PC Component Database Report

William Gonzalez

Sullivan University

CSC210: Database Design

Joe Martinez

June 8, 2025

After learning more about databases, it seems to be clearer on what they are used for and how they are used. At first, I was under the impression that databases were just giant file storage systems managed by groups of people. I never actually knew how the file/record system was implemented and handled. Now I understand how the whole design and implementation process works, and after being able to design, structure and create my own, I feel that I will be able to implement some of the knowledge into my own projects in the future.

This database is somewhat like a basic database system from what I've learned so far and the way I see it. The purpose of this database is to have records of PC parts. The design is made to keep data about companies who make PC components, PC components themselves, and store information. The process for designing this started with receiving what the database should be about, creating data models (Conceptual, Logical and Physical), creating the script to make the database and revising any values in the tables to make data handling simple.

Starting with the project idea, as mentioned earlier, it is to keep data regarding pc components, the brands that make them, and any store information. A total of 10 tables, 6 main tables which are the "brand" table, "store" table, "CPU" table, "GPU" table, "Motherboard" table, and finally the "RAM" table. The other 4 tables are junction tables, which link tables with a many-to-many relationship. These junction tables link the PC component tables with the brand table.

Once all information of what the database should be about, the next step was creating the data models that help visualize how the database should be constructed. The first data model was the Conceptual data model, and this contained the main parts of the tables which were just the names of the six main tables and links between them. The Logical data model includes attribute names of the tables (name of product, what brand, etc.) and what type of relation each table has

(one-to-one, many-to-many, etc.). The Physical data model adds more description and detail of how the tables are structured, like the addition of primary and foreign keys and junction tables that help link tables with many-to-many relationships.

After the creation of the data models, the writing of the database script is next. The script contains all the commands that create the database, its tables, attributes, and entities. This file contains Data Definition Language (DDL) and Data Manipulation Language (DML) commands. DDL commands in SQL are used to structure a database and its tables with commands using CREATE, DROP, ALTER, and other commands. DDL commands are used to insert, remove, or alter data in a database using INSERT, UPDATE, DELETE, and other commands.

With the use of the CREATE DATABASE command, I created the database with the name PcParts. With the CREATE TABLE command I created the 10 tables with each table having their own field or column names, data type values, primary/foreign keys, and some constraints. With the use of the INSERT INTO command, values were entered into each column for each table to fill the database with data. All this information regarding table structures was based on the data models.

One thing to note about some of the tables is that the junction tables were created because of the many-to-many relationship between the PC component tables and the brand table. These junction tables link those tables together using constraints and foreign keys. The foreign keys in the junction tables are set to reference the primary keys in the other two tables.

Overall, this database has been a learning experience and good practice in database design. So far it seems to handle the basic query commands with basic conditions to pull data from the tables. The only obstacles that appeared were revising some column names to make

them more understandable of their purpose and picking which data types to use for columns for future data input. Another learning obstacle was figuring out and getting comfortable with JOIN commands to pull data from the tables joined with junction tables. The JOIN commands were used to pull certain records based on certain criteria. For example, pulling records from the GPU table that were related to records in the Brand table. Other than that, everything seemed okay and hopefully this project is a good representation of what I've learned so far.

Images For References

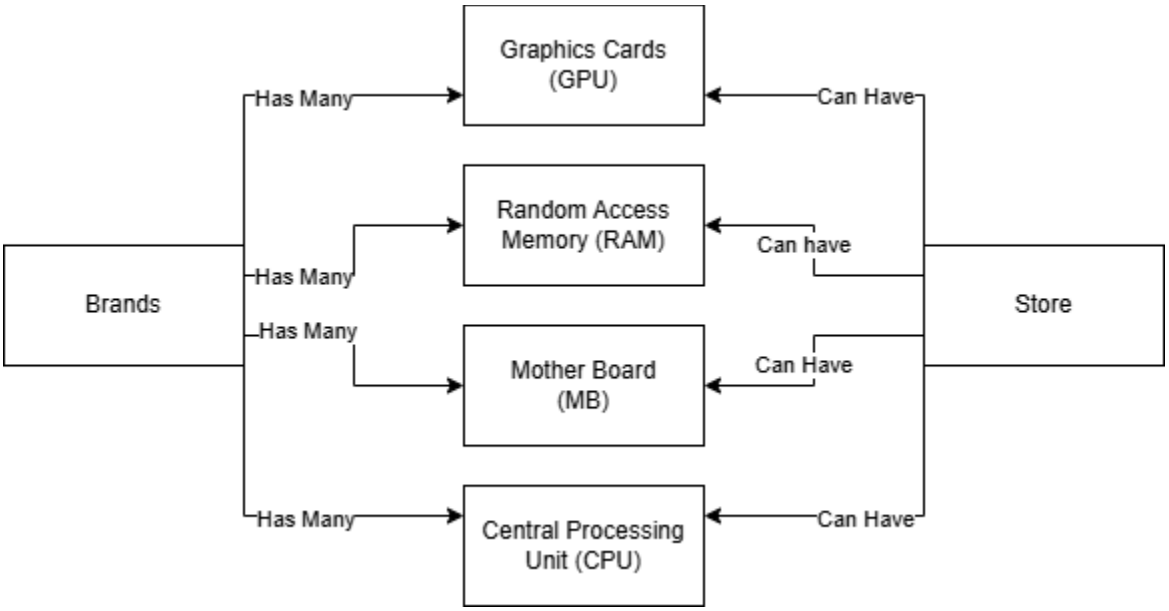
```
CREATE DATABASE PcParts;
USE PcParts;

CREATE TABLE Brand(
BrandID INT AUTO_INCREMENT PRIMARY KEY,
CompanyName VARCHAR (255) NOT NULL,
CountryBased VARCHAR (255),
OwnerName VARCHAR (255),
Founded DATE,
CustomerServNum VARCHAR (20)
);
```

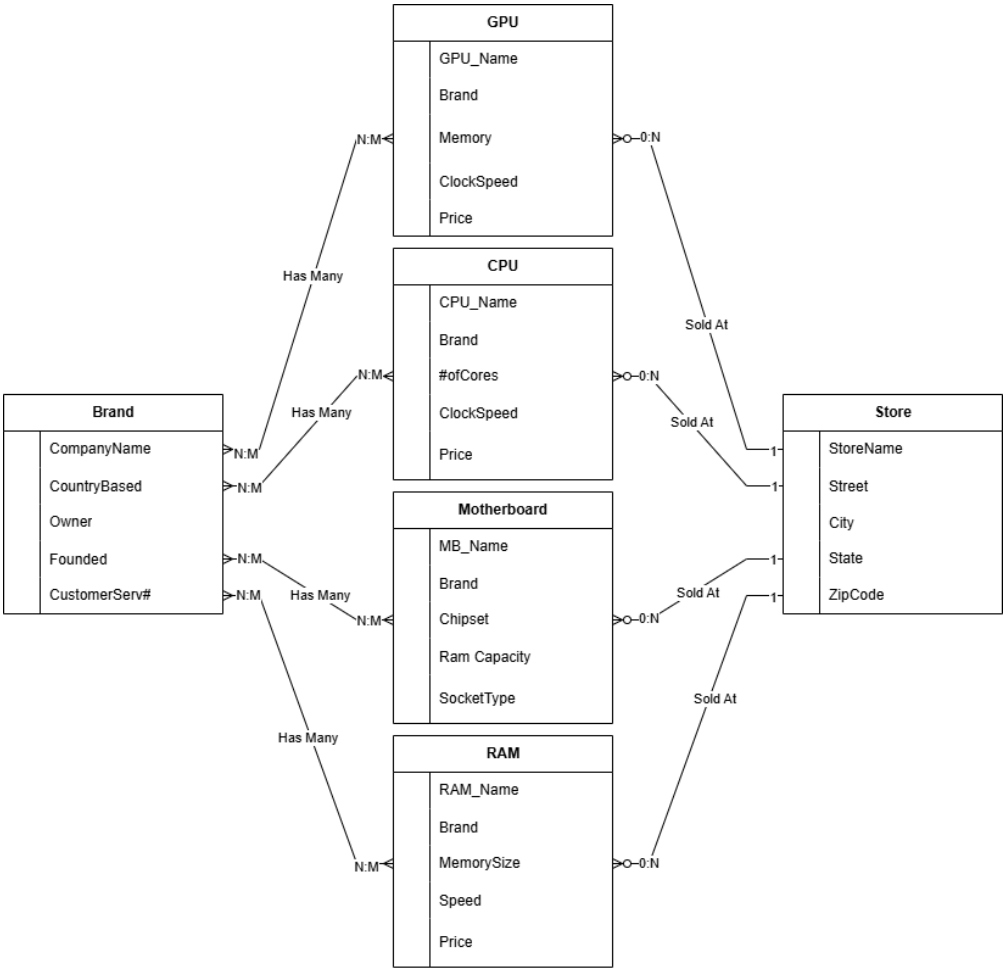
```
INSERT INTO Brand(CompanyName, CountryBased, OwnerName, Founded, CustomerServNum)
Values
('MSI','Taiwan','Manu Shah','1986-8-4','1(626)271-1004'),
('AMD','United States','Jerry Sanders','1969-5-1','1(877)284-1566'),
('Corsair','United States','Andy Paul','1994-1-10','1(888)222-4346'),
('Intel','United States','Lip-Bu Tan','1968-7-18','1(916)277-7000'),
('ASUS','Taiwan','Tzu-Hsien Tung','1989-10-12','1(888)678-3688'),
('G.Skill','Taiwan','Johnson Huang','1989-12-24','1(909)598-6860');
```

BrandID	CompanyName	CountryBased	OwnerName	Founded	CustomerServNum
1	MSI	Taiwan	Manu Shah	1986-08-04	1(626)271-1004
2	AMD	United States	Jerry Sanders	1969-05-01	1(877)284-1566
3	Corsair	United States	Andy Paul	1994-01-10	1(888)222-4346
4	Intel	United States	Lip-Bu Tan	1968-07-18	1(916)277-7000
5	ASUS	Taiwan	Tzu-Hsien Tung	1989-10-12	1(888)678-3688
6	G.Skill	Taiwan	Johnson Huang	1989-12-24	1(909)598-6860
NULL	NULL	NULL	NULL	NULL	NULL

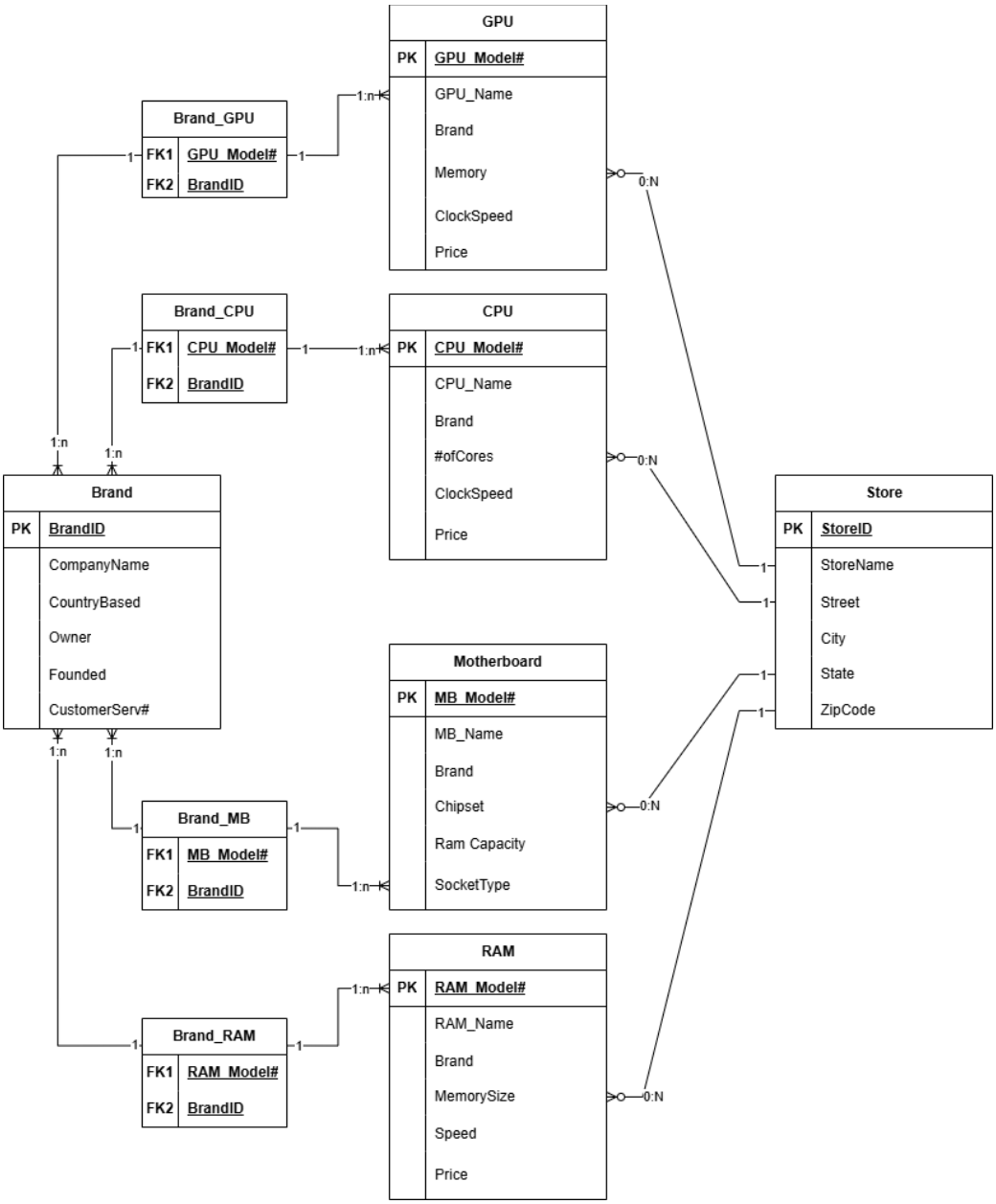
Conceptual Model:



Logical Model:



Physical Model:



Database Script

```
CREATE DATABASE PcParts;
USE PcParts;

CREATE TABLE Brand(
BrandID INT AUTO_INCREMENT PRIMARY KEY,
CompanyName VARCHAR (255) NOT NULL,
CountryBased VARCHAR (255),
OwnerName VARCHAR (255),
Founded DATE,
CustomerServNum VARCHAR (20)
);

CREATE TABLE Store(
StoreID INT AUTO_INCREMENT PRIMARY KEY,
StoreName VARCHAR (255) NOT NULL,
Street VARCHAR (255),
City VARCHAR (255),
State VARCHAR (255),
ZipCode VARCHAR (255)
);

CREATE TABLE GPU(
GPU_ModelNum VARCHAR (255) PRIMARY KEY,
GPU_Name VARCHAR (255) NOT NULL,
Brand VARCHAR (255) NOT NULL,
Memory VARCHAR (255) NOT NULL,
ClockSpeed VARCHAR (255) NOT NULL,
Price Decimal (20,2) NOT NULL
```

```
);

CREATE TABLE CPU(
CPU_ModelNum VARCHAR (255) PRIMARY KEY,
CPU_Name VARCHAR (255) NOT NULL,
Brand VARCHAR (255) NOT NULL,
CoreNum INT NOT NULL,
ClockSpeed VARCHAR (255) NOT NULL,
Price Decimal (20,2) NOT NULL
);

CREATE TABLE Motherboard(
MB_ModelNum VARCHAR (255) PRIMARY KEY,
MB_Name VARCHAR (255) NOT NULL,
Brand VARCHAR (255) NOT NULL,
Chipset VARCHAR (255) NOT NULL,
SocketType VARCHAR (255) NOT NULL,
Price Decimal (20,2) NOT NULL
);

CREATE TABLE RAM(
RAM_ModelNum VARCHAR (255) PRIMARY KEY,
RAM_Name VARCHAR (255) NOT NULL,
Brand VARCHAR (255) NOT NULL,
MemorySize VARCHAR (255) NOT NULL,
Speed VARCHAR (255) NOT NULL,
Price Decimal (20,2) NOT NULL
);

CREATE TABLE Brand_GPU(
```

```
GPU_ModelNum VARCHAR(255),
BrandID INT,
CONSTRAINT GPU_ModelNum_FK FOREIGN KEY (GPU_ModelNum) REFERENCES GPU(GPU_ModelNum),
CONSTRAINT BrandID_GPU_FK FOREIGN KEY (BrandID) REFERENCES Brand(BrandID)
);

CREATE TABLE Brand_CPU(
CPU_ModelNum VARCHAR(255),
BrandID INT,
CONSTRAINT CPU_ModelNum_FK FOREIGN KEY (CPU_ModelNum) REFERENCES CPU(CPU_ModelNum),
CONSTRAINT BrandID_CPU_FK FOREIGN KEY (BrandID) REFERENCES Brand(BrandID)
);

CREATE TABLE Brand_MB(
MB_ModelNum VARCHAR(255),
BrandID INT,
CONSTRAINT MB_ModelNum_FK FOREIGN KEY (MB_ModelNum) REFERENCES Motherboard(MB_ModelNum),
CONSTRAINT BrandID_MB_FK FOREIGN KEY (BrandID) REFERENCES Brand(BrandID)
);
CREATE TABLE Brand_RAM(
RAM_ModelNum VARCHAR(255),
BrandID INT,
CONSTRAINT RAM_ModelNum_FK FOREIGN KEY (RAM_ModelNum) REFERENCES RAM(RAM_ModelNum),
CONSTRAINT BrandID_RAM_FK FOREIGN KEY (BrandID) REFERENCES Brand(BrandID)
);
```

```
INSERT INTO Brand(CompanyName, CountryBased, OwnerName, Founded,
CustomerServNum)
Values
('MSI','Taiwan','Manu Shah','1986-8-4','1(626)271-1004'),
('AMD','United States','Jerry Sanders','1969-5-1','1(877)284-1566'),
('Corsair','United States','Andy Paul','1994-1-10','1(888)222-4346'),
('Intel','United States','Lip-Bu Tan','1968-7-18','1(916)277-7000'),
('ASUS','Taiwan','Tzu-Hsien Tung','1989-10-12','1(888)678-3688'),
('G.Skill','Taiwan','Johnson Huang','1989-12-24','1(909)598-6860');

INSERT INTO Store(StoreName, Street, City, State, ZipCode)
Values
('Micro Center','11755 Mosteller Rd','Sharonville','OH','45241'),
('Best Buy','5085 Shelbyville Rd','Louisville','KY','40207'),
('KyTrade Computer Services','373 Virginia Ave UNIT 140','Lexington','KY','40504'),
('WTR Computer Sales','10106 Taylorsville Rd','Jeffersontown','KY','40299'),
('PCBros','11400 Bluegrass Pkwy','Louisville','KY','40299');

INSERT INTO GPU(GPU_ModelNum, GPU_Name, Brand, Memory, ClockSpeed, Price)
Values
('ARAOR5090','RTX 5090 ASUS ROG Astral OC','ASUS','32GB','2010 MHz','3372.98'),
('MGTOR5090','RTX 5090 MSI Gaming Trio OC','MSI','32GB','2010 MHz','3049.99'),
('ATGOR5080','RTX 5080 ASUS TUF Gaming OC','ASUS','16GB','2300 MHz','1599.99'),
('MS3OR5080','RTX 5080 MSI Shadow 3X OX','MSI','16GB','2300 MHz','1399.99'),
('ATGOR5070','RTX 5070 ASUS TUF Gaming OC','ASUS','12GB','2160 MHz','739.99');

INSERT INTO CPU(CPU_ModelNum, CPU_Name, Brand, CoreNum, ClockSpeed, Price)
Values
('AR959X','AMD Ryzen 9 5900X','AMD','12','3.7 GHz','238.95'),
('AR758X','AMD Ryzen 7 5800X','AMD','8','2.8 GHz','160.54'),
```

('AR556X','AMD Ryzen 5 5600X','AMD','6','3.7 GHz','135.00'),
('IC9109K','Intel Core i9-10900K','Intel','10','3.7 GHz','499.00'),
('IC7107K','Intel Core i7-10700K','Intel','8','3.8 GHz','304.99');

INSERT INTO Motherboard(MB_ModelNum, MB_Name, Brand, Chipset, SocketType, Price)
Values
('MMB550TM','MSI MAG B550 Tomahawk Max WiFi','MSI','B550','AM4','185.99'),
('ARSB550GW','ASUS ROG Strix B550-F Gaming WiFi II','ASUS','B550','AM4','154.99'),
('ATGB550W','ASUS TUF Gaming B550-Plus WiFi II','ASUS','B550','AM4','139.99'),
('MMZ490GEW','MSI MPG Z490 Gaming Edge WiFi','MSI','Z490','LGA1200','419.00'),
('ARSZ590GW','ASUS ROG Strix Z590-E Gaming WiFi','ASUS','Z590','LGA1200','519.00');

INSERT INTO RAM(RAM_ModelNum, RAM_Name, Brand, MemorySize, Speed, Price)
Values
('CVLPX16','Corsair Vengeance LPX 16 GB','Corsair','16 GB','3200 MHz','37.99'),
('CVRGBP32','Corsair Vengeance RGB Pro 32 GB','Corsair','32 GB','3600 MHz','86.99'),
('CDPRGB32','Corsair Dominator Platinum RGB 32 GB','Corsair','32 GB','3200 MHz','111.99'),
('GSRV32','G.Skill Ripjaws V 32 GB','G.Skill','32 GB','3200 MHz','51.99'),
('GSTZRGB32','G.Skill Trident Z RGB 32 GB','G.Skill','32 GB','3600 MHz','69.99');

INSERT INTO Brand_GPU(GPU_ModelNum, BrandID)
Values
('ARAOR5090',5),
('MGTOR5090',1),
('ATGOR5080',5),
('MS3OR5080',1),
('ATGOR5070',5);

INSERT INTO Brand_CPU(CPU_ModelNum, BrandID)
Values

```
('AR959X',2),
('AR758X',2),
('AR556X',2),
('IC9109K',4),
('IC7107K',4);

INSERT INTO Brand_MB(MB_ModelNum, BrandID)
Values
('MMB550TM',1),
('ARSB550GW',5),
('ATGB550W',5),
('MMZ490GEW',1),
('ARSZ590GW',5);

INSERT INTO Brand_RAM(RAM_ModelNum, BrandID)
Values
('CVLPX16',3),
('CVRGBP32',3),
('CDPRGB32',3),
('GSRV32',6),
('GSTZRGB32',6);

CREATE VIEW Brand_List AS
SELECT *
FROM Brand;

CREATE VIEW Store_List AS
SELECT *
FROM Store;
```

```
CREATE VIEW GPU_List AS
SELECT *
FROM GPU;

CREATE VIEW CPU_List AS
SELECT *
FROM CPU;

CREATE VIEW MB_List AS
SELECT *
FROM Motherboard;

CREATE VIEW RAM_List AS
SELECT *
FROM RAM;
```