

Step A

For our project, we have a store in an airport. Our project shows the principal entities that can be found in a store. We have the "Sales" side and the "Inventory" side, creating a simple sales and inventory management system.

We identified the principal entities as: Customers, Sales, Products, Inventory, Staff, and Suppliers

Defining Attributes:

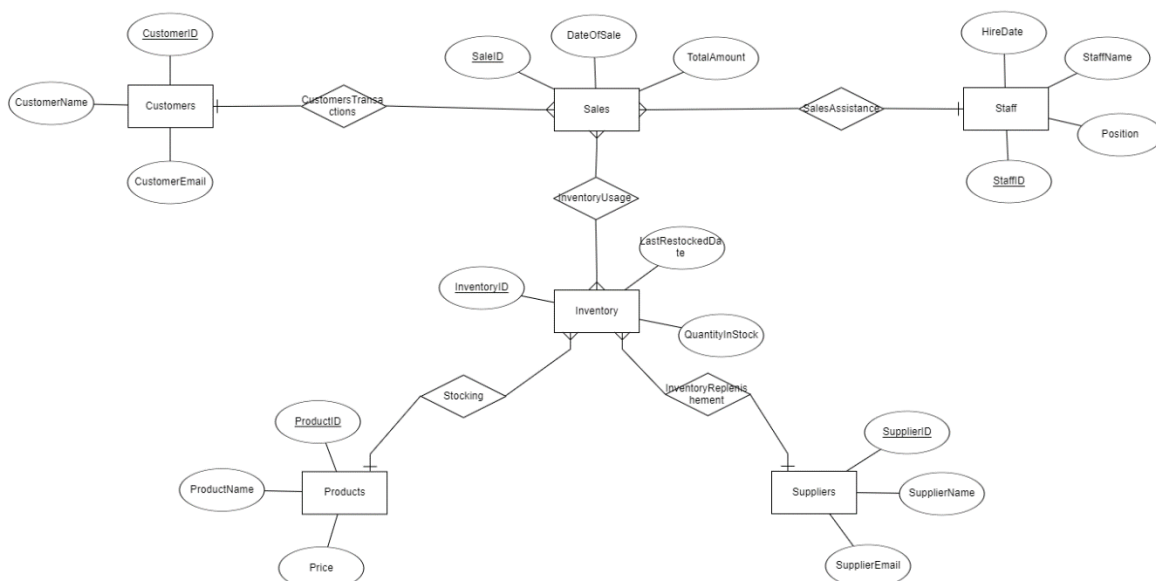
Each entity has attributes that describe its properties. For example, the Customers entity has attributes like CustomerID (primary key), CustomerName, and CustomerEmail.

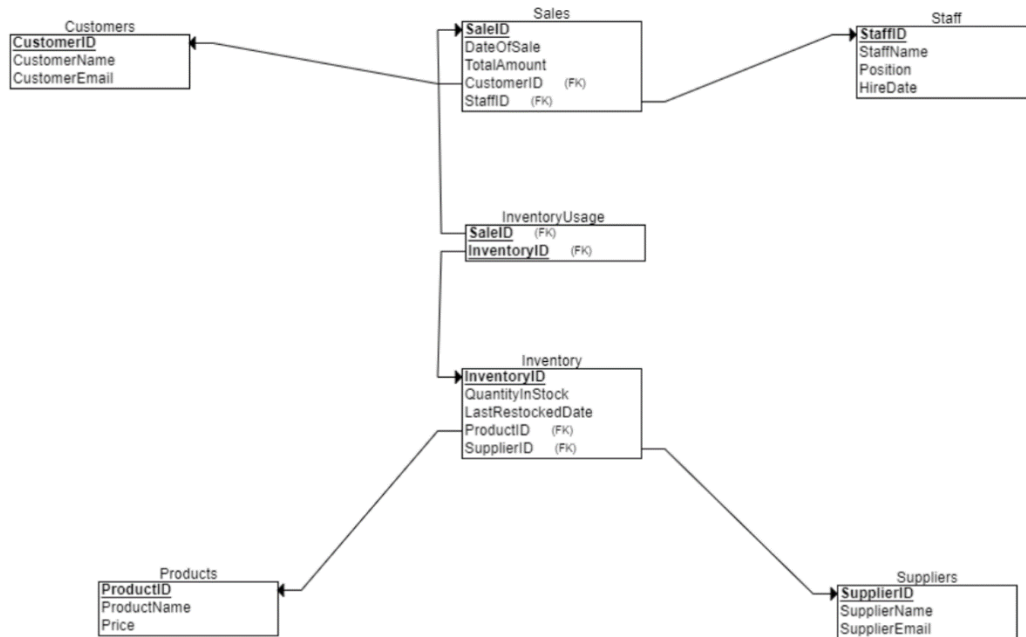
We have some relationships, such as InventoryUsage, which creates a relation between Sales and Inventory to link the items in inventory to the sales transactions.

