MILESTONE 1

Development of a Relational Database for Comprehensive Laptop Specifications and Supplier Management

Team Details :

VENKATA SAI BHARGAV DHARA	50559703
THANMAYA SRI SIGIREDDI	50559907
LAKSHMI MOUNIKA MASIMUKKU	50560106

Problem Statement :

The laptop market is highly saturated with a lot of models, each of them with different specifications, pricing, and availability, which makes it very difficult for consumers to manage such a complex knowledge base in all respectives and to perform effective analysis. Traditional tools based on Excel are not sufficient to represent these intricate relationships between laptops, their components, and suppliers for large datasets and when enhanced queries and updates have to be considered. Hence, there is an urgent need for an extensible relational database that will store the details of laptop specifications, maintain supplier information, and support such complex operations. The solution should provide the capability to enhance data integrity, efficiency, and decision-making by facilitating various responsibilities: finding laptops with specific configurations, analyzing the distribution of prices of laptops, dealing with supplier inventory, and providing comprehensive comparisons for decisions to be made by the customers.

Who will use this database?

The data will be used by -
Consumers: The above information can be utilised by the consumers to find and compare the laptops with a particular specification and make an informed decision to buy.

Retailers and Suppliers: will use the information to manage their inventory, update their stock level, and optimise their product based on demand.

Manufacturers: They will utilise it for the development of the product so that they get to know the current trend of the market and the consumer preference.

Market Analysts: The analysts can study various kinds of pricing, demand, and trends in the laptop market to gain insight and forecast from the same.

Technical support teams can log in for detailed specifications to help troubleshoot, maintain, and support customers effectively.

The stakeholders will then utilize the database to gain access efficiently, analyze, and update the laptop information related to their needs.

Who will administer the database? You are encouraged to give a real-life scenario.

Who will administer the database:

It would then be maintained by a full-time Database Administrator or the IT Department of an organization that would actually use the database. Operational and Performance Optimization: Ensuring the database works at an optimum, performing regular backups, ensuring performance is optimized.
Security Management: User access controls, permissions management, sensitive data protection.
Data Integrity: It involves monitoring of updates of data, doing validation checks, and checking consistency across different tables.
Support: Providing users with the assistance required in respect of databases, including where necessary training.

Real-Life Scenario:

TechStore Inc. is a retail company that deals in laptops and other electronic equipment, and their database is maintained by Emma, who is a highly professional Database Administrator in their IT Department.
Responsibilities of Emma:
Database Administration: "Carryout daily database backups and monitor system performance".
User Account Management: Creating user accounts for sales and support people and assigning them with proper access.
Data Updates: Liaising with suppliers concerning updates on stock levels and new laptop models in the database.
Security: Security practices given that do not allow unauthorized access to the data.
She does this through the effective administration of the database, hence ensuring other departments access updated information in an easy and safe way for operational activities and making decisions based on full knowledge across TechStore Inc.

What kind of queries do you want to ask?

We intend to execute a variety of advanced SQL queries to extract meaningful insights from the data. These include:

Specification-Based Searches

- Find laptops with specific hardware configurations, such as a certain processor type (e.g., Intel Core i7), minimum RAM size (e.g., at least 16 GB), or specific GPU models.

Grouping and Aggregation:

- Group laptops by brand or GPU brand and count the number of models available, calculate average prices, or analyze distributions.

Price Analysis with Sub-Queries:

- Compute the average price of laptops within a specific category and display models above or below this average.

Joins Across Multiple Tables:

- Retrieve comprehensive information by joining multiple tables, such as laptops along with their supplier names, stock levels, and shipping times.

Nested Queries and Advanced Analysis:

- Identify laptops with the highest battery life within a specific price range or performance category.

Inventory Checks and Supplier Analysis:

- Check stock availability across different suppliers for particular laptop models and assess shipping times.

Trend Analysis and Performance Metrics:

- Analyze sales data (if available) to identify trends, such as the most popular models or brands over a specific period.

Creating a Sample Database :

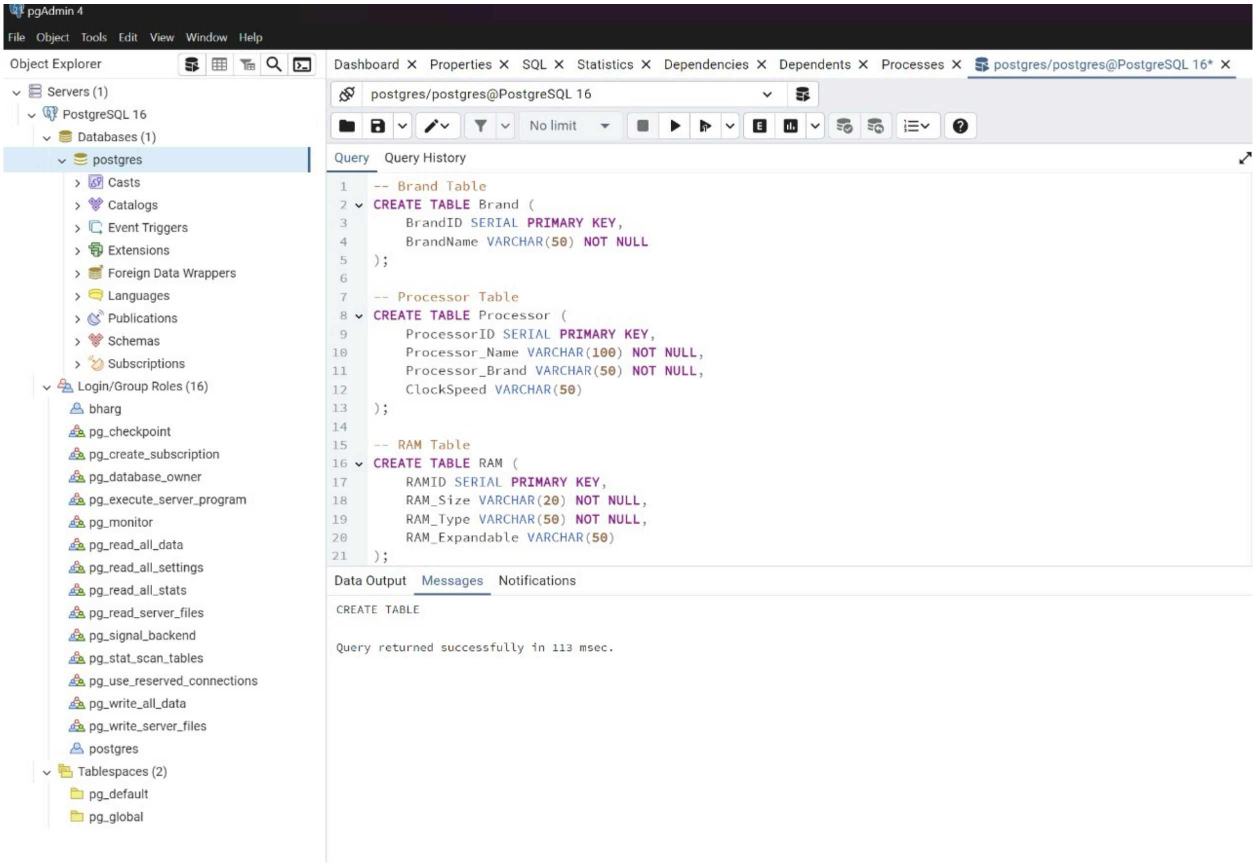
Created a test database from a small subset of the dataset for ease of testing and debugging.

Sample Dataset
This was followed by a random sampling done manually, taking 10 to 15 entries concerning a laptop from the original CSV file. This subset will retain a good representation of different kinds of laptops with specifications changed in such a way that thorough testing can be performed.

SQL Script
A script was created in SQL that was reusable for creating the tables with insertion of the sample data. Since it performs existing table drop, creates new ones, and inserts sample records, one can easily reset the database during development.

Table Creation and Data Insertion

```
-- Drop existing tables if they exist
DROP TABLE IF EXISTS Laptop_Supplier CASCADE;
DROP TABLE IF EXISTS Laptop CASCADE;
DROP TABLE IF EXISTS Supplier CASCADE;
DROP TABLE IF EXISTS Battery CASCADE;
DROP TABLE IF EXISTS Storage CASCADE;
DROP TABLE IF EXISTS Display CASCADE;
DROP TABLE IF EXISTS GPU CASCADE;
DROP TABLE IF EXISTS RAM CASCADE;
DROP TABLE IF EXISTS Processor CASCADE;
DROP TABLE IF EXISTS Brand CASCADE;
```



```
-- Brand Table
CREATE TABLE Brand (
    BrandID SERIAL PRIMARY KEY,
    BrandName VARCHAR(50) NOT NULL
);

-- Processor Table
CREATE TABLE Processor (
    ProcessorID SERIAL PRIMARY KEY,
    Processor_Name VARCHAR(100) NOT NULL,
    Processor_Brand VARCHAR(50) NOT NULL,
    ClockSpeed VARCHAR(50)
);

-- RAM Table
CREATE TABLE RAM (
    RAMID SERIAL PRIMARY KEY,
    RAM_Size VARCHAR(20) NOT NULL,
    RAM_Type VARCHAR(50) NOT NULL,
    RAM_Expandable VARCHAR(50)
);
```

```
CREATE TABLE

Query returned successfully in 113 msec.
```

```
-- Create tables
CREATE TABLE Brand (
   BrandID SERIAL PRIMARY KEY,
   BrandName VARCHAR(100)
);

CREATE TABLE Processor (
   ProcessorID SERIAL PRIMARY KEY,
   Processor_Name VARCHAR(100),
   Processor_Brand VARCHAR(100),
   ClockSpeed VARCHAR(50)
```

```
);

CREATE TABLE RAM (
   RAMID SERIAL PRIMARY KEY,
   RAM_Size VARCHAR(50),
   RAM_Type VARCHAR(50),
   RAM_Expandable VARCHAR(50)
);

CREATE TABLE GPU (
   GPUID SERIAL PRIMARY KEY,
   GPU_Name VARCHAR(100),
   GPU_Brand VARCHAR(100)
);

CREATE TABLE Display (
   DisplayID SERIAL PRIMARY KEY,
   DisplayType VARCHAR(100),
   DisplaySize VARCHAR(50)
);

CREATE TABLE Storage (
   StorageID SERIAL PRIMARY KEY,
   SSD_Size VARCHAR(50),
   HDD_Size VARCHAR(50)
);

CREATE TABLE Battery (
   BatteryID SERIAL PRIMARY KEY,
   Battery_Life VARCHAR(50),
   Adapter VARCHAR(100)
);

CREATE TABLE Supplier (
   SupplierID SERIAL PRIMARY KEY,
   SupplierName VARCHAR(100),
   Contact VARCHAR(100),
   Country VARCHAR(100)
);

CREATE TABLE Laptop (
   LaptopID SERIAL PRIMARY KEY,
   Name VARCHAR(255),
   Price FLOAT,
   BrandID INT REFERENCES Brand(BrandID),
   ProcessorID INT REFERENCES Processor(ProcessorID),
   RAMID INT REFERENCES RAM(RAMID),
   GPUID INT REFERENCES GPU(GPUID),
   DisplayID INT REFERENCES Display(DisplayID),
   StorageID INT REFERENCES Storage(StorageID),
   BatteryID INT REFERENCES Battery(BatteryID)
);

CREATE TABLE Laptop_Supplier (
   LaptopID INT REFERENCES Laptop(LaptopID),
   SupplierID INT REFERENCES Supplier(SupplierID),
   Stock INT,
   Shipping_Time VARCHAR(50),
   PRIMARY KEY (LaptopID, SupplierID)
);

-- Insert sample data into tables
INSERT INTO Brand (BrandName) VALUES
('HP'),
('Lenovo'),
('Dell');

INSERT INTO Processor (Processor_Name, Processor_Brand, ClockSpeed) VALUES
('MediaTek Octa-core', 'MediaTek', '2.0 GHz'),
('AMD Hexa-Core Ryzen 5', 'AMD', '4.0 GHz'),
('Intel Core i5 (12th Gen)', 'Intel', '3.3 GHz');

INSERT INTO RAM (RAM_Size, RAM_Type, RAM_Expandable) VALUES
('4 GB', 'DDR4', 'Not Expandable'),
('8 GB', 'DDR4', '12 GB Expandable'),
('16 GB', 'DDR5', '32 GB Expandable');

INSERT INTO GPU (GPU_Name, GPU_Brand) VALUES
('Integrated Graphics', 'MediaTek'),
('Radeon', 'AMD'),
('GeForce RTX 3050 GPU, 4 GB', 'NVIDIA'),
('Iris Xe', 'Intel');

INSERT INTO Display (DisplayType, DisplaySize) VALUES
('LED', '11.6 inches'),
('LCD', '15.6 inches');

INSERT INTO Storage (SSD_Size, HDD_Size) VALUES
```

```
('64 GB SSD Storage', 'No HDD'),
('512 GB SSD Storage', 'No HDD');

INSERT INTO Battery (Battery_Life, Adapter) VALUES
('Upto 12 Hrs Battery Life', '45W AC Adapter'),
('Upto 11 Hrs Battery Life', '65W AC Adapter'),
('Upto 10 Hrs Battery Life', '56W AC Adapter'),
('Upto 7.30 Hrs Battery Life', 'No Adapter Info');

INSERT INTO Supplier (SupplierName, Contact, Country) VALUES
('Tech Distributors Inc.', 'contact@techdistributors.com', 'USA'),
('Global Tech Supplies', 'info@globaltechsupplies.com', 'China'),
('EuroTech Partners', 'sales@eurotech.com', 'Germany');

INSERT INTO Laptop (Name, Price, BrandID, ProcessorID, RAMID, GPUID, DisplayID,
StorageID, BatteryID) VALUES
('HP Chromebook 11A-NA0002MU', 22990, 1, 1, 1, 1, 1, 1, 1),
('Lenovo Ideapad Slim 3 (82KU017KIN)', 36289, 2, 2, 2, 2, 2, 2, 2),
('Dell G15-5520 (D560822WIN9B)', 78500, 3, 3, 3, 3, 2, 2, 3),
('HP 15s-fy5007TU (91R03PA)', 55490, 1, 3, 2, 4, 2, 2, 4);

INSERT INTO Laptop_Supplier (LaptopID, SupplierID, Stock, Shipping_Time) VALUES
(1, 1, 50, '3-5 days'),
(2, 2, 30, '5-7 days'),
(3, 3, 20, '7-10 days'),
(4, 1, 25, '4-6 days');
```

ER Diagram:

Schema Design

The database schema consists of 10 tables, each representing a specific aspect of the laptop data:

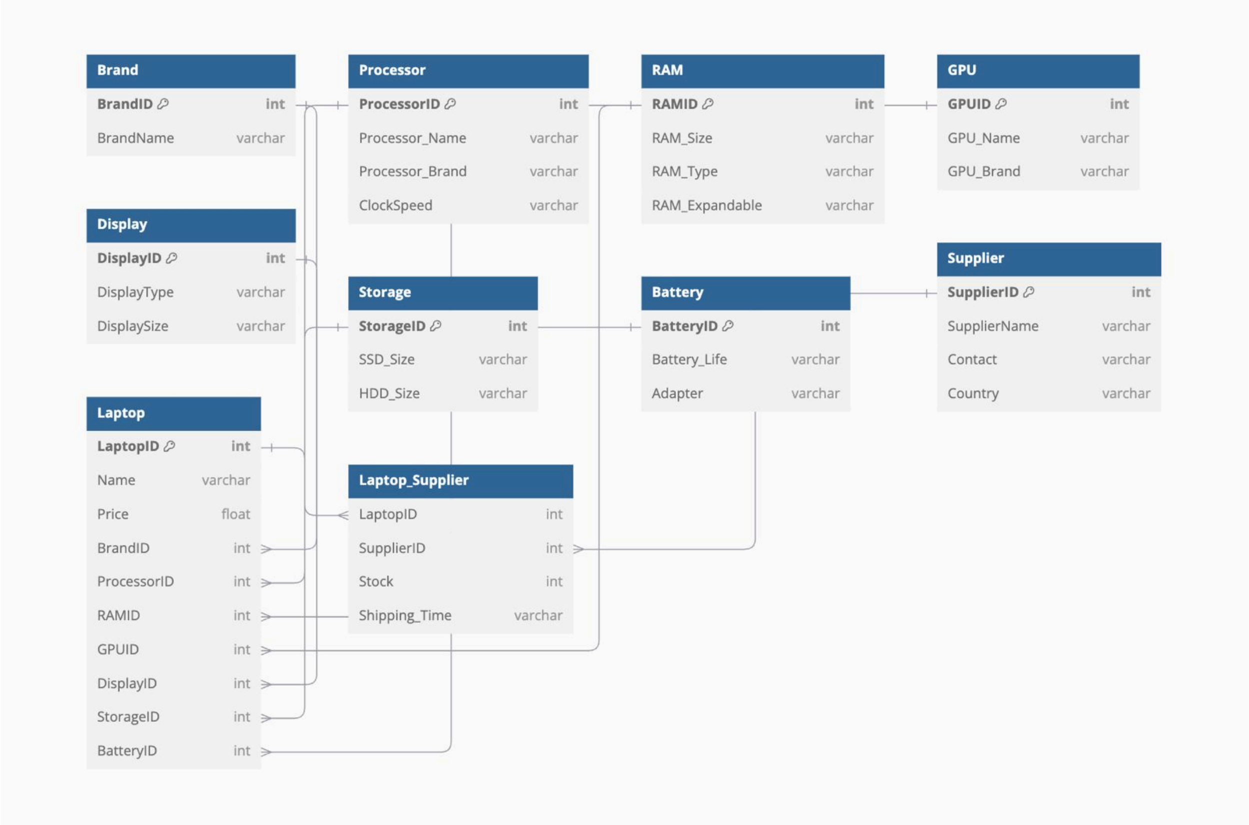
- **Brand**
- **Processor**
- **RAM**
- **GPU**
- **Display**
- **Storage**
- **Battery**
- **Supplier**
- **Laptop**
- **Laptop_Supplier**

This shows the centrality of the Laptop table to the schema, as the various component tables each link to it via their foreign keys. The many-to-many relationship between laptops and their suppliers is handled via the Laptop_Supplier table.

Description

Diagram Description

Laptop: Central entity connecting to all component tables.
Brand: Laptop brand information; connected to Laptop by BrandID.
Processor, RAM, GPU, Display, Storage, Battery: Specific details of each component are stored here and are connected to Laptop through the respective foreign keys.
Supplier: Supplier information is here; connected to Laptop_Supplier.
Laptop_Supplier: The junction table that manages the many-to-many relationships between laptops and suppliers.



1. Brand Entity

Attributes:

BrandID (Primary Key): Unique identifier for each brand.

BrandName: The name of the brand.

ER Diagram Representation:

Entity: Brand

Attributes:

- BrandID (PK)

- BrandName

2. Processor Entity

Attributes:

ProcessorID (Primary Key): Unique identifier for each processor.

Processor_Name: Name of the processor.

Processor_Brand: Brand of the processor (e.g., Intel, AMD).

ClockSpeed: Clock speed of the processor (e.g., 2.6 GHz).

ER Diagram Representation:

Entity: Processor

Attributes:

- ProcessorID (PK)

- Processor_Name

- Processor_Brand

- ClockSpeed

3. RAM Entity

Attributes:

RAMID (Primary Key): Unique identifier for each RAM.

RAM_Size: Size of the RAM (e.g., 8 GB, 16 GB).

RAM_Type: Type of RAM (e.g., DDR4, LPDDR5).

RAM_Expandable: Indicates if the RAM is expandable (Yes or No).

ER Diagram Representation:

Entity: RAM

Attributes:

- RAMID (PK)

- RAM_Size

- RAM_Type

- RAM_Expandable

4. GPU Entity

Attributes:

GPUID (Primary Key): Unique identifier for each GPU.

GPU_Name: Name of the GPU.

GPU_Brand: Brand of the GPU (e.g., NVIDIA, AMD).

ER Diagram Representation:

Entity: GPU

Attributes:

- GPUID (PK)

- GPU_Name

- GPU_Brand

5. Display Entity

Attributes:

DisplayID (Primary Key): Unique identifier for each display.

DisplayType: Type of the display (e.g., LCD, LED).

DisplaySize: Size of the display (e.g., 15.6 inches).

ER Diagram Representation:

Entity: Display

Attributes:

- DisplayID (PK)

- DisplayType

- DisplaySize

6. Storage Entity

Attributes:

StorageID (Primary Key): Unique identifier for each storage configuration.

SSD_Size: Size of the SSD storage (e.g., 512 GB).

HDD_Size: Size of the HDD storage (e.g., 1 TB).

ER Diagram Representation:

Entity: Storage

Attributes:

- StorageID (PK)

- SSD_Size

- HDD_Size

7. Battery Entity

Attributes:

BatteryID (Primary Key): Unique identifier for each battery.

Battery_Life: Battery life in hours (e.g., 8 hours).

Adapter: Type of adapter (e.g., 65W, 90W).

ER Diagram Representation:

Entity: Battery

Attributes:

- BatteryID (PK)

- Battery_Life

- Adapter

8. Supplier Entity

Attributes:

SupplierID (Primary Key): Unique identifier for each supplier.

SupplierName: Name of the supplier.

Contact: Contact details (phone or email) of the supplier.

Country: Country where the supplier is located.

ER Diagram Representation:

Entity: Supplier

Attributes:

- SupplierID (PK)

- SupplierName

- Contact

- Country

9. Laptop Entity

Attributes:

LaptopID (Primary Key): Unique identifier for each laptop.

Name: Model name of the laptop.

Price: Price of the laptop.

BrandID (Foreign Key): References the Brand table.

ProcessorID (Foreign Key): References the Processor table.

RAMID (Foreign Key): References the RAM table.

GPUID (Foreign Key): References the GPU table.

DisplayID (Foreign Key): References the Display table.

StorageID (Foreign Key): References the Storage table.

BatteryID (Foreign Key): References the Battery table.

ER Diagram Representation:

Entity: Laptop

Attributes:

- LaptopID (PK)

- Name

- Price

- BrandID (FK)

- ProcessorID (FK)

- RAMID (FK)

- GPUID (FK)

- DisplayID (FK)

- StorageID (FK)

- BatteryID (FK)

10. Laptop_Supplier Entity (Composite Entity)

Attributes:

LaptopID (Foreign Key): References the Laptop table.

SupplierID (Foreign Key): References the Supplier table.

Stock: Number of units available from this supplier.

Shipping_Time: Time required to ship from the supplier (e.g., 3-5 days).

ER Diagram Representation:

Entity: Laptop_Supplier (Composite Entity)

Attributes:

- LaptopID (FK)

- SupplierID (FK)

- Stock

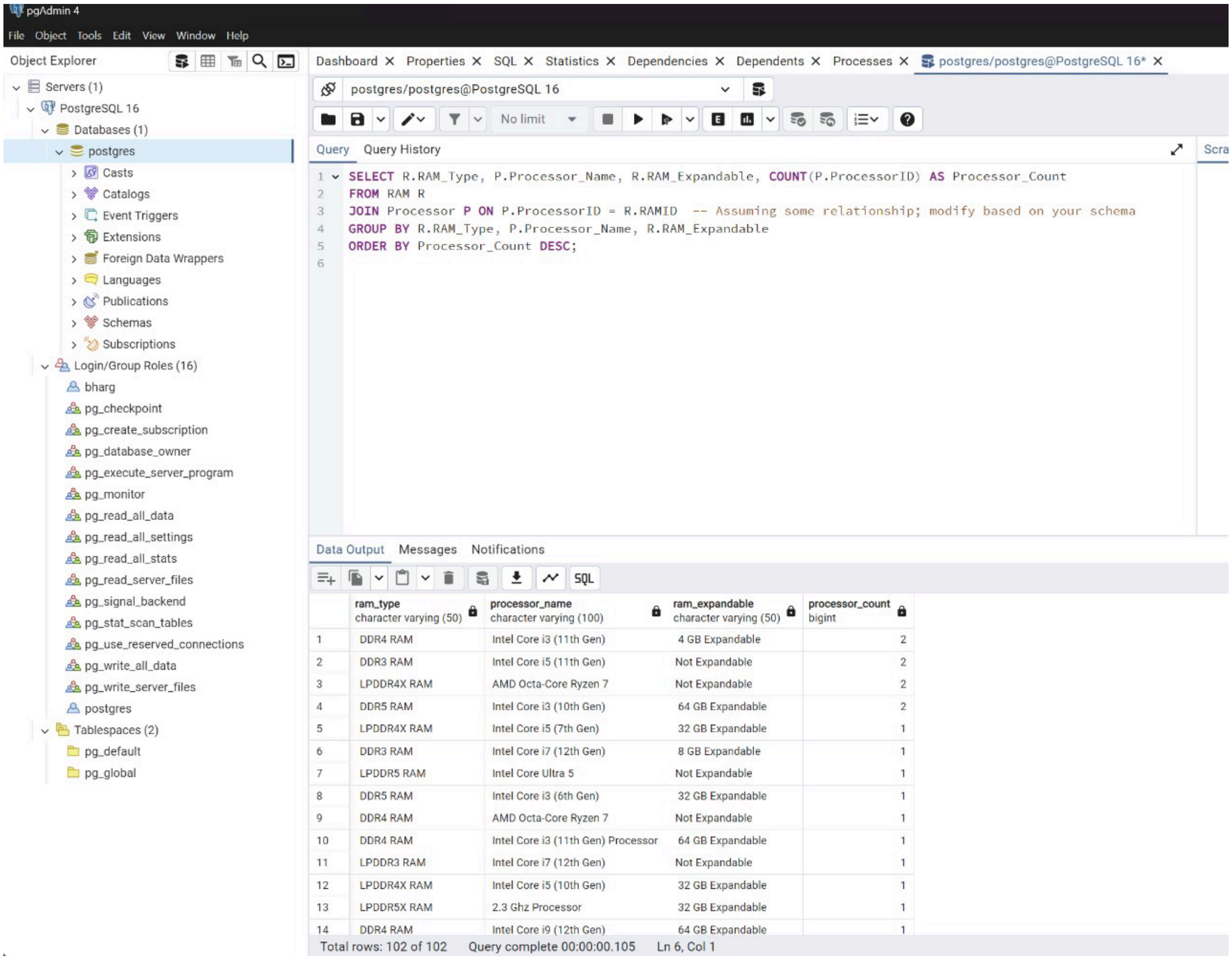
- Shipping_Time

Example Queries:

(1)

```
SELECT R.RAM_Type, P.Processor_Name, R.RAM_Expandable, COUNT(P.ProcessorID) AS Processor_Count
FROM RAM R
JOIN Processor P ON P.ProcessorID = R.RAMID  -- Assuming some relationship; modify based on your schema
GROUP BY R.RAM_Type, P.Processor_Name, R.RAM_Expandable
ORDER BY Processor_Count DESC;
```

ist the processor details for each RAM type along with its expandable capacity



```sql
SELECT R.RAM_Type, P.Processor_Name, R.RAM_Expandable, COUNT(P.ProcessorID) AS Processor_Count
FROM RAM R
JOIN Processor P ON P.ProcessorID = R.RAMID  -- Assuming some relationship; modify based on your schema
GROUP BY R.RAM_Type, P.Processor_Name, R.RAM_Expandable
ORDER BY Processor_Count DESC;
```

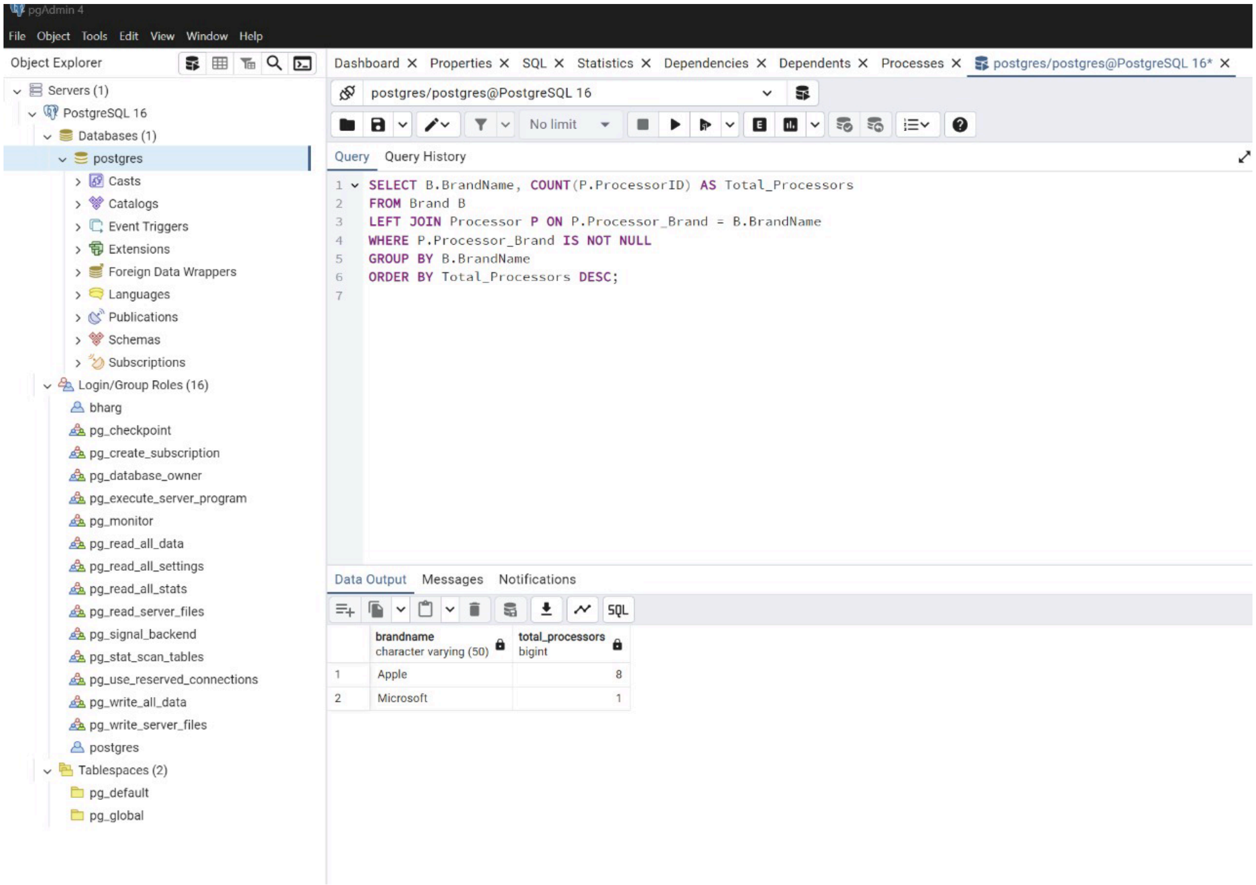
	ram_type character varying (50)	processor_name character varying (100)	ram_expandable character varying (50)	processor_count bigint
1	DDR4 RAM	Intel Core i3 (11th Gen)	4 GB Expandable	2
2	DDR3 RAM	Intel Core i5 (11th Gen)	Not Expandable	2
3	LPDDR4X RAM	AMD Octa-Core Ryzen 7	Not Expandable	2
4	DDR5 RAM	Intel Core i3 (10th Gen)	64 GB Expandable	2
5	LPDDR4X RAM	Intel Core i5 (7th Gen)	32 GB Expandable	1
6	DDR3 RAM	Intel Core i7 (12th Gen)	8 GB Expandable	1
7	LPDDR5 RAM	Intel Core Ultra 5	Not Expandable	1
8	DDR5 RAM	Intel Core i3 (6th Gen)	32 GB Expandable	1
9	DDR4 RAM	AMD Octa-Core Ryzen 7	Not Expandable	1
10	DDR4 RAM	Intel Core i3 (11th Gen) Processor	64 GB Expandable	1
11	LPDDR3 RAM	Intel Core i7 (12th Gen)	Not Expandable	1
12	LPDDR4X RAM	Intel Core i5 (10th Gen)	32 GB Expandable	1
13	LPDDR5X RAM	2.3 Ghz Processor	32 GB Expandable	1
14	DDR4 RAM	Intel Core i9 (12th Gen)	64 GB Expandable	1

Total rows: 102 of 102 Query complete 00:00:00.105 Ln 6, Col 1

List the processor details for each RAM type along with its expandable capacity.

(2)

```sql
SELECT B.BrandName, COUNT(P.ProcessorID) AS Total_Processors
FROM Brand B
LEFT JOIN Processor P ON P.Processor_Brand = B.BrandName
WHERE P.Processor_Brand IS NOT NULL
GROUP BY B.BrandName
ORDER BY Total_Processors DESC;
```



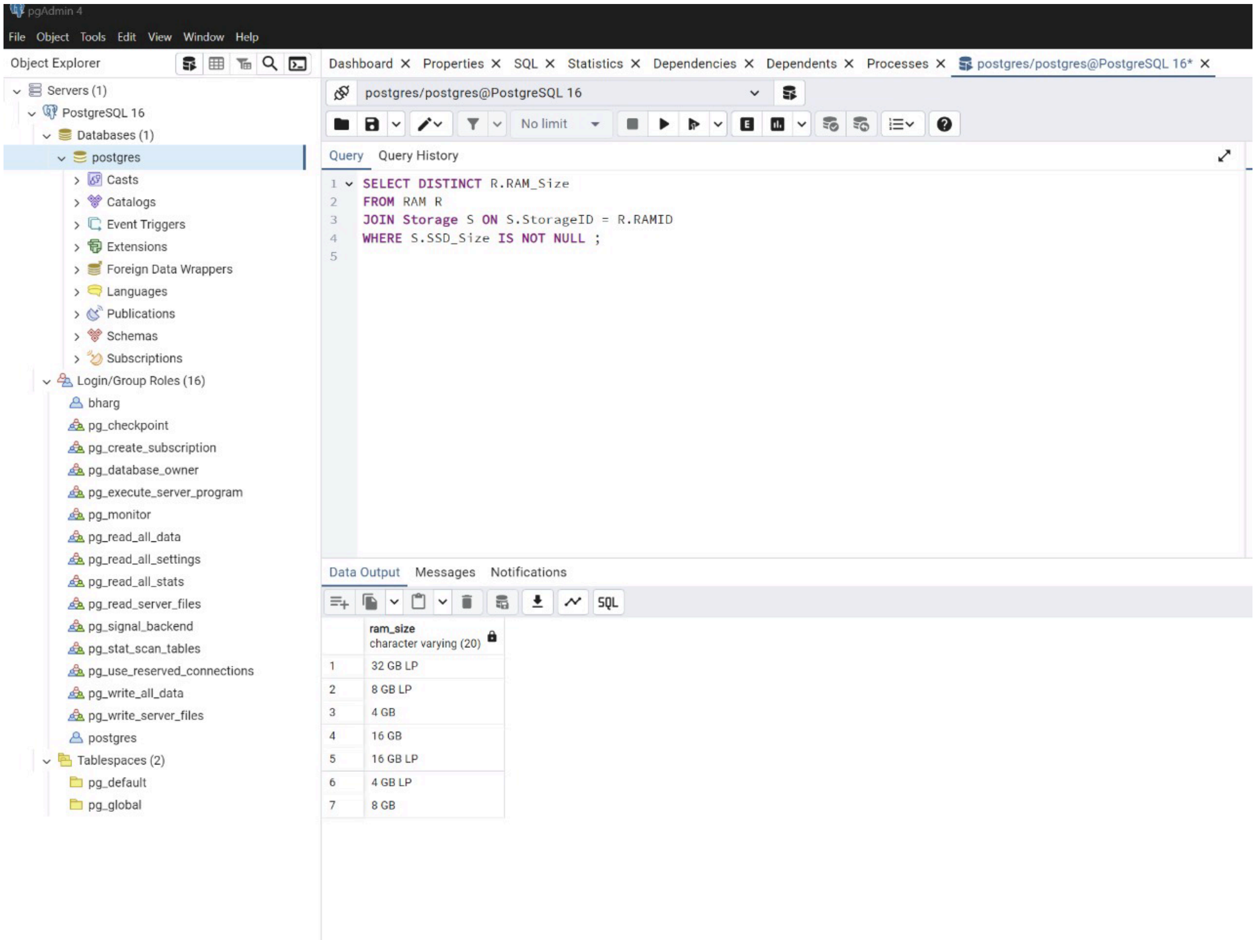
```
SELECT B.BrandName, COUNT(P.ProcessorID) AS Total_Processors
FROM Brand B
LEFT JOIN Processor P ON P.Processor_Brand = B.BrandName
WHERE P.Processor_Brand IS NOT NULL
GROUP BY B.BrandName
ORDER BY Total_Processors DESC;
```

	brandname character varying (50)	total_processors bigint
1	Apple	8
2	Microsoft	1

Find the total number of processors available for each brand.

(3)

SELECT DISTINCT R.RAM_Size
FROM RAM R
JOIN Storage S ON S.StorageID = R.RAMID
WHERE S.SSD_Size IS NOT NULL ;



```
SELECT DISTINCT R.RAM_Size
FROM RAM R
JOIN Storage S ON S.StorageID = R.RAMID
WHERE S.SSD_Size IS NOT NULL ;
```

	ram_size character varying (20)
1	32 GB LP
2	8 GB LP
3	4 GB
4	16 GB
5	16 GB LP
6	4 GB LP
7	8 GB

Retrieve the distinct RAM sizes that are associated with systems having SSD storage.

(4)

SELECT P.Processor_Brand, R.RAM_Type, COUNT(P.ProcessorID) AS Total_Processors

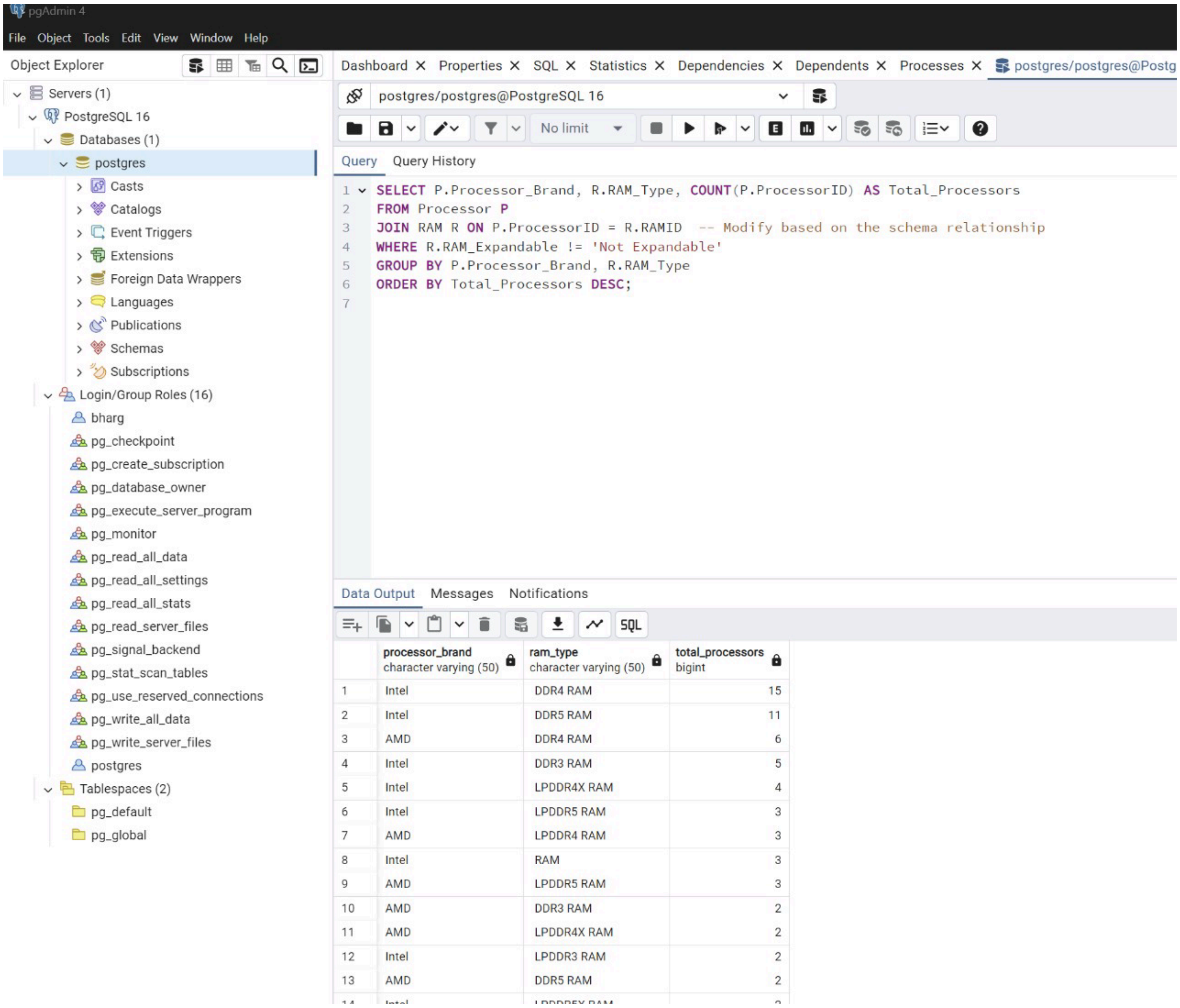
FROM Processor P

JOIN RAM R ON P.ProcessorID = R.RAMID -- Modify based on the schema relationship

WHERE R.RAM_Expandable != 'Not Expandable'

GROUP BY P.Processor_Brand, R.RAM_Type

ORDER BY Total_Processors DESC;



```sql
SELECT P.Processor_Brand, R.RAM_Type, COUNT(P.ProcessorID) AS Total_Processors
FROM Processor P
JOIN RAM R ON P.ProcessorID = R.RAMID  -- Modify based on the schema relationship
WHERE R.RAM_Expandable != 'Not Expandable'
GROUP BY P.Processor_Brand, R.RAM_Type
ORDER BY Total_Processors DESC;
```

	processor_brand character varying (50)	ram_type character varying (50)	total_processors bigint
1	Intel	DDR4 RAM	15
2	Intel	DDR5 RAM	11
3	AMD	DDR4 RAM	6
4	Intel	DDR3 RAM	5
5	Intel	LPDDR4X RAM	4
6	Intel	LPDDR5 RAM	3
7	AMD	LPDDR4 RAM	3
8	Intel	RAM	3
9	AMD	LPDDR5 RAM	3
10	AMD	DDR3 RAM	2
11	AMD	LPDDR4X RAM	2
12	Intel	LPDDR3 RAM	2
13	AMD	DDR5 RAM	2
14	Intel	LPDDR5X RAM	2

Retrieve the processor brands that are associated with RAM types that have expandable capacity.

Loading the Dataset

COPY Brand(BrandID, BrandName)
FROM '/path/to/Brand.csv' DELIMITER ',' CSV HEADER;

COPY Processor(ProcessorID, Processor_Name, Processor_Brand, ClockSpeed)
FROM '/path/to/Processor.csv' DELIMITER ',' CSV HEADER;

COPY RAM(RAMID, RAM_Size, RAM_Type, RAM_Expandable)
FROM '/path/to/RAM.csv' DELIMITER ',' CSV HEADER;

```
COPY GPU(GPUID, GPU_Name, GPU_Brand)
FROM '/path/to/GPU.csv' DELIMITER ',' CSV HEADER;

COPY Display(DisplayID, DisplayType, DisplaySize)
FROM '/path/to/Display.csv' DELIMITER ',' CSV HEADER;

COPY Storage(StorageID, SSD_Size, HDD_Size)
FROM '/path/to/Storage.csv' DELIMITER ',' CSV HEADER;

COPY Battery(BatteryID, Battery_Life, Adapter)
FROM '/path/to/Battery.csv' DELIMITER ',' CSV HEADER;

COPY Supplier(SupplierID, SupplierName, Contact, Country)
FROM '/path/to/Supplier.csv' DELIMITER ',' CSV HEADER;

COPY Laptop(LaptopID, Name, Price, BrandID, ProcessorID, RAMID, GPUID, DisplayID,
StorageID, BatteryID)
FROM '/path/to/Laptop.csv' DELIMITER ',' CSV HEADER;

COPY Laptop_Supplier(LaptopID, SupplierID, Stock, Shipping_Time)
FROM '/path/to/Laptop_Supplier.csv' DELIMITER ',' CSV HEADER;
```

Verification

The following verification steps were performed post-loading:

Record Counts: Record count was checked to make sure it was as expected.

Foreign Key Checks: The referential integrity between tables was validated.

Example Questions Executed queries to verify consistency and integrity of data.