Bytewise Fellowship

Task 4: SQL Exercise - Sales Database Schema Creation and Data Queries

Name	Muhammad Ahsan Saleem
Date	30th June, 2024
Submitted to:	Muhammad Bilal

Introduction:

The objective of this task was to create a new database, schema, and tables based on provided CSV files. Furthermore, this document presents a series of SQL queries designed to explore and analyze data from the Sales database. The goal is to address specific questions related to customer, product, supplier, and order information. By executing these queries, we aim to retrieve valuable insights and provide accurate numerical results, which are essential for understanding the data landscape and supporting data-driven decision-making processes. The results and methodologies are documented below.

Task 4

- Create a new database (Resource link: YouTube Video).
- Create a schema in it.
- Create tables for the provided CSVs by understanding the data in the CSVs. Use the correct data type for each column.

Questions:

1. Write a query to fetch all customer names and sort them alphabetically. (Topic: Sorting)
2. Write a query to fetch all product names and their prices, sorted by price from low to high. (Topic: Sorting)
3. Write a query to fetch supplier names that start with the letter 'A' and sort them by their names. (Topic: Sorting with Operators and Wildcards)
4. Write a query to fetch all items and sort them by their status, placing NULL values first.
5. Write a query to fetch all products, sort them first by category and then by price in descending order.
6. Write a query to fetch all customer names and phone numbers, but sort them by the last four digits of their phone numbers in ascending order. (Hint: Use sorting with substrings)

CSVs attached:

1. cities.csv
2. categories.csv
3. customers.csv
4. items.csv
5. orders.csv
6. products.csv
7. suppliers.csv

Database and Schema Creation on PostgreSQL 16:

1. **Create a new database (Resource link: YouTube Video)**
    a. Right click on the 'Databases' heading under 'Servers' → 'PostgreSQL 16' on the side bar on the left.
    b. Click on 'Create'.
    c. Click on 'Database'.
    d. Give the database an appropriate name.
    e. Click save.

2. **Create a new schema**
    a. On the same sidebar, right click on the 'Schemas' heading under the new heading of the new database created.
    b. Click on 'Create'.
    c. Click on 'Schema'.
    d. Give the schema and appropriate name.
    e. Create a schema named **'sales'** in the new database.
    f. Click save.

Creating Tables and Importing Data:

1. **Creating tables for the CSVs provided.**
    a. The code used for creating each table is as follows. Each table is designed to match the structure and data types of the corresponding CSV file.

SQL Statements:

```
-- Creating Tables and Columns

CREATE TABLE sales.categories (
    category_id VARCHAR(100) PRIMARY KEY,
    name VARCHAR(100),
    status VARCHAR(20),
    description VARCHAR(500)
);

CREATE TABLE sales.cities (
    city_id VARCHAR(100) PRIMARY KEY,
    name VARCHAR(100),
    province VARCHAR(20),
    country VARCHAR(100),
    status VARCHAR(20)
);

CREATE TABLE sales.customers (
    customer_id VARCHAR(100) PRIMARY KEY,
    name VARCHAR(100),
    phone VARCHAR(50),
    location VARCHAR(50),
    status VARCHAR(20)
);

CREATE TABLE sales.items (
    item_id VARCHAR(100) PRIMARY KEY,
    order_id VARCHAR(100),
    product_id VARCHAR(100),
```

```
    amount FLOAT,
    status VARCHAR(20),
    item_timestamp TIMESTAMP
);

CREATE TABLE sales.orders (
    order_id VARCHAR(100) PRIMARY KEY,
    customer_id VARCHAR(100),
    status VARCHAR(50),
    order_time TIMESTAMP,
    total_amount FLOAT
);

CREATE TABLE sales.products (
    product_id VARCHAR(100) PRIMARY KEY,
    name VARCHAR(50),
    supplier_id VARCHAR(100),
    category VARCHAR(50),
    price FLOAT,
    stock_available INT,
    status VARCHAR(50),
    product_createtimestamp TIMESTAMP
);

CREATE TABLE sales.suppliers (
    supplier_id VARCHAR(100) PRIMARY KEY,
    name VARCHAR(100),
    phone VARCHAR(100),
    location VARCHAR(50),
    status VARCHAR(50),
    category VARCHAR(50)
);
```

2. **Import Data into the Created Tables**
    a. The following SQL statements will import data from the CSV files into the corresponding tables in the sales schema.

