Relationship Database Final Project

Company Background

Loren is the owner of BuzzWorks, a local honey and beekeeping supply store. For years, Loren has managed beekeeping operations with paper ledgers and spreadsheets, but as the business expanded into retail honey sales, hive equipment, beeswax products, and beekeeper workshops, the current manual system has become inefficient. Loren now wants to implement a Relational Database Management System to handle customer sales, hive management, product inventory, supplier relationships, and employee records. This new system will streamline day-to-day operations, track honey production trends, and provide insights into which products and workshops are most popular, ensuring BuzzWorks thrives as both a community hub for beekeepers and a sustainable business.

Create an Entity Relationship Model

Entities, Attributes, and Identifiers

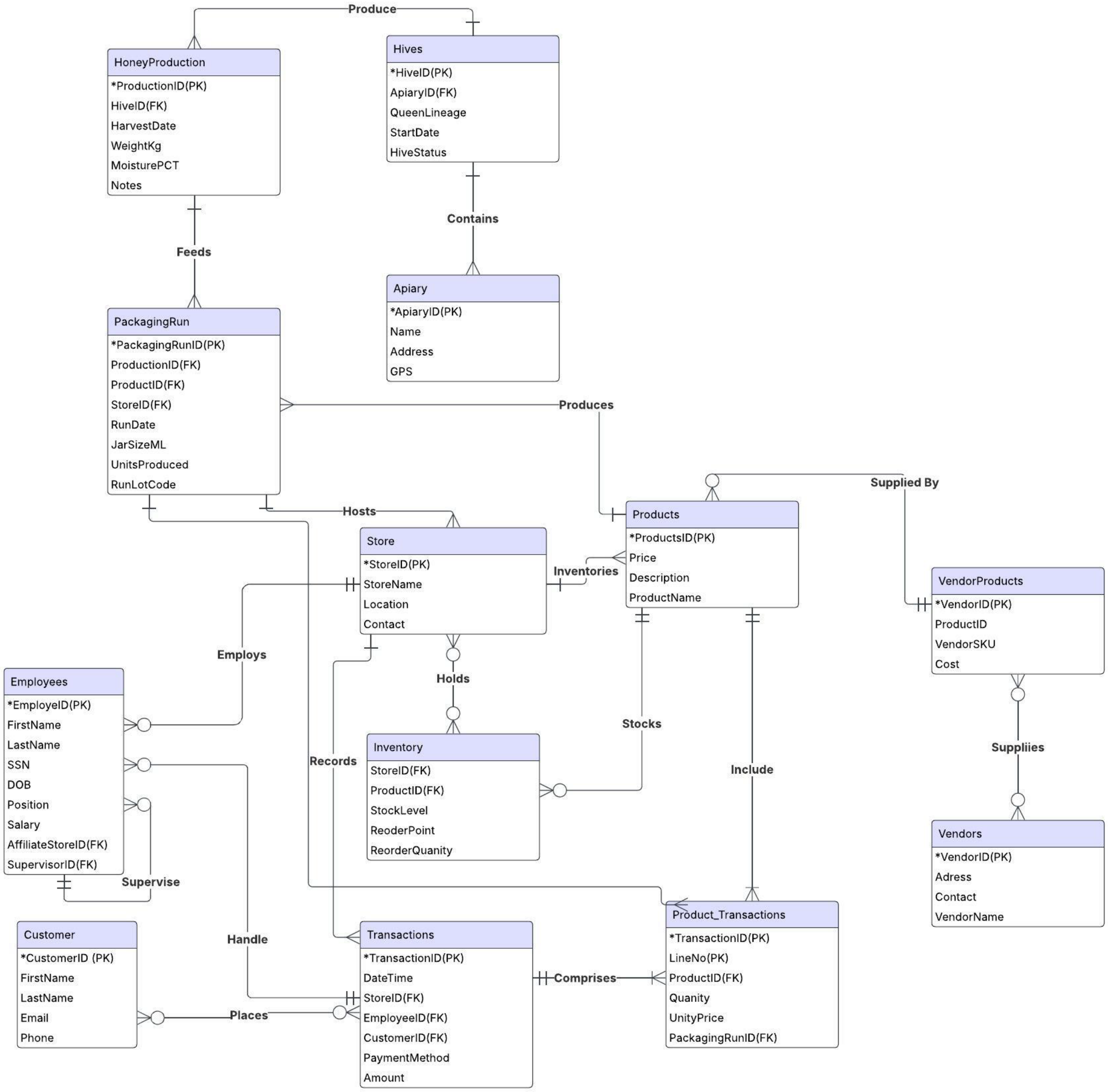
- Stores: *StoreID, Storename, Location, Contact
- Employees: *EmployeeID, Firstname, Lastname, DOB, Position, Salary, AffiliateStore, SupervisorID
- Customer: *CustomerID, Firstname, Lastname, Email, Phone
- Suppliers: *SupplierID, SupplierName, Address, Contact
- Products: *ProductID, ProductName, Description, Price
- Inventory: StoreID, ProductID, StockLevel, ReorderPOint, ReorderQuantity
- Transactions: *TransactionID, DataTime, CutomerID, EmployeeID, StoreID, TotalAmount, PaymentMethod
- Hives: *HiveID, QueenLineage, StartDate, HiveStatus
- HoneyProduction: *ProductionID, HiveID, Data, AmountKg, Notes
- Vendors: *VendorID, VendorName, Address, Contact
- VendorProducts: VendorID, ProductID, VendorSKU, Cost
- Apiary: *ApiaryID, ApiaryName, Location, Manager
- Activity Records: *ActivityID, HiveID, ActivityDate, Description, PerformedBy
- ProductTransactions: *TransactionID, LineNo, ProductID, Quantity, UnitPrice, PackagingRun
- PackagingRuns: *PackagingRunID, ProductionID, StoreID, Date, Quantity

