NORMALIZE ENTITY RELATION DIAGRAM (ERD) FOR TECHCORP.

A screenshot of a computer screen

Description automatically generated

**Design choices for primary and foreign keys in each normalized table.**

Customers Table:

Primary Key: CustomerID

The CustomerID column is chosen as the primary key to uniquely identify each customer record. This ensures that each customer can be distinguished, even if there are multiple customers with the same name.

Suppliers Table:

Primary Key: SupplierID

The SupplierID column is chosen as the primary key to uniquely identify each supplier record. This allows for the association of products with their respective suppliers.

Products Table:

Primary Key: ProductID

The ProductID column is chosen as the primary key to uniquely identify each product record. This ensures that each product can be distinguished, even if there are multiple products with the same name.

Foreign Key: SupplierID

The SupplierID column is used as a foreign key to establish the relationship between products and their respective suppliers. This allows for the retrieval of supplier information associated with each product.

Orders Table:

Primary Key: OrderID

The OrderID column is chosen as the primary key to uniquely identify each order record. This ensures that each order can be distinguished, even if there are multiple orders for the same customer or product.

Foreign Keys: CustomerID

The CustomerID column is used as a foreign key to link each order to the corresponding customer. This allows for the retrieval of customer information associated with each order.

ProductID: The ProductID column is used as a foreign key to link each order to the corresponding product. This allows for the retrieval of product information associated with each order.

**How normalization help reduce redundancy and improve data integrity.**

Normalization help reduce redundancy by breaking down the data into smaller, manageable tables. In the original table, there were multiple instances of the same customer and supplier information repeated across rows. By creating separate customers and suppliers table, the data is stored only once, reducing redundancy.

Normalization also ensures that the data is stored in a consistent and accurate manner, reducing the risk of data anomalies and inconsistencies. By establishing primary and foreign key relationships between the tables, the database can enforce referential integrity.

**Trade-off for Denormalization**

In environments where read operations significantly outnumber write operations (e.g., reporting systems), denormalization can speed up query performance by reducing the number of joins required. Also simplifying complex queries by reducing the number of tables can lead to faster execution times.