**Database Project: Technology Store Database**

Corey St Jean

**Introduction:**

For this project, we were tasked with creating a database for scratch for a fictional situation. I chose to create a database for a hypothetical technology store. These databases will be responsible for holding information about customers, employees, products, operating systems, and orders.

**Methods:**

To start, all tables were given an ID attribute that automatically increments to give each row in the table a unique attribute so they can be uniquely identified. This ID attribute has the data type varchar and is also the primary key within each table and all other values within the row are dependent on this attribute and no other attribute(s) ensuring the tables are in 3rd Normal Form. Also, all attributes also have the constraint where they cannot be NULL unless stated otherwise.

Products Table:

The products table contains information about the laptops the store sells. It includes physical specifications like dimensions and weight and technical specifications like OS id, ram, and storage. The OS id attribute is used as a foreign key to the OS table so that OS information like version and type can also be associated with the product.

OS Table:

The OS table is one of the simpler tables in the database. It includes basic OS information for three different OS’s. The brand, name, and version are all varchar data type, and the release date attribute is a date data type.

Customer Table:

The customer table contains information about the customers, both past and present, who have at one point made an order. This information includes an ID, their last name, first name, address, address2 (if they require a second address line), city, state, zip code, phone, and email. Last name, first name, both addresses, city, and email are all the varchar data type, phone is a big integer, zip code is a normal integer, and then state is just the character data type restricted to two characters. The attribute address2 does not have the constraint preventing NULL values because not all addresses require a second line, only apartments, offices, suites, buildings, etc. Then finally, the ID is the primary key for the table.

Employee Table:

The employee table essentially follows the same pattern and methodologies as the customer table, but with the addition of a position attribute. The position attribute is the varchar data type and holds the position of the employee in the company.

Orders Table:

Finally, the orders table just contains integer attributes which are all foreign keys for connecting customers, products, and the employee that packs their order. The count attribute though is not a foreign key and is instead just the count of the product the customer ordered. This table contains three foreign keys in total and the primary key is the ID attribute.

**Entity Relationship (ER) Diagrams:**

A diagram of a company

Description automatically generatedTwo ER diagrams were created for this database. The top one is a conceptual diagram showing basic relationships, entities, and attributes. The bottom one is a logical diagram showing more complex relationships as well as entities and attributes.

A screenshot of a computer

Description automatically generated

**Functional Dependencies:**

Because the database is in third normal form, the primary keys in each of the tables are what the rest of the attributes are functionally dependent on to avoid transitive dependence.

**Products:** ID -> ID, name, brand, storage, ram, dimensions, weight, OS\_id

**OS:** ID -> ID, brand, name, version, release\_date

**Customer:** ID -> last\_name, first\_name, address, address2, city, state, zip\_code, phone, email

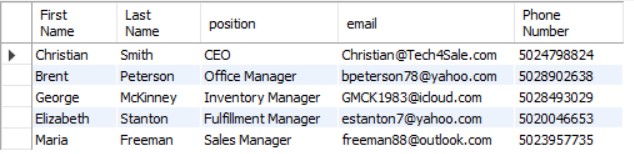
**Employee:** ID -> last\_name, first\_name, address, address2, city, state, zip\_code, phone, email, position

**Orders:** ID -> product\_id, customer\_id, count, employee\_id

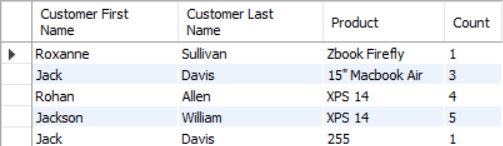
**Query Examples:**

Query One:

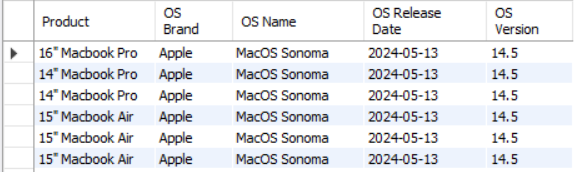
Select first\_name AS ‘First Name’, last\_name AS ‘Last Name’, position, email, phone AS ‘Phone Number’ FROM employees WHERE position = ‘CEO’ OR position LIKE ‘%Manager’;



Query Two:

SELECT customers.first\_name AS 'Customer First Name', customers.last\_name AS 'Customer Last Name', products.name AS Product, orders.count AS Count FROM orders JOIN customers ON orders.customer\_id = customers.cust\_id JOIN products ON orders.product\_id = products.prod\_id;

Query Three:

****SELECT products.name AS Product, os.brand AS 'OS Brand', os.name AS 'OS Name', os.release\_date AS 'OS Release Date', os.version AS 'OS Version' FROM orders JOIN products ON orders.product\_id = products.prod\_id JOIN OS ON products.opsys\_id = os.os\_id;