Relationships Among Entities

- A store may employ one or more employees, and each employee must belong to one and only one store.

- An employee may have zero or one supervisor, and each employee may supervise zero or more employees.
- A store may stock zero or more products, and a product may be stocked in zero or more stores.
- A customer may place zero or more transactions, and each transaction is placed by zero or one customer (to allow walk-ins).
- A store records one or more transactions, and each transaction must be recorded at one and only one store.
- An employee handles one or more transactions, and each transaction must be handled by one and only one employee.
- A transaction must contain one or more line items, and each line item must belong to one and only one transaction.
- A product may appear on zero or more line items, and each line item must reference one and only one product.
- A vendor may supply one or more products, and a product may be supplied by one or more vendors.
- An apiary must contain one or more hives, and each hive must belong to one and only one apiary.
- A hive may have one or more activity records, and each activity record must belong to one and only one hive.
- A hive may produce one or more honey production records (harvests), and each honey production record must belong to one and only one hive.
- A honey production record (harvest) may feed one or more packaging runs, and each packaging run must derive from one and only one honey production record.
- A product (Honey 500 ml) may be created by one or more packaging runs, and each packaging run must produce one and only one product SKU.
- A store may host one or more packaging runs (where finished jars are credited to that store’s inventory), and each packaging run must be assigned to one and only one store.

