**Problem Description**

We were given a company and were tasked to create a database that represents the employees and customers that interact within the company in different aspects, such as hiring or selling the product.

**Individual Contribution**

Kenneth:

We split the work up evenly having each person do certain parts of the project and check on each other to stay consistent. I focused mainly on the EER diagram, third normal schema and dependency chart. Alongside those I also created the CREATE sql code for the database and half the views and half the queries. In the documentation, I wrote out the third normal.

Eric:

My participation in the project was through mapping the relational schema in the database schema format and charting the cardinalities of the EER diagram. Alongside those, I created the assumptions attached to the EER Diagram, and coding wise, I wrote the second half of the views and the second half of the queries. For the documentation work, I wrote out the normalization to 3NF and the assumptions as seen below. After completing normalization, revised the dependency diagram to reflect the normalized table as well as the views.

EER Diagram w/ Assumptions

EER Diagram will be attached as a png.

Assumptions:

A person can be classified as an employee, potential employee, or customer. A person must belong to at least one of these specifications and may belong to more than one. Any employee applying for a job, both existing and new, will belong to the specification “potential employee” along with any existing specifications. Each employee receives exactly one salary, regardless of the number of departments they are employed by. Salaries cannot exist without being tied to an employee. Departments can post multiple job openings. Applicants can apply for multiple job openings. Each interview is conducted by exactly one employee (who is also solely responsible for grading the interview) and is attended by exactly one potential employee. Each product can use multiple parts and each part can be used by several different products. The same vendor can produce multiple different parts and different may produce the same part. Each sale recorded is specific to one marketing site. One sale can consist of several parts. Sales are made to one customer and accredited to one employee.

Normalizing to 3NF

To normalize to 3NF, attributes must satisfy certain constraints. They are, in order

1. All attributes must be single, atomic values. This means that there cannot exist multi-valued attributes.
2. All attributes in a table must solely depend on the primary key. This disallows attributes to partially dependant on a primary composite key or depend on more than just the primary key.
3. All attributes must depend on the primary key and not any other attributes.

**1. Person**

**Initial Schema:**

Person

- PersonalID (PK)

- LastName

- FirstName

- Age

- PhoneNumber

- Gender

- Address

- City

- State

- ZipCode

**Issues:** 1NF violation: Phone Number is not an atomic value because it is a multi-valued attribute.

Normalized Schema:

1. Person:

PersonalID (PK), LastName, FirstName, Age, PhoneNumber, Gender, Address, City, State, ZipCode.

2. Person\_PhoneNumber:

PersonalID (FK, PK), PhoneNumber(PK)

2NF met (no partial dependencies)

3NF violation: City, State, and Zipcode are dependant on Address, and not on PersonalID (PK). To ensure that all attributes are wholly dependant on the PK, a new Table for Address must be made to address the transitive dependency PersonalID->Address->(City, State, Zipcode). Instead of decomposing the table, PersonalID, Address, City, State, and ZipCode can be combined to form one composite key. Though there are drawbacks to this method like update and query complexity, it is unlikely that using this composite key in this DBMS will cause redundancies as attributes in Person\_Address are not referenced in any other relation.

Normalized Schema:

1. Person:

PersonalID (PK), LastName, FirstName, Age, PhoneNumber, Gender

2. Person\_Address:

PersonalID (FK, PK), Address(PK), City(PK), State(PK), ZipCode(PK)

3. Person\_PhoneNumber:

PersonalID (FK, PK), PhoneNumber(PK)

**2. Employee (Already in 3NF)**

Initial Schema:

Employee

- EmployeeID (PK, FK)

- PersonalID (FK)

- Rank

- Title

- SupervisorID (FK, references EmployeeID)

Normalized Schema:

Employee

- EmployeeID (PK, FK), PersonalID (FK), Rank, Title, SupervisorID (FK)

**3. Customer (Already in 3NF)**

Initial Schema:

Customer

- CustomerID (PK)

- PersonalID (FK)

- PreferredSalespersonID (FK, references EmployeeID)

Normalized Schema:

Customer

- CustomerID (PK), PersonalID (FK), PreferredSalespersonID (FK)

**4. Potential\_Employee (Already in 3NF)**

Initial Schema:

Potential\_Employee

- PotentialEmployeeID (PK, FK)

- PersonalID (FK)

Normalized Schema:

Potential\_Employee

- PotentialEmployeeID (PK, FK), PersonalID (FK)

**5. Department (Already in 3NF)**

Initial Schema:

Department

- DepartmentID (PK)

- DepartmentName

Normalized Schema:

Department

- DepartmentID (PK), DepartmentName

**6. Salary (Already in 3NF)**

Initial Schema:

Salary

- TransactionNumber (PK)

- EmployeeID (PK, FK)

- PayDate

- Amount

Normalized Schema:

Salary

- TransactionNumber (PK), EmployeeID (PK, FK), PayDate, Amount

**7. Job\_Position (Already in 3NF)**

Initial Schema:

Job\_Position

- JobID (PK)

- DepartmentID (FK)

- JobDescription

- PostedDate

Normalized Schema:

Job\_Position

- JobID (PK), DepartmentID (FK), JobDescription, PostedDate

**8. Product (Already in 3NF)**

Initial Schema:

Product

- ProductID (PK)

- ProductType

- Size

- ListPrice

- Weight

- Style

Normalized Schema:

Product

- ProductID (PK), ProductType, Size, ListPrice, Weight, Style

**9. Part (Already in 3NF)**

Initial Schema:

Part

- PartID (PK)

- PartName

- Price

- Weight

**Issues:** 1NF constraint met as there are no partial dependencies. 2NF violation: Attribute not wholly dependent on PK. Price is dependent on PartID as well as VendorID. Price should instead be included in the Supplies M:N relation table, which contains a composite key of partID and VendorID (see Table 11).

Normalized Schema:

Part

- PartID (PK), PartName, Weight

There are no transitive dependencies.

**10. Vendor**

Initial Schema:

Vendor

- VendorID (PK)

- Name

- AddressLine

- City

- State

- ZipCode

- AccountNumber

- CreditRating

- PurchasingWebServiceURL

**Issues:** 1NF and 2NF constraints are met because attributes are atomic and there are no partial dependencies. 3NF violation: City, State, and Zipcode are dependent on AddressLine, and not on VendorID (PK). To ensure that all attributes are wholly dependent on the PK, a new Table for Address must be made to address transitive dependency. VendorID->AddressLine->(City, State, Zipcode). Instead of decomposing the table, PersonalID, Address, City, State, and ZipCode can be combined to form one composite key.

Normalized Schema:

Vendor

- VendorID (PK), Name, AccountNumber, CreditRating, PurchasingWebServiceURL

Vendor\_Address

- VendorID (FK, PK), AddressLine(PK), City(PK), State(PK), ZipCode(PK)

**11. Supplies (Already in 3NF)**

Initial Schema:

Supplies

- VendorID (FK)

- PartID (FK)

- Price

Normalized Schema:

Supplies

- VendorID (FK, PK), PartID (FK, PK), Price

12. Consists\_of (Already in 3NF)

Initial Schema:

Consists\_of

- ProductID (FK)

- PartID (FK)

Normalized Schema:

Uses

- ProductID (FK, PK), PartID (FK, PK)

**13. Marketing\_Site (Already in 3NF)**

Initial Schema:

Marketing\_Site

- SiteID (PK)

- SiteName

- Location

Normalized Schema:

Marketing\_Site

- SiteID (PK), SiteName, Location

**14. Works\_At (Already in 3NF)**

Initial Schema:

Works\_At

- EmployeeID (FK)

- SiteID (FK)

Normalized Schema:

Works\_At

- EmployeeID (FK, PK), SiteID (FK, PK)

**15. Works\_For (Already in 3NF)**

Initial Schema:

Works\_For

- EmployeeID (FK)

- DepartmentID (FK)

- Shift\_Start

- Shift\_End

Issues: The inclusion of a flag is necessary to indicate which singular department an employee is currently working for. Include attribute CurrentDept with a domain of [0,1] that is dependent on the composite key composed of EmployeeID and DepartmentID to satisfy 2NF.

Normalized Schema:

Works\_For

- EmployeeID (FK, PK), DepartmentID (FK, PK), Shift\_Start, Shift\_End, CurrentDept

**16. Applies (Already in 3NF)**

Initial Schema:

Applies

- PotentialEmployeeID (FK)

- JobID (FK)

- ApplicationDate

Normalized Schema:

Applies

- PotentialEmployeeID (FK, PK), JobID (FK, PK), ApplicationDate

**17. Interview**

Initial Schema:

Interview

- InterviewID (PK)

- JobID (FK)

- IntervieweeID (FK, references potentialEmployeeID)

- InterviewerID (FK, references EmployeeID)

- InterviewTime

- Grade

- Status

**Issues:** 1 NF constraint met as there are not non-atomic attributes. 2NF violation: Partial dependencies in table: Grade and Status depends on the InterviewID, IntervieweeID, and InterviewerID because a grade is given in an interview to a candidate assigned by an employee. This means that Grade and Status do not wholly depend on InterviewID. To resolve this, an option is to create a composite key from the above mentioned attributes. However, this causes partial dependency issues relating to InterviewTime because InterviewTime is only dependent on InterviewID and not the composite key. Therefore, the table must be decomposed. (Note: JobID can be considered dependant on InterviewID because each interview is associated with exactly one job)

Normalized Schema:

Interview

InterviewID (PK), JobID (FK), InterviewTime

Interview\_Grade

InterviewID (PK), InterviewerID (FK, PK), IntervieweeID (FK, PK), Grade, Status

3NF violation: Status is dependent on Grade and not on the composite key. This creates a transitive dependency [InterviewID (PK), InterviewerID (FK, PK), IntervieweeID (FK, PK)]->Grade->Status. An additional table must be made to record an applicants pass/fail status where Status will depend on a composite key of InterviewID and IntervieweeID.

