Shoe Store Management System database

Step 01: Entity Sets for Shoe Store Management System:

1. Product (Shoes)
2. Category
3. Customer
4. Employee
5. Supplier
6. Inventory
7. Sale
8. Purchase\_Order

Step 02: Properties/attributes of each entity set:

1. Product(product\_id, name, brand, size, color, price, description)
2. Category(category\_id, name, description)
3. Customer(customer\_id, name, email, phone, address)
4. Employee(employee\_id, name, email, phone, hire\_date, salary, role)
5. Supplier(supplier\_id, name, contact\_person, email, phone, address)
6. Inventory(inventory\_id, quantity, last\_updated)
7. Sale(sale\_id, date, total\_amount, payment\_method)
8. Purchase\_Order(order\_id, date, total\_amount, status)

Step 03: Relations between entity sets:

1. Product-Category (belongs\_to): Many-to-One
2. Product-Inventory (has\_stock): One-to-One
3. Product-Sale (sale\_items): Many-to-Many
4. Product-Purchase\_Order (order\_items): Many-to-Many
5. Product-Supplier (supplies): Many-to-Many
6. Sale-Customer (makes): Many-to-One
7. Sale-Employee (processes): Many-to-One
8. Purchase\_Order-Supplier (provides): Many-to-One
9. Purchase\_Order-Employee (manages): Many-to-One

Step 04: E-R diagram of these Entity sets and Relationship Sets is given below:

A diagram of a product

Description automatically generated

Step 05: Conversion of the diagram into tables by using the reduction rule.

(a) Entity set 'Category' is connected with Product via relationship set belongs\_to. Category side is one. So Category entity set directly converted to Category table with the same attributes.

* Category(category\_id, name, description)

(b) The entity sets 'Customer', 'Employee', and 'Supplier' are also converted to their respective tables for similar reasons.

* Customer(customer\_id, name, email, phone, address)
* Employee(employee\_id, name, email, phone, hire\_date, salary, role)
* Supplier(supplier\_id, name, contact\_person, email, phone, address)

(c) Relationship set 'Product\_Supplier' is many-many. So create a table for Product\_Supplier. The attributes will be the primary keys from the corresponding entity sets and descriptive attribute supply\_price.

* Product\_Supplier(product\_id, supplier\_id, supply\_price)

(d) 'Inventory' is a weak entity set related to Product. So the primary key of the identifying strong entity set (Product) will be part of the weak entity set.

* Inventory(product\_id, inventory\_id, quantity, last\_updated)

(e) Relationship sets 'belongs\_to', 'makes', 'processes', 'provides', and 'manages' are one-many. So these relationship sets need not be converted to tables. Rather, the entity sets of the many side are converted to tables by adding the primary key of the one side

* Product(product\_id, name, brand, size, color, price, description, category\_id)
* Sale(sale\_id, date, total\_amount, payment\_method, customer\_id, employee\_id)
* Purchase\_Order(order\_id, date, total\_amount, status, supplier\_id, employee\_id)

(f) Relationship sets 'sale\_items' and 'order\_items' are many-many. So create two tables for these relationships. The attributes will be the primary keys from the corresponding entity sets and descriptive attributes.

* Sale\_Items(sale\_id, product\_id, quantity, unit\_price, subtotal)
* Order\_Items(order\_id, product\_id, quantity, unit\_price, subtotal)