAR(20)
last_name VARCHAR(25)
email VARCHAR(25)
phone_number VARCHAR(20)
hire_date DATE
store_id INT
job_id INT
Indexes
orders
id INT
customer_id INT
store_id INT
code VARCHAR(25)
time DATETIME
description VARCHAR(255)
payment_method ENUM(...)
type ENUM(...)
amount DECIMAL(10,2)
status ENUM(...)
employee_id INT
created_at DATETIME
updated_at DATETIME
Indexes
Triggers
product_rate
id INT
customer_id INT
store_id INT
product_id INT
rate DECIMAL(2,1)
created_at DATETIME
updated_at DATETIME
Indexes
Triggers
store
id INT
store_name VARCHAR(30)
retail_chain_id INT
location_id INT
Indexes
retail_chain
id INT
name VARCHAR(30)
industry VARCHAR(30)
founded_year VARCHAR(10)
headquarter VARCHAR(10)
revenue VARCHAR(100)
url TEXT
country_id INT
Indexes
location
id INT
street_address VARCHAR(255)
postal_code VARCHAR(12)
city VARCHAR(30)
region VARCHAR(25)
country_id INT
Indexes
category
id INT
name VARCHAR(100)
description TEXT
Indexes
product
id INT
name VARCHAR(100)
description TEXT
picture VARCHAR(250)
weight DOUBLE
category_id INT
price DECIMAL(10,2)
supplier_id INT
Indexes
store_products
id INT
store_id INT
product_id INT
price DECIMAL(10,2)
quantity INT
rate DECIMAL(2,1)
created_at DATETIME
updated_at DATETIME
Indexes
country
id INT
name VARCHAR(40)
code CHAR(4)
language VARCHAR(10)
Indexes
supplier
id INT
name VARCHAR(100)
contact_name VARCHAR(100)
contact_title VARCHAR(100)
phone VARCHAR(25)
location_id INT
Indexes
supply_process
supplier_id INT
product_id INT
store_id INT
date DATETIME
quantity DOUBLE
created_at DATETIME
updated_at DATETIME
Indexes
Triggers
order_products
order_id INT
store_product_id INT
quantity INT
unit_price DECIMAL(10,2)
amount DECIMAL(10,2)
created_at DATETIME
updated_at DATETIME
Indexes
Triggers
logs
id BIGINT
l_table VARCHAR(250)
l_action VARCHAR(250)
row_identifier VARCHAR(250)
new_value TEXT
logged_time DATETIME
Indexes