Sneakers+

**Task:​** You have been contracted to develop a database system for a small company. This company makes several products that it sells to customers.

**Part 3 Final**

**Submitted to:**

Professor Rachida Parks; CIS-351,02

**Prepared by:**

Joseph Zarzano, Matthew Donahue, David Kolesar, Luke MacMillan,

Alex Feuilly

November 20, 2020

Table of Contents:

Part 3: Requirements and Submission Information…………………………… pg.4-7

ERD…………………………………………………………………………… Pg.8

**Database Logical and Physical Design Part 2:**

Vendor.……………………………………………………………………….. pg.9

Employee…………………………………………………………………….. pg.10

Call………………………………………………………………………….... pg.11

Sale…………………………………………………………………………… pg.12

Customer……………………………………………………………………… pg.13-14 Product……………………………………………………………………….. pg.15

Equipment……………………………………………………………………. pg.16

Sold Product…………………………………………………………………. Pg.17-18

Product Call…………………………………………………………………. pg.19

Order Product………………………………………………………………… pg.20

Order…………………………………………………………………………. pg.21

Departments………………………………………………………………….. Pg.22

**Database Implementation (Table):**

Vendor.……………………………………………………………………….. pg.24

Employee…………………………………………………………………….. pg.25

Call………………………………………………………………………….... pg.26 Sale…………………………………………………………………………… pg.27

Customer……………………………………………………………………… pg.28

Product……………………………………………………………………….. pg.29

Equipment……………………………………………………………………. pg.30

Sold Product…………………………………………………………………. pg.31

Product Call…………………………………………………………………. pg.32

Order Product………………………………………………………………… pg.33

Order…………………………………………………………………………. pg.34 Departments………………………………………………………………….. Pg.35

Queries:

1……..……………………………………………………………………….. Pg.36

2……..……………………………………………………………………….. Pg.37

3……..……………………………………………………………………….. Pg.38

4……..……………………………………………………………………….. Pg.39

5……..……………………………………………………………………….. Pg.40

6……..……………………………………………………………………….. pg.41

7……..……………………………………………………………………….. pg.42

8……..……………………………………………………………………….. Pg.43

9……..……………………………………………………………………….. Pg.44

10……..……………………………………………………………………… pg.45

**Part 3: Database Implementation and SQL Programming**

**Part 3 Requirements**

1. **Create Tables**

Using your corrected data structures from Project Part 2, create all the tables in your team database.

1. **Populate Tables**

Once the table structure is finalized, you must add data to the table with at least five records in each table. Some tables may require more data. Make-up your own data to populate the tables based on your company information. Data should be selected to demonstrate the required outputs outlined in the queries.

There are two ways to add data to a table.

· Right click on the table and select Open Table from the menu. The table will open in a new pane with data displayed in row/column format. Add new data to next available row.

· Execute an INSERT SQL statement for each row.

1. **Design and Run Queries**
2. List all fields from table Employee. Sort by employee last name
3. List all products provided by vendor *XXX.* ​ ​(*XXX*​ ​ correspond to the name of one of your vendors).
4. List all sale. For each sale, list the products sold (name and description of product), quantity, the employee name who handled the transaction and the customer who made the purchase. The date of the sale should also be retrieved. List all purchases made prior to today. (you may have to use many joins)
5. List all different department and the count of employees assigned to each department.
6. Insert department named ‘R&D’ to the department table
7. Delete R&D from the department Table.
8. List all the average price for your products.
9. Update the cost of one of your product from $$ to $$$.
10. Create a query using the following requirements:
    1. Joining two tables
    2. Using an outer join (right or left)

10.​ Create a query using the following requirements:​

* 1. Joining two tables or more
  2. Using an aggregate functions (SUM, AVG, MAX, MIN, COUNT)
  3. Using a filter

**Part 3 Submission**

**(Submission Requirements for Project Part 3)**

**This is a comprehensive submission and includes all the previous parts to which you add part 3.**

# Project Part 1:​ Database Logical Design​

In this part, include **all**​ the material you submitted in part 1. This include the original ERD, a​ copy of the ERD with Dr. Parks’ note and the latest ERD.

