

put the following in pdf format

1. Think carefully about the business, describe a clear background story of the business.

Ryan is the proprietor of ActiveGear, a local sporting goods store that has been a staple in the community for over a decade. Recognizing the inefficiencies of the traditional paper-based filing system, Ryan has sought to modernize his operations by implementing a Relational Database Management System (RDBMS). At ActiveGear, customers can find a wide range of sporting equipment, apparel, and accessories catering to various sports and outdoor activities. Ryan aims to streamline the process of managing customer information, inventory, sales transactions, store locations, and employee data through the implementation of an RDBMS. This transition will not only enhance the efficiency of day-to-day operations but also provide valuable insights into customer preferences, inventory trends, and sales performance, allowing ActiveGear to better serve its clientele and maintain its position as a leading provider of sporting goods in the community.

2. Make sufficient assumptions, and create the ER model.

Entities:

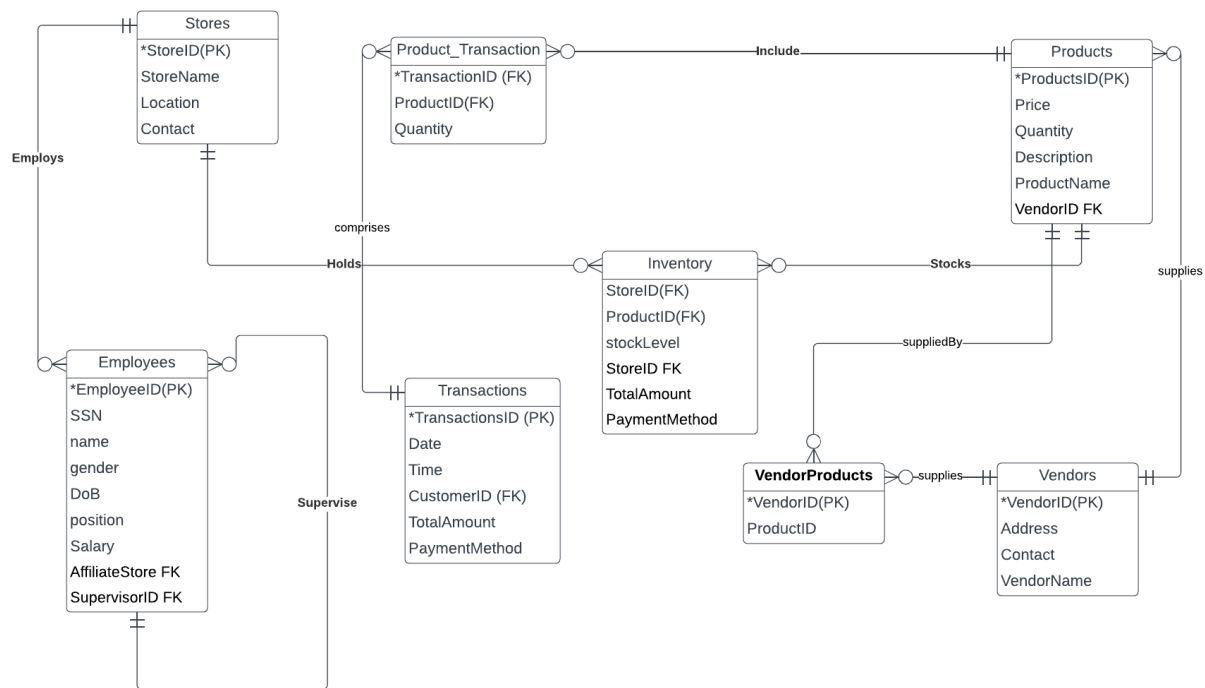
- Stores: StoreID, StoreName, Location, Contact
- Employees: EmployeeID, SSN, Name, Gender, DoB, Position, Salary, AffiliateStore, SupervisorID
- Inventory: StoreID, ProductID, StockLevel, TotalAmount, PaymentMethod
- Vendors: VendorID, Address, Contact, VendorName
- Products: ProductID, Price, Quantity, Description, ProductName, VendorID
- Transactions: TransactionsID (PK), Date, Time, CustomerID(FK), TotalAmount, PaymentMethod
- Product_Transaction: TransactionID (FK), ProductID(FK), Quantity
- ProductVendors Associative Table: ProductID, VendorID

Relationships:

- Stores may have 1 or more employees; A employee belongs to one store
- Stores may have 1 or more inventories; an inventory belongs to one store
- Vendors have 1 or more products; and products belong to one inventory
- An employee must have one and only one (exactly one) supervisor; A staff may supervise one or more (zero or more) staff.
- Products must have 1 and only one (exactly one) inventory; an inventory may stock one or more (zero or more) products.
- Products must have one and only one (exactly one) Product_transaction; A Product_transaction may include one or more (zero or more) Products
- Transactions must have one and only one (exactly one) Product_Transaction; a Product_transaction may include one or more (zero or more) Transactions.
- One-to-many from Products to ProductVendors: Each product can appear in many records in the ProductVendors table.

- One-to-many from Vendors to ProductVendors: Each vendor can appear in many records in the ProductVendors table.

3. From the ER model, create the ERD.



4. Relational Model

- Stores
 - Stores(StoreID PK, StoreName, Location, Contact)
- Employees
 - Employees(EmployeeID PK, SSN, Name, Gender, DoB, Position, Salary, AffiliateStoreID FK, SupervisorID FK)
- Transactions
 - Transactions(TransactionsID PK, Date, Time, CustomerID FK, TotalAmount, PaymentMethod, StoreID FK)
- Product_Transaction
 - Product_Transaction(TransactionID FK, ProductID FK, Quantity)
- Products
 - Products(ProductsID PK, Price, Quantity, Description, ProductName, VendorID FK)
- Inventory
 - Inventory(StoreID FK, ProductID FK, stockLevel)
- Vendors

- Vendors(VendorID PK, Address, Contact, VendorName)
- VendorProducts
 - VendorProducts(VendorID FK, ProductID FK)

5. Normalization to the Third Normal Form (3NF)

1. Stores

- Already in 3NF because all attributes are non-prime and depend only on the primary key (StoreID).

2. Employees

- Already in 3NF. Each attribute depends on the primary key (EmployeeID), and there are no transitive dependencies.

3. Transactions

- Already in 3NF. All attributes depend only on the primary key (TransactionsID), with no evidence of transitive dependencies.

4. Product_Transaction

- Already in 3NF as well. The Quantity attribute depends only on the composite primary key (TransactionID, ProductID).

5. Products

- Quantity attribute needs to be addressed. With Quantity referring to the quantity of this product available in all stores, it should be in the Inventory table, not here.
- Revised Products table:
 - Products(ProductsID PK, Price, Description, ProductName, VendorID FK)
- Updated Inventory table:
 - Inventory(StoreID FK, ProductID FK, stockLevel, Quantity)

6. Inventory

- If Quantity is moved from the Products table, then Inventory would be in 3NF.

7. Vendors

- Already in 3NF because all attributes depend only on the primary key (VendorID).

8. VendorProducts

- Already in 3NF as well since it only contains the relationship between Vendors and Products.

The possible issue with the Products table suggests I need to move Quantity to Inventory. After this change, all tables should be in 3NF.

6. Finalize the relational model in 3NF for further implementation.

- Stores
 - Stores(StoreID PK, StoreName, Location, Contact)
- Employees

- Employees(EmployeeID PK, SSN, Name, Gender, DoB, Position, Salary, AffiliateStoreID FK, SupervisorID FK)
- Transactions
 - Transactions(TransactionsID PK, Date, Time, CustomerID FK, TotalAmount, PaymentMethod, StoreID FK)
- Product_Transaction
 - Product_Transaction(TransactionID FK, ProductID FK, Quantity)
- Products
 - Products(ProductsID PK, Price, Description, ProductName, VendorID FK)
- Inventory
 - Inventory(StoreID FK, ProductID FK, stockLevel, Quantity)
- Vendors
 - Vendors(VendorID PK, Address, Contact, VendorName)
- VendorProducts
 - VendorProducts(VendorID FK, ProductID FK)