Create the Entity Relationship Diagram



Convert the Entity Relationship Diagram to a Relational Model

- Stores: StoreID, Storename, Location, Contact
- Employees: EmployeeID, Firstname, Lastname, DOB, Position, Salary, AffiliateStore, SupervisorID (fk)
- Customer: CustomerID, Firstname, Lastname, Email, Phone
- Suppliers: SupplierID, SupplierName, Address, Contact
- Products: ProductID, ProductName, Description, Price
- Inventory: StoreID (fk), ProductID (fk), StockLevel, ReorderPOint, ReorderQuantity
- Transactions: TransactionID, DataTime, CutomerID (fk), EmployeeID (fk), StoreID (fk), TotalAmount, PaymentMethod
- Hives: HiveID, QueenLineage, StartDate, HiveStatus
- HoneyProduction: ProductionID, HiveID (fk), Data, AmountKg, Notes
- Vendors: VendorID, VendorName, Address, Contact
- VendorProducts: VendorID (fk), ProductID (fk), VendorSKU, Cost
- Apiary: ApiaryID, ApiaryName, Location, Manager
- Activity Records: ActivityID, HiveID (fk), ActivityDate, Description, PerformedBy
- ProductTransactions: TransactionID, LineNo (fk), ProductID (fk), Quantity, UnitPrice, PackagingRun (fk)
- PackagingRuns: PackagingRunID, ProductionID (fk), StoreID (fk), Date, Quantity

Normalize the Relational Model to 3NF

Functional Dependencies

Before normalization, check for Functional Dependencies of the relations.

- Stores (StoreID, Storename, Location, Contact)
  - FD1: StoreID → Storename, Location, Contact
- Employees (EmployeeID, Firstname, Lastname, DOB, Position, Salary, AffiliateStore, SupervisorID)
  - FD1: EmployeeID → Firstname, Lastname, DOB, Position, Salary, AffiliateStore, SupervisorID
- Customer (CustomerID, Firstname, Lastname, Email, Phone)
  - FD1: CustomerID → Firstname, Lastname, Email, Phone
- Suppliers (SupplierID, SupplierName, Address, Contact)
  - FD1: SupplierID → SupplierName, Address, Contact
- Products (ProductID, ProductName, Description, Price)
  - FD1: ProductID → ProductName, Description, Price
- Inventory (StoreID, ProductID, StockLevel, ReorderPoint, ReorderQuantity)
  - FD1: StoreID, ProductID → StockLevel, ReorderPoint, ReorderQuantity

- Transactions (TransactionID, DataTime, CustomerID, EmployeeID, StoreID, TotalAmount, PaymentMethod)
    - FD1: TransactionID → DataTime, CustomerID, EmployeeID, StoreID, TotalAmount, PaymentMethod
- Hives (HiveID, QueenLineage, StartDate, HiveStatus)
    - FD1: HiveID → QueenLineage, StartDate, HiveStatus
- HoneyProduction (ProductionID, HiveID, Data, AmountKg, Notes)
    - FD1: ProductionID → HiveID, Data, AmountKg, Notes
- Vendors (VendorID, VendorName, Address, Contact)
    - FD1: VendorID → VendorName, Address, Contact
- VendorProducts (VendorID, ProductID, VendorSKU, Cost)
    - FD1: VendorID, ProductID → VendorSKU, Cost
- Apiary (ApiaryID, ApiaryName, Location, Manager)
    - FD1: ApiaryID → ApiaryName, Location, Manager
- Activity Records (ActivityID, HiveID, ActivityDate, Description, PerformedBy)
    - FD1: ActivityID → HiveID, ActivityDate, Description, PerformedBy
- ProductTransactions (*TransactionID, LineNo, ProductID, Quantity, PackagingRun)
    - FD1: TransactionID, LineNo → ProductID, Quantity, PackagingRun
- Packaging Runs (PackagingRunID, ProductionID, StoreID, Date, Quantity)
    - FD1: PackagingRunID → ProductionID, StoreID, Date, Quantity

<u>Normalize to 3NF</u>

- Stores, Employees, Customer, Suppliers, Products, Hives, HoneyProduction, Vendors, Apiary, Activity Records, Transactions, Inventory, VendorProducts, and ProductTransactions relations are in 3NF, because they are in 1NF; they have no partial functional dependencies so they are in 2NF; and they have no transitive functional dependencies so they are in 3NF.
    - Stores (*StoreID, Storename, Location, Contact) with FD1: StoreID → Storename, Location, Contact.

- Packaging Runs relation is in 1NF. However, it is not in 2NF because it has attributes that might suggest partial dependencies, but assuming FD1: PackagingRunID → ProductionID, StoreID, Date, Quantity holds with a single primary key, it is in 2NF. It's also in 3NF because it is in 1NF and 2NF, and there are no transitive functional dependencies unless specified.
    - Packaging Runs (*PackagingRunID, ProductionID, StoreID, Date, Quantity)
        - FD1: PackagingRunID → ProductionID, StoreID, Date, Quantity.

- Packaging Runs relation is in 1NF and 2NF. However, it is not in 3NF because of a potential transitive functional dependency. Assuming FD1: PackagingRunID → ProductionID, StoreID, Date, Quantity and an inferred FD2: ProductionID → StoreID (e.g., store assignment based on production location), PackagingRunID → ProductionID and ProductionID → StoreID form a transitive functional dependency. We need to normalize Packaging Runs to 3NF:

    - Create a new relation to put ProductionID, StoreID and modify Packaging Runs. Now, ProductionStoreLink and Packaging Runs relations are:
        - ProductionStoreLink (*ProductionID, StoreID)
            - FD1: ProductionID → StoreID
        - Packaging Runs (*PackagingRunID, ProductionID, Date, Quantity)
            - FD1: PackagingRunID → ProductionID, Date, Quantity.

Finalize the Relational Model in 3NF

- Customer (CustomerID, FirstName, LastName, Email, Phone, PaymentInfo, ReferredByCustomerID(fk))
    - FD1: CustomerID → FirstName, LastName, Email, Phone, PaymentInfo, ReferredByCustomerID
- Stores (StoreID, Storename, Location, Contact)
    - FD1: StoreID → Storename, Location, Contact
- Employees (EmployeeID, Firstname, Lastname, DOB, Position, Salary, AffiliateStore(fk), SupervisorID(fk))
    - FD1: EmployeeID → Firstname, Lastname, DOB, Position, Salary, AffiliateStore, SupervisorID
- Products (ProductID, ProductName, Description, Price)
    - FD1: ProductID → ProductName, Description, Price
- Inventory (StoreID(fk), ProductID(fk), StockLevel, ReorderPoint, ReorderQuantity)
    - FD1: StoreID, ProductID → StockLevel, ReorderPoint, ReorderQuantity
- Transactions (TransactionID, DataTime, CustomerID(fk), EmployeeID(fk), StoreID(fk), TotalAmount, PaymentMethod)
    - FD1: TransactionID → DataTime, CustomerID, EmployeeID, StoreID, TotalAmount, PaymentMethod
- Hives (HiveID, QueenLineage, StartDate, HiveStatus)
    - FD1: HiveID → QueenLineage, StartDate, HiveStatus
- HoneyProduction (ProductionID, HiveID(fk), Data, AmountKg, Notes)
    - FD1: ProductionID → HiveID, Data, AmountKg, Notes
- Vendors (VendorID, VendorName, Address, Contact)
    - FD1: VendorID → VendorName, Address, Contact
- Suppliers (SupplierID, SupplierName, Address, Contact)
    - FD1: SupplierID → SupplierName, Address, Contact
- VendorProducts (VendorID(fk), ProductID(fk), VendorSKU, Cost)
    - FD1: VendorID, ProductID → VendorSKU, Cost
- Apiary (ApiaryID, ApiaryName, Location, Manager)
    - FD1: ApiaryID → ApiaryName, Location, Manager
- Activity Records (ActivityID, HiveID(fk), ActivityDate, Description, PerformedBy)
    - FD1: ActivityID → HiveID, ActivityDate, Description, PerformedBy

- ProductTransactions (*TransactionID(fk), LineNo, ProductID(fk), Quantity, PackagingRun(fk))
    - FD1: TransactionID, LineNo → ProductID, Quantity, PackagingRun
- Packaging Runs (PackagingRunID, ProductionID(fk), Date, Quantity)
    - FD1: PackagingRunID → ProductionID, Date, Quantity
- ProductionStoreLink (ProductionID(fk), StoreID(fk))

There is no non-primary-key attribute