# Project Part 2:Database Logical and Physical Design

In this part, include **all**​ the material you submitted in part 2 including corrections suggested by​ Dr. Parks (from the 2 page grading sheet).

# Project Part 3: Database Implementation

This section includes the following:

1. **Tables**

For each of the tables you created, use the following form to list the screenshots of table structure and data.

|  |
| --- |
| (​**TABLE NAME**​ entered in all capital letters)  ***Example***​​ ​**TABLE:** ​EMPLOYEE |
| (**Table Structure Screenshot**​ captured in SQL Server)​    ***How-to:***  In the ​**Object Explorer**​ (the tree view on the left panel), navigate to the table (e.g. Employee) in your team database, pop up all the Columns and Keys, and make a screenshot.  **Note: Make sure all the columns and Keys are shown up in the screenshot.**    ***Example:*** |
| (**Table Data Screenshot**​ captured in SQL Server)​    ***How-to:***  Open a new query window, create a SQL statement to select all the records from the table (e.g. Employee) in your team database, and execute. The results should be shown up. Make a screenshot of all the results.    ***Example:*** |

1. **Queries**

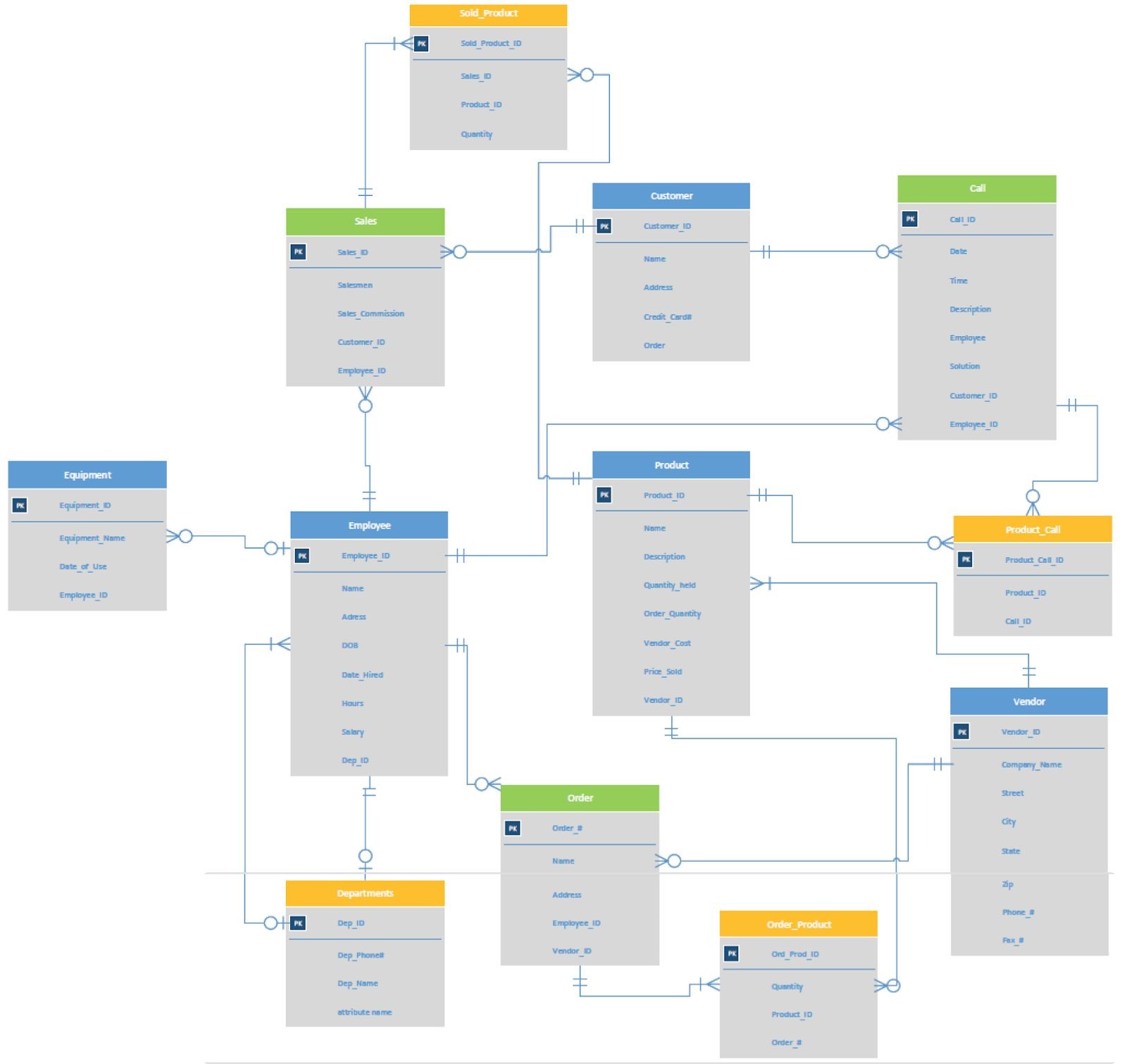
For each of the queries, use the following form to list the query, SQL statement, and results.