Normalized Schema:

Interview

InterviewID (PK), JobID (PK), InterviewTime

Interview\_Grade

InterviewID (FK, PK), InterviewerID (FK, PK), IntervieweeID (FK, PK), Grade

Interview\_Status

InterviewID (FK,PK), IntervieweeID (FK, PK), Status

**18. Sale**

Initial Schema:

Sale

- SaleID (PK)

- ProductID (FK)

- CustomerID (FK)

- EmployeeID (FK)

- SiteID (FK)

- SaleTime

- Quantity

**Issues:** 1NF constraint met as all attributes are atomic values. 2NF violation: There are partial dependencies in the table: Quantity is dependant on SaleID and ProductID because one each sale can consist of different quantities of different products. However, SaleTime is only dependent on SaleID. (Note: CustomerID, EmployeeID, and SiteID can be considered dependent on SaleID because each sale is associated with exactly one customer and one employee at one site).

Normalized Schema:

1. Sale:

SaleID (PK), CustomerID (FK), EmployeeID (FK), SiteID (FK), SaleTime

2. Product\_Details:

SaleID (FK, PK), ProductID (FK, PK), Quantity

No transitive dependencies.

**19. Sold\_in**

Initial Schema:

Sold\_in

- SaleID (FK)

- ProductID (FK)

Issues: Redundancy: after normalizing Sale to 3NF, the table Product\_details is created. Both tables map M:N relations, and Product\_details is a superset of Sold\_in. Therefore, this table can be eliminated.

Normalized Schema:

Product\_Details

SaleID (FK,PK), ProductID (FK,PK), Quantity

**Final List of Normalized Tables:**

**1. Person**

Person

- PersonalID (PK)

- LastName

- FirstName

- Age

- Gender

**2. Person\_Address**

Person\_Address

- PersonalID (FK, PK)

- Address (PK)

- City (PK)

- State (PK)

- ZipCode (PK)

**3. Person\_PhoneNumber**

Person\_PhoneNumber

- PersonalID (FK, PK)

- PhoneNumber (PK)

**4. Employee**

Employee

- EmployeeID (PK, FK)

- PersonalID (FK)

- Rank

- Title

- SupervisorID (FK, references EmployeeID)

**5. Customer**

Customer

- CustomerID (PK)

- PersonalID (FK)

- PreferredSalespersonID (FK, references EmployeeID)

**6. Potential\_Employee**

Potential\_Employee

- PotentialEmployeeID (PK, FK)

- PersonalID (FK)

**7. Department**

Department

- DepartmentID (PK)

- DepartmentName

**8. Salary**

Salary

- TransactionNumber (PK)

- EmployeeID (FK)

- PayDate

- Amount

**9. Job\_Position**

Job\_Position

- JobID (PK)

- DepartmentID (FK)

- JobDescription

- PostedDate

**10. Product**

Product

- ProductID (PK)

- ProductType

- Size

- ListPrice

- Weight

- Style

**11. Part**

Part

- PartID (PK)

- PartName

- Weight

**12. Supplies**

Supplies

- VendorID (FK, PK)

- PartID (FK, PK)

- Price

**13. Vendor**

Vendor

- VendorID (PK)

- Name

- AccountNumber

- CreditRating

- PurchasingWebServiceURL

**14. Vendor\_Address**

Vendor\_Address

- VendorID (FK, PK)

- AddressLine (PK)

- City (PK)

- State (PK)

- ZipCode (PK)

**15. Consists\_of**

Uses

- ProductID (FK, PK)

- PartID (FK, PK)

**16. Marketing\_Site**

Marketing\_Site

- SiteID (PK)

- SiteName

- Location

**17. Works\_At**

Works\_At

- EmployeeID (FK, PK)

- SiteID (FK, PK)

**18. Works\_For**

Works\_For

- EmployeeID (FK, PK)

- DepartmentID (FK, PK)

- Shift\_Start

- Shift\_End

- CurrentDept (Domain: [0,1])

**19. Applies**

Applies

- PotentialEmployeeID (FK, PK)

- JobID (FK, PK)

- ApplicationDate

**20. Interview**

Interview

- InterviewID (PK)

- JobID (FK)

- InterviewTime

**21. Interview\_Grade**

Interview\_Grade

- InterviewID (FK, PK)

- IntervieweeID (FK, PK, references Potential\_Employee)

- InterviewerID (FK, PK, references Employee)

- Grade

**22. Interview\_Status**

Interview\_Status

- InterviewID (FK, PK)

- IntervieweeID (FK, PK, references Potential\_Employee)

- Status

**23. Sale**

Sale

- SaleID (PK)

- CustomerID (FK)

- EmployeeID (FK)

- SiteID (FK)

- SaleTime

**24. Product\_Details**

Product\_Details

- SaleID (FK, PK)

- ProductID (FK, PK)

- Quantity