SQL Statements:

```
-- Import data into sales.categories
COPY sales.categories(category_id, name, status, description)
FROM 'C:\Users\mahsa\Desktop\Bytewise Fellowship\Daily Tasks\Month 1\Task 4\Resources\categories.csv'
DELIMITER ','
CSV HEADER;

-- Import data into sales.cities
COPY sales.cities(city_id, name, province, country, status)
FROM 'C:\Users\mahsa\Desktop\Bytewise Fellowship\Daily Tasks\Month 1\Task 4\Resources\cities.csv'
DELIMITER ','
CSV HEADER;

-- Import data into sales.customers
COPY sales.customers(customer_id, name, phone, location, status)
FROM 'C:\Users\mahsa\Desktop\Bytewise Fellowship\Daily Tasks\Month 1\Task 4\Resources\customers.csv'
DELIMITER ','
CSV HEADER;

-- Import data into sales.items
COPY sales.items(item_id, order_id, product_id, amount, status, item_timestamp)
FROM 'C:\Users\mahsa\Desktop\Bytewise Fellowship\Daily Tasks\Month 1\Task 4\Resources\items.csv'
DELIMITER ','
CSV HEADER;

-- Import data into sales.orders
COPY sales.orders(order_id, customer_id, status, order_time, total_amount)
FROM 'C:\Users\mahsa\Desktop\Bytewise Fellowship\Daily Tasks\Month 1\Task 4\Resources\orders.csv'
```

```
DELIMITER ','
CSV HEADER;

-- Import data into sales.products
COPY sales.products(product_id, name, supplier_id, category, price, stock_available,
status, product_createtimestamp)
FROM 'C:\Users\mahsa\Desktop\Bytewise Fellowship\Daily Tasks\Month 1\Task
4\Resources\products.csv'
DELIMITER ','
CSV HEADER;

-- Import data into sales.suppliers
COPY sales.suppliers(supplier_id, name, phone, location, status, category)
FROM 'C:\Users\mahsa\Desktop\Bytewise Fellowship\Daily Tasks\Month 1\Task
4\Resources\suppliers.csv'
DELIMITER ','
CSV HEADER;
```

Queries and Their Documentation:

Each query addresses a specific question regarding the data.

Question 1

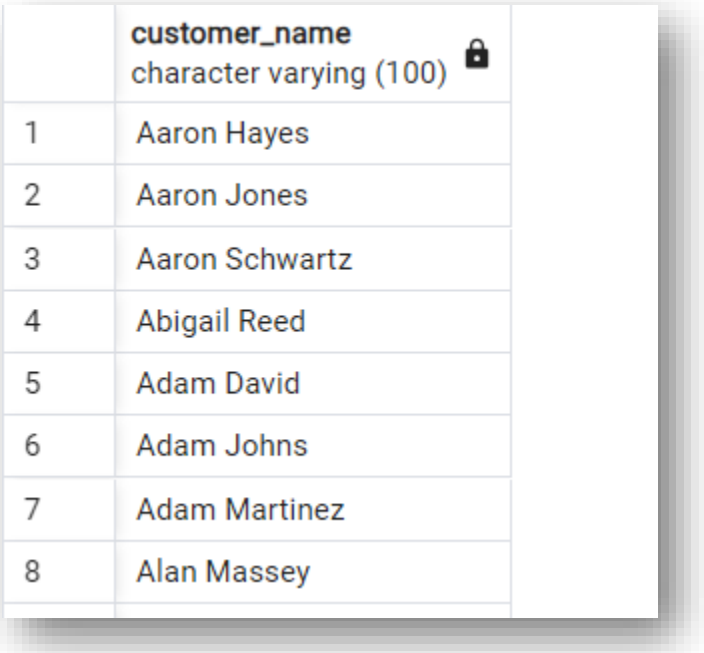
Statement: Write a query to fetch all customer names and sort them alphabetically.

Query:

```
SELECT name AS customer_name
FROM sales.customers
ORDER BY customer_name;
```

```
-- Question 1
SELECT name AS customer_name
FROM sales.customers
ORDER BY customer_name;
```

Screenshot:



	customer_name character varying (100)
1	Aaron Hayes
2	Aaron Jones
3	Aaron Schwartz
4	Abigail Reed
5	Adam David
6	Adam Johns
7	Adam Martinez
8	Alan Massey

Question 2

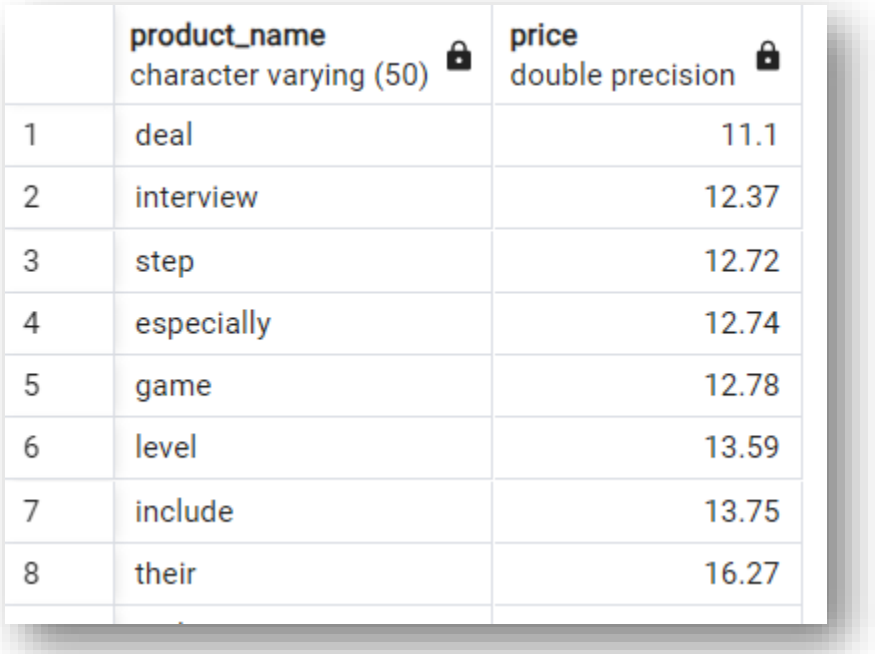
Statement: Write a query to fetch all product names and their prices, sorted by price from low to high.

Query:

SELECT name AS product_name, price
FROM sales.products
ORDER BY price;

```
-- Question 2
SELECT name AS product_name, price
FROM sales.products
ORDER BY price;
```

Screenshot:



	product_name character varying (50)	price double precision
1	deal	11.1
2	interview	12.37
3	step	12.72
4	especially	12.74
5	game	12.78
6	level	13.59
7	include	13.75
8	their	16.27

Question 3

Statement: Write a query to fetch supplier names that start with the letter 'A' and sort them by their names.

Query:

SELECT name AS supplier_name
FROM sales.suppliers
WHERE name LIKE 'A%'
ORDER BY name;

```sql
-- Question 3
SELECT name AS supplier_name
FROM sales.suppliers
WHERE name LIKE 'A%'
ORDER BY name;
```

Screenshot:

	supplier_name character varying (100)
1	Acosta-Freeman
2	Adams-Roberts
3	Adams-Schwartz
4	Adams, Hull and Wise
5	Adams, Rojas and Morgan
6	Adkins and Sons
7	Aguilar-Carter
8	Aguilar Ltd

Question 4

Statement: Write a query to fetch all items and sort them by their status, placing NULL values first.