Attributes like HireDate and LastRestockedDate are time attributes that provide a chronology of events for our project. I gave some examples, but we can see the different entities and attributes along with their relationships.

Here are the entities:

- Customers: CustomerID, CustomerName, CustomerEmail
- Products: ProductID, ProductName, Price
- Staff: StaffID, StaffName, Position, HireDate
- Suppliers: SupplierID, SupplierName, SupplierEmail
- Sales: SaleID, DateOfSale, TotalAmount
- Inventory: InventoryID, ProductID, QuantityInStock, LastRestockedDate, SupplierID





Our design depends on our purpose, and we needed to create a store, so the entities we chose are able to fulfill this purpose. Initially, we have:

- **Customers:** These attributes are essential for identifying and contacting customers, as well as managing their transactions.
- **Sales:** Tracking sales is crucial for revenue calculation and inventory management. Attributes like DateOfSale and TotalAmount provide essential information for sales analysis and financial reporting.
- **Staff:** Managing staff information is important for HR purposes and for understanding who is responsible for various sales transactions.
- **Suppliers:** Keeping track of supplier details is necessary for procurement and maintaining good supplier relationships.
- **Products:** Detailed product information is essential for inventory management, pricing strategies, and sales analysis.
- **Inventory:** Managing inventory ensures that the store can meet customer demand without overstocking. Attributes like QuantityInStock and LastRestockedDate are critical for inventory control.

SQL Output Statistics

```
SELECT * FROM Customers;
SELECT * FROM Sales;
SELECT * FROM Products;
SELECT * FROM Inventory;
SELECT * FROM Staff;
SELECT * FROM Suppliers;
SELECT * FROM InventoryUsage;
```

Select customers Select sales Select products Select inventory Select staff Select suppliers Select inventoryusage

CUSTOMERID	CUSTOMERNAME	CUSTOMEREMAIL
1	John Doe	john.doe@example.com
2	Jane Smith	jane.smith@example.com

AS SYSDBA [19:26:13] 2 rows selected in 0.019 seconds

PL/SQL Clipboard

```
CREATE TABLE Customers (
  CustomerID INT PRIMARY KEY,
  CustomerName VARCHAR2(100),
  CustomerEmail VARCHAR2(100)
);
SELECT * FROM Customers;
SELECT * FROM Sales;
SELECT * FROM Products;
SELECT * FROM Inventory;
INSERT INTO Customers (CustomerID, CustomerName, CustomerEmail) VALUES (1, 'John Doe', 'john.doe@example.com');
INSERT INTO Customers (CustomerID, CustomerName, CustomerEmail) VALUES (2, 'Jane Smith', 'jane.smith@example.com');
DROP TABLE InventoryUsage;
CREATE TABLE InventoryUsage (
  SaleID INT,
  InventoryID INT,
  Quantity INT,
  DateOfSale DATE,
  TotalAmount INT,
  ProductID INT
);
CREATE TABLE Sales (
  SaleID INT PRIMARY KEY,
  InventoryID INT,
  ProductID INT,
  QuantityInStock INT,
  InventoryID INT,
  Quantity INT
);
CREATE TABLE Inventory (
  InventoryID INT PRIMARY KEY,
  ProductID INT,
  QuantityInStock INT,
  InventoryID INT,
  Quantity INT
);
```

INVENTORYUSAGE

< Owner Table Number of records

> SYS INVENTORYUSAGE 400

Name	Type	Size	Data	Master
SALEID	NUMBER		Sequence(1)	
INVENTORYID	NUMBER		Sequence(1)	

Definition Options Result

system@BXE Generated in 0.016 seconds. 1 table(s) with 400 rows.

PL/SQL Clipboard

```
CREATE TABLE Customers (
  CustomerID INT PRIMARY KEY,
  CustomerName VARCHAR2(100),
  CustomerEmail VARCHAR2(100)
);
SELECT * FROM Customers;
SELECT * FROM Sales;
SELECT * FROM Products;
SELECT * FROM Inventory;
INSERT INTO Customers (CustomerID, CustomerName, CustomerEmail) VALUES (1, 'John Doe', 'john.doe@example.com');
INSERT INTO Customers (CustomerID, CustomerName, CustomerEmail) VALUES (2, 'Jane Smith', 'jane.smith@example.com');
DROP TABLE InventoryUsage;
CREATE TABLE InventoryUsage (
  SaleID INT,
  InventoryID INT,
  Quantity INT,
  DateOfSale DATE,
  TotalAmount INT,
  ProductID INT
);
CREATE TABLE Sales (
  SaleID INT PRIMARY KEY,
  InventoryID INT,
  ProductID INT,
  QuantityInStock INT,
  InventoryID INT,
  Quantity INT
);
CREATE TABLE Inventory (
  InventoryID INT PRIMARY KEY,
  ProductID INT,
  QuantityInStock INT,
  InventoryID INT,
  Quantity INT
);
```