|  |
| --- |
| (​**Query**​)  ***Example***  List name, job title, office and phone for all ​**employees** ​whose salary is above 4000 |
| (​**SQL Statement**​)  ***Example***  SELECT LastName, FirstName, JobTitle, Office, Phone  FROM Employee  WHERE Salary > 4000 |
| (​**Results**​)  ***Example*** |

**Check Submission Format document**

* **for submission details**
* **grading rubric**

ERD



VENDOR​

· A Vendor can produce one Item or many Items, but an Item can be produced by one and only one Vendor.

· A Vendor can place zero or many Orders, but an Order is associated with one and only one Vendor.

· Vendor ID is defined as the primary key for the Vendor table.

· Only one phone number will be kept for each Vendor. If additional phone numbers are needed, additional fields must be added to the table or an additional table created to contain phone numbers.

· Only one fax number will be kept for each Vendor.

· If an Item’s inventory level falls below the reorder point, the item is noted for reorder from the vendor that supplies it. (All items have only a single source.)

· A supply order is placed with each vendor periodically (once a week for most vendors).

**Data Structure:**

VENDOR (Vendor\_ID​ (PK), Company\_Name, Street, City, State, Zip, Phone\_#, Fax\_#)​

**Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Vendor ID | Number | Required, PK, Unique sequential 4-digit ID number, Indexed |
| Company Name | Text(40) | Required, Name of the Company acting as the vendor |
| Street | Text(40) | Required, must be valid street address |
| City | Text(20) | Required, must be valid U.S. city |
| State | Text(2) | Required, must be in the format: XX, Must be a valid US state abbreviation |
| Zip | Number | Required, 5-digits, must be valid U.S. zip code |
| Phone Number | Number | Required, 10-digits, must be valid phone number |
| Fax Number | Number | Required, 10-digits, must be valid fax number |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 25 records | 5 updates per day, 10 queries per day | · Security because it contains personal information about Vendor  · Applications: New Vendor(C, R), Marketing(R),  Vendor directory Update(U, R), Management  Update(C, R, U, D) |

EMPLOYEE​

· An employee can make zero or many sales, but a sale can be made by one and only one employee

· An employee can have zero or many pieces of equipment, but a piece of equipment can have zero or one employee owners

· An employee can have 0 or 1 departments that they work in, but a department will have 1 or many employees working in it

· An employee can take 0 or many orders, but an order must be taken by 1 and only 1 employee

· An can have 0 or many calls, but a call must be taken by 1 and only 1 employee

· Employee ID is labeled as the Primary Key for Employee

· Each Employee ID must be unique

**Data Structure:**

EMPLOYEE (Employee ID​ (PK), Name, Address, DoB, Date\_Hired, Hours, Salary, Dep\_ID(fk))​ **Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Employee\_ID | Number | Required, PK, Unique sequential four-digit ID number, Indexed |
| Name | Text(40) | Required, must be the first and last name of the Employee |
| Address | Text(50) | Required, must be valid street address, city (XX format), state (XX format), and zip code |
| Date of Birth | Date/Time | Required, must be a valid birth date and of legal working age in  MM/DD/YYYY form |
| Date Hired | Date/Time | Required, must read the date of when the employee was hired in  MM/DD/YYYY form |
| Hours | Date/Time | Required, hours worked in given days and weeks in MM/DD/YYYY form |
| Salary | Number | Required, 10-digits, must be valid phone number |
| Department ID | Number | Required, fk, Unique sequential four-digit ID number, Indexed |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 20 records | 1 update per day, 20 queries per day | · Security because it contains personal  information about the Employee  · Applications: New Employee(C, R), Employee directory Update(U, R, D), Employee Management Update(C, R, U, D) Department Update (C,R,U,D)  Salary Update (C,R,U,D) Hours Update (C,R,U,D) |

Call​

· A Customer can produce no Calls or many Calls, but a Call can be produced by one and only one Customer. · An Employee can produce no Calls or many Calls, but a Call can be produced by one and only one Employee · A Product can have zero or many Calls , but a Call can be about 0 or many Products.

· Caller ID is defined as the primary key for the Vendor table.

· Only one phone number will be kept for each C. If additional phone numbers are needed, additional fields must be added to the table or an additional table created to contain phone numbers.

**Data Structure:**

CALL (Call ID​ (PK), Date, Time, Description, Employee, Solution, Customer\_ID (fk), Employee\_ID(fk))​ **Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Call ID | Number | Required, PK, Unique sequential two-digit ID number, Indexed |
| Date/Time | Date/Time | Required, description of the call specifics, MM/DD/YYYY form |
| Description | Text(100) | Required, explanation must be descriptive |
| Employee | Text(12) | Required, must be valid U.S. city |
| Solution | Text(40) | Required, must be in the format: XX, Must be a valid US state abbreviation |
| Customer ID | Number | Required, fk, 6 digit ID number that represents the customer |
| Employee ID | Number | Required, fk, Unique, references Employee ID in the Employee table, sequential five-digit, must be in format: ##### |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 30 records per day | 30 updates per day, 60 queries per day | · Security because it contains personal information of callers  · Applications: New Call (C,R), Call Directory  (R,U,D), Call Info (C,R,U), Customer Info (C,R,U),  Employee Info (C,R,U) |

Sale​

* A Sale can be made by one and only one Customer, but a Customer can have none or many sales
* A Sale can be made by one and only one Employee, but an Employee can products none or many sales
* A Sale can be associated with one or many Sold\_Product, but Sold\_Product can be one and only one to associate with sale
* Sale\_ID is the Primary Key for the Sale Table
* Both Customer\_ID and Employee\_ID are present foreign keys in Sale Table
* Salesman and ones commission will be kept in only the Sales Table information

·

**Data Structure:**

SALE (Sale\_ID​ (​ PK), Salesman, Sales\_commission, Customer\_ID (fk), Employee\_ID (fk)) **Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Sale\_ID | Number | Required, PK, 7 digit ID number that represents the sale made |
| Salesman | Text (40) | Required, references the salesman who made the sale, represented by name and employee ID |
| Sales\_Commission | Number | Required, references the percentage of commission an employee gets from a sale, written in % form |
| Customer\_ID | Number | Required, fk, 6 digit ID number that represents the customer |
| Employee\_ID | Number | Required, fk, Unique, references Employee ID in the Employee table, sequential five-digit, must be in format: ##### |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 65 records per day | 65 updates per day, 40 queries per day | · Security to protect the purchasing information of the buyer (such as credit or debit card)  · Applications: New Sale (C,R,U), Return (U,D), Customer Info (C,R,U,D), Salesman Info (C,R,U,D)  Salesman’s Commission (C,R,U) |

Customer​

· A Customer can produce no Calls or many Calls, but a Call can be produced by one and only one Customer. · A Customer can produce no Sales or many Sales, but a Sale can be produced by one and only one Customer · Customer ID is defined as the primary key for the Customer table.

· Only one phone number will be kept for each Customer. If additional phone numbers are needed, additