**Database Design and Implementation for Cell Technologies**

*Georgia State University*

*Fundamentals of Database Management Systems*

*Summer 2024*

*Instructor: Parasuraman Nurani*

*Courtney Baker*

This detailed report was created as part of my Database Administration class at Georgia State University. The assignment required my group to identify a business problem for a hypothetical organization, Cell Technologies, and design a database solution to address this problem. The report includes an analysis of the company's data needs, the design of an Entity-Relationship (E-R) Diagram, and the implementation of SQL queries for data manipulation. My contribution involved conducting the initial research, designing the E-R Diagram, creating the logical and physical database designs, and developing the SQL queries. This project demonstrates my ability to apply database concepts to real-world scenarios, showcasing my creative and critical thinking, as well as my ability to convey complex material clearly.

**Database Design and Implementation for Cell Technologies**

*Georgia State University*

*Fundamentals of Database Management Systems*

*Section 002 Summer 2024*

*Instructor: Parasuraman Nurani*

*Courtney Baker, Hunter Wakefield, Kyla James, Mingxin Chen, Yahye Abdi*

**Organization Description**

Cell Technologies is a small firm that specializes in building custom PCs with a limited catalog. The company prides itself on delivering tailored computer systems that meet specific customer needs, whether for gaming, productivity, or general use. With a strong focus on quality and customer satisfaction, Cell Technologies offers a curated selection of high-performance components and provides expert advice to ensure each build is optimized for its intended purpose.

**Business Problem**

The primary business problem addressed is the efficient management and utilization of customer data and PC component information. Cell Technologies needs a robust database solution to catalog customer orders and maintain detailed specifications on each part they offer. The goal is to assist both internal users (employees) and external users (customers) in making informed decisions about component selection and build configurations. This will streamline the process of creating custom PCs, enhance customer experience, and support strategic business decisions.

**Software and Tools Used**

* Kaggle
* Excel
* Draw.io
* Microsoft SQL

**Database Modeling with ER Diagram**

**Entities:**

1. **Order**
   * Attributes: OrderID (PK), FName, LName, CPhone, OrderDate
2. **Build**
   * Attributes: BuildID (PK), OrderID(FK), [Part ID’s] (FK)
3. **Part**
   * **CPU:** CPUid(PK), Brand, Name, CoreCount, BaseSpeed, BoostSpeed, TDP, Price
   * **GPU:** GPUid(PK), Brand, Name, VRAM, BaseSpeed, BoostSpeed, TDP, Price
   * **RAM:** RAMid(PK), Capacity, Price
   * **PSU:** PSUid(PK), Wattage, Price
   * **Storage:** Storageid(PK), Capacity, Price

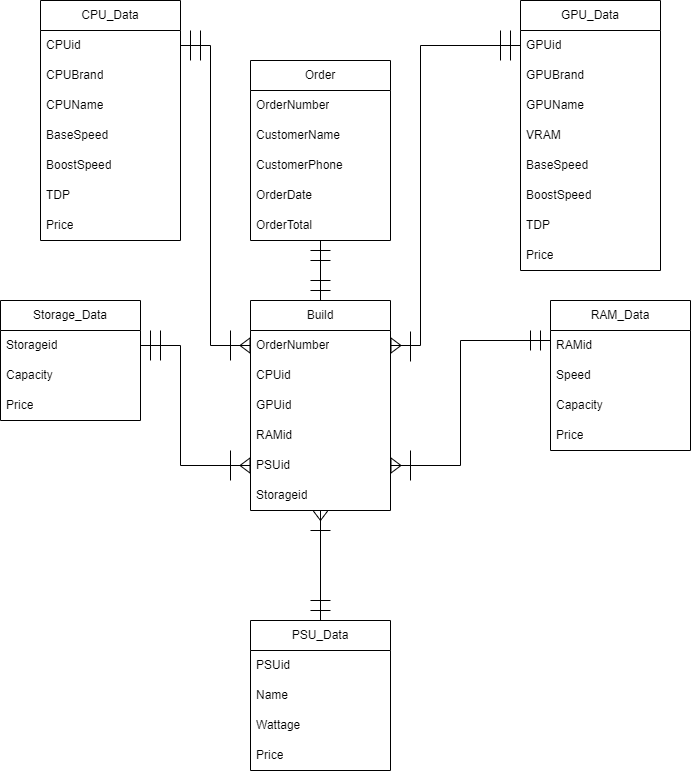
**Relationships:**

This database primarily has two main relationships: the customer data and the different parts of their builds.

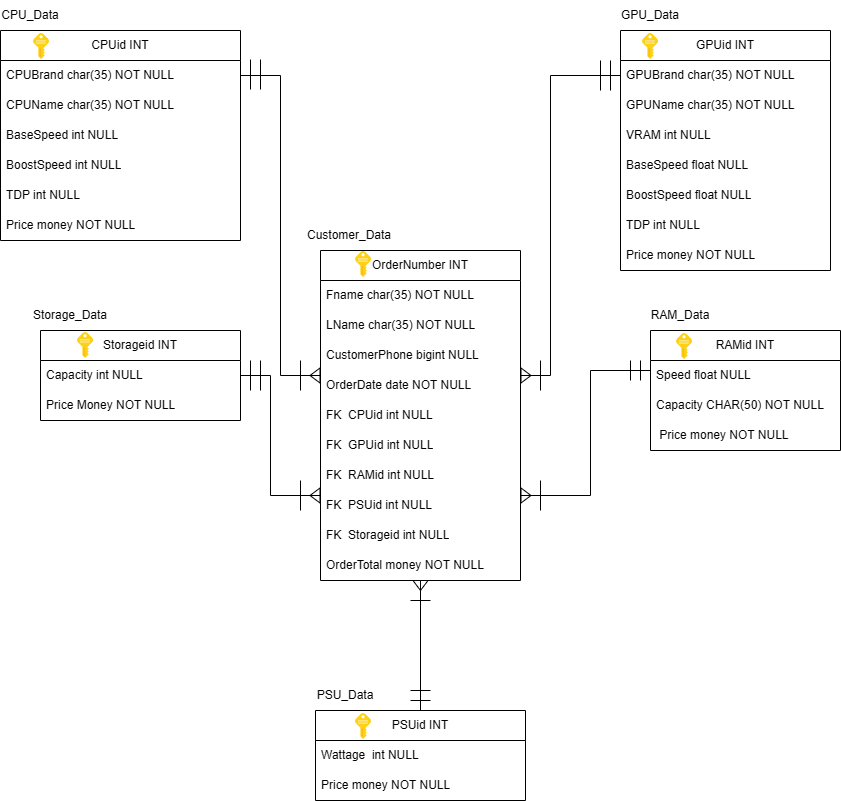
**Customer\_Data to Part:**From Customer\_Data to Part the minimum cardinality is one and maximum cardinality is one. (1:1)

**Part to Customer\_Data:** Part to Customer\_Data has a minimum cardinality of one and a maximum cardinality of many. (1:M)

**Database Model:**



**Physical Model:**

****

**Normalization**

**First Normal Form (1NF):**

* Ensure each table has a primary key.
* Remove repeating groups.

**Second Normal Form (2NF):**

* Remove partial dependencies.

**Third Normal Form (3NF):**

* Remove transitive dependencies.

Each table is in 3NF because all non-key attributes are fully functionally dependent on the primary key, and there are no transitive dependencies. One notable change had to be made where in the initial stages, Customer\_Data and Build were two separate tables. Because these tables had a 1:1 cardinality both ways, it made sense to combine the two. It also made it easier to calculate the order total and add it to the customer data table.

**SQL Queries**

CREATE TABLE Costumer\_Data (

Orderid int NOT NULL,

FName char(55) NOT NULL,

LName char(55) NOT NULL,

CPhon bigint NOT NULL,

OrderDate date NOT NULL,

CPUid int NULL,

GPUid int NULL,

RAMid int NULL,

PSUid int NULL,

Storageid int NULL,

PRIMARY KEY (Orderid))

;

ALTER TABLE Customer\_Data

ADD FOREIGN KEY (CPUid) REFERENCES CPU\_Data(CPUid)

ALTER TABLE Customer\_Data

ADD FOREIGN KEY (GPUid) REFERENCES GPU\_Data(GPUid)

ALTER TABLE Customer\_Data

ADD FOREIGN KEY (RAMid) REFERENCES RAM\_Data(RAMid)

ALTER TABLE Customer\_Data

ADD FOREIGN KEY (PSUid) REFERENCES PSU\_Data(PSUid)

ALTER TABLE Customer\_Data

ADD FOREIGN KEY (Storageid) REFERENCES Storage\_Data(Storageid)

;

* The data in this table consists of basic customer information and id numbers of parts that correspond to the primary key of each related table.

CREATE TABLE CPU\_Data (

CPUid int NOT NULL,

CPUName char(55) NOT NULL,

CoreCount int NOT NULL,

BaseClock float NOT NULL,

BoostClock float NOT NULL,

TDP int NOT NULL,

CPUPrice money NOT NULL,

PRIMARY KEY (CPUid));

CREATE TABLE GPU\_Data (

GPUid int NOT NULL,

GPUName char(55) NOT NULL,

GPUBrand char(55) NOT NULL,

VRAM varchar(55) NOT NULL,

BaseClock float NOT NULL,

BoostClock float NOT NULL,

TDP int NOT NULL,

CPUPrice money NOT NULL,

PRIMARY KEY (CPUid));

* The tables CPU\_Data and GPU\_Data have information about 5 different CPUs and GPUs. This information is the brand, name, clock speeds, thermal design power, and price. The main difference between these two is that CPU\_Data has Core count while GPU\_Data has VRAM amount.

CREATE TABLE RAM\_Data (

RAMid int NOT NULL,

RSpeed int NOT NULL,

Capacity varchar(55) NOT NULL,

Price money NOT NULL,

PRIMARY KEY (RAMid);

* RAM\_Data’s data is similar to the previous two tables, but as this component is random access memory, it has no boost clock speed nor did we bother adding TDP information as this is negligible.

CREATE TABLE PSU\_Data (

PSUid int NOT NULL,

PSUWatt int NOT NULL,

Price money NOT NULL,

PRIMARY KEY (PSUid);

CREATE TABLE Storage\_Data (

Storageid int NOT NULL,

Capacity varchar(55) NOT NULL,

Price money NOT NULL,

PRIMARY KEY (Storageid);

* From an outside standpoint, PSU\_Data and Storage\_Data are essentially the same table under a different name. PSU\_Data deals with the specific wattage of the 3 power supplies, while Storage\_Data has information regarding the capacity of the 3 storage offerings from Cell Tech.

Instead of using an INSERT function for each CSV, we opted to familiarize ourselves with the SQL Server environment by taking advantage of the “Import Data” task within the database environment. First the data was selected from a CSV, the source data types were selected, the destination database and table were selected, then finally the data was converted into usable data types and placed into the specified columns.

CREATE VIEW FULLDetails AS

SELECT Customer\_Data.\*, (CPU\_Data.Price + GPU\_Data.Price + RAM\_Data.Price + PSU\_Data.price + Storage\_Data.price) as OrderTotal

FROM Customer\_Data

INNER JOIN CPU\_Data ON Customer\_Data.CPUid = CPU\_Data.CPUid

INNER JOIN GPU\_Data ON Customer\_Data.GPUid = GPU\_Data.GPUid

INNER JOIN RAM\_Data ON Customer\_Data.RAMid = RAM\_Data.RAMid

INNER JOIN PSU\_Data ON Customer\_Data.PSUid = PSU\_Data.PSUid

INNER JOIN Storage\_Data ON Customer\_Data.Storageid = Storage\_Data.Storageid

WHERE Customer\_Data.CPUid = CPU\_Data.CPUid

;

* This View joins both the Customer\_Data table and a new column created by adding the price of each component based on what id numbers are in Customer\_Data.

**How the Database Implementation Resolves the Business Problem**

The implemented database system provides a structured and efficient way to manage customer orders and part inventories. It allows Cell Technologies to:

* **Track Customer Orders**: Easily add, update, and retrieve customer order details, enhancing customer relationship management.
* **Compare Components**: Enable users to search and compare parts based on specifications and prices, aiding in informed decision-making.
* **Analyze Trends**: Provide insights into popular components and sales trends, supporting strategic business decisions.

**Individual Analysis:**

The database project on Cell Technologies was a comprehensive and collaborative effort aimed at addressing the data management challenges faced by a hypothetical company specializing in custom PCs. The primary objective was to design and implement a database solution that could efficiently manage customer data, order information, and PC component details. This analysis reflects on my individual contributions, the challenges encountered, and the lessons learned throughout the project.

As a key member of the project team, my responsibilities included conducting the initial research to understand the business requirements of Cell Technologies. I took the lead in designing the Entity-Relationship (E-R) Diagram, which served as the blueprint for the database. This involved identifying key entities such as customers, orders, products, and categories, and defining their relationships and attributes. I also played a significant role in the logical and physical design of the database, ensuring that the schema was normalized to eliminate redundancy and enhance data integrity.

In addition to the design phase, I developed the SQL queries for creating tables, inserting data, and updating records. These queries were essential for implementing the database and ensuring it could handle the required operations. My attention to detail and thorough testing of the SQL scripts helped in identifying and resolving any issues, thereby contributing to the robustness of the database solution.

One of the primary challenges encountered was ensuring the database design could handle the complex relationships between different entities, such as the many-to-many relationship between orders and products. To address this, I introduced an associative entity, 'OrderDetails,' which effectively mapped the relationships and captured additional attributes like quantity and price.

Another challenge was optimizing the database for performance, given the potential volume of data Cell Technologies might handle. I implemented indexing on key attributes and carefully designed the queries to minimize execution time and resource consumption. Collaborating with team members, we also conducted extensive testing and validation to ensure the database met all business requirements.

This project provided invaluable insights into the practical application of database concepts. One of the key lessons learned was the importance of thorough planning and design before implementation. The initial E-R Diagram served as a critical guide throughout the project, highlighting the need for a solid foundation in database design.

Additionally, this project reinforced the significance of teamwork and effective communication. Regular meetings and collaborative problem-solving were crucial in addressing challenges and ensuring the project stayed on track. I also learned the importance of testing and validation in developing a reliable database solution.

The database project for Cell Technologies was a significant learning experience that enhanced my skills in database design and implementation. My contributions to the E-R Diagram, logical and physical design, and SQL query development were pivotal in achieving a functional and efficient database solution. The challenges encountered and lessons learned have prepared me for future projects and reinforced my commitment to excellence in the field of database administration.