Query:

SELECT *
FROM sales.items
ORDER BY status NULLS FIRST;

```
-- Question 4
SELECT *
FROM sales.items
ORDER BY status NULLS FIRST;
```

Screenshot:

	item_id [PK] character varying (100)	order_id character varying (100)	product_id character varying (100)	amount double precision	status character varying (20)	item_timestamp timestamp without time zone
1	55d21c57-d8a7-4c8e-96f1-af1c3db13160	6f66a611-407a-4851-b699-4ce8d8d095...	d03892f1-f8f3-44fc-bb86-e431717e4b6f	14.31	Delivered	2024-03-09 06:47:37.275443
2	8a2c9420-43b5-4b51-91aa-1fcae29b7811	7ea54b60-64d2-4583-96df-bf4ef031da...	5a49b5e1-a14d-4a84-b131-4b2de86b63...	32.95	Delivered	2024-04-18 05:24:22.285073
3	72af47e3-9998-420a-86ed-56c01cb9bd81	ddee01dc-20a1-42ff-a286-7124267ad0...	7217ab96-f866-4c74-b9ae-9c3eaf767ab4	86.44	Delivered	2024-03-31 20:51:49.228488
4	4b5a19b1-64be-41d7-ae74-a2543ca734e4	b5e30740-1585-4ea6-9c99-cd6a7877f4...	f8366f6b-ecf1-4e75-935d-ab91137c2933	8.1	Delivered	2024-01-16 03:30:29.394309
5	0bdb0f27-6bf6-4e0c-9b49-9fee26a36b60	411c82eb-100e-48fc-8095-befe7791b6f6	e6d06da3-c252-4117-9a45-6c6511aa71...	56.5	Delivered	2024-04-19 16:04:29.456686
6	e7df11b8-2943-4e28-9168-12e136a1ada0	1170d4f6-9a0f-4de6-b0e5-f671870a49...	9ff2250b-0845-41a1-9ee4-b2bbf5cc50bb	77.22	Delivered	2024-04-11 04:17:19.372846
7	03f57e54-db61-4fbd-b59a-8d5d45baa180	160a6358-7348-4e75-89dc-a37a8caf60...	3bc16381-bd29-4b7f-8366-3ecf707883f8	51.07	Delivered	2024-06-01 14:44:57.709625
8	72f7d19f-5c38-473f-a712-ad354c8a89bc	ce967796-9486-406e-9955-6c53442a8...	dd9a1016-b858-4245-a173-dc99ea0f63...	55.03	Delivered	2024-04-05 13:47:09.495484

Question 5

Statement: Write a query to fetch all products, sort them first by category and then by price in descending order.

Query:

SELECT *
FROM sales.products
ORDER BY category, price DESC;

```
-- Question 5
SELECT *
FROM sales.products
ORDER BY category, PRICE DESC
```

Screenshot:

	product_id [PK] character varying (100)	name character varying (50)	supplier_id character varying (100)	category character varying (50)	price double precision	stock_available integer	status character varying (50)	product_cr timestamp
1	12ea3377-cca5-4965-a41c-9a30998310...	of	3f89d404-d78f-4aca-8c67-ee4e434674d5	a	302.68	979	InActive	2024-01-0
2	494c69e7-db73-42fd-9eef-7880822585a5	offer	01abe709-0567-47dc-ae92-3b077c44b8...	able	385.18	137	Active	2024-05-0
3	b8adc046-eef0-4f58-9c7a-7c6f58a2d628	share	6252ec97-c226-4c94-8b70-7ce1642719...	able	76.71	283	InActive	2024-06-1
4	a63e349b-fb6e-4222-aca0-5079603843bf	wide	16157d6a-f40a-4140-b233-c8e5e8542b...	about	398.27	851	Out of Stock	2024-02-2
5	e61f1dad-1ec7-4114-aaf5-7a449f91c7b4	factor	c9a89c46-4f89-4cc4-a804-be28738fc04b	about	91.72	974	Out of Stock	2024-03-0
6	2d61572a-ad68-4eb9-a327-cbde171ce7...	middle	f292cfe7-c77d-4dd4-8b75-b5eb72f08ece	above	337.62	812	InActive	2024-02-1
7	e37e126c-f0d4-4b67-8efc-5d4226a23a18	door	7b539636-fcdb-4749-9e31-f9e38c83cec8	above	281.2	275	InActive	2024-04-2
8	dad2a61c-41d1-49ea-96b1-6c78daa687...	Congress	c2721d21-27ef-474c-82a6-5dcf1e3b6fcd	above	93.24	928	Active	2024-05-1

Question 6

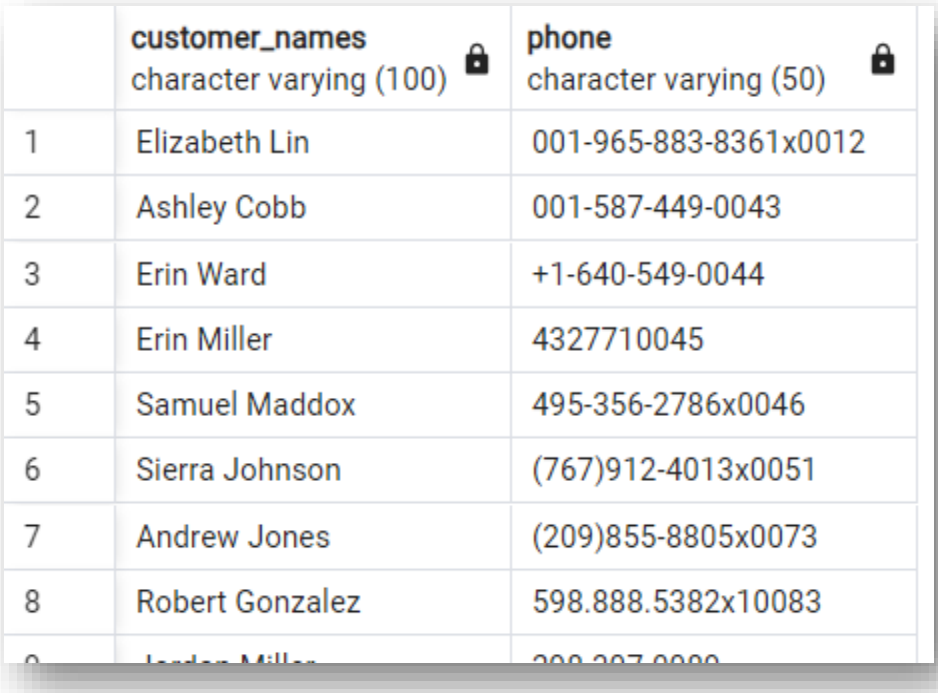
Statement: Write a query to fetch all customer names and phone numbers, but sort them by the last four digits of their phone numbers in ascending order.

Query:

SELECT name AS customer_name, phone
FROM sales.customers
ORDER BY SUBSTRING(phone FROM LENGTH(phone) - 3 FOR 4);

```
-- Question 6
SELECT name AS customer_names, phone
FROM sales.customers
ORDER BY SUBSTRING(phone FROM LENGTH(phone) - 3 FOR 4);
```

Screenshot:



	customer_names character varying (100)	phone character varying (50)
1	Elizabeth Lin	001-965-883-8361x0012
2	Ashley Cobb	001-587-449-0043
3	Erin Ward	+1-640-549-0044
4	Erin Miller	4327710045
5	Samuel Maddox	495-356-2786x0046
6	Sierra Johnson	(767)912-4013x0051
7	Andrew Jones	(209)855-8805x0073
8	Robert Gonzalez	598.888.5382x10083

Conclusion:

This document details the steps to create a new database and schema, define tables based on provided CSV files, import data, and execute SQL queries to analyze the data. Following these steps ensures the successful setup and querying of the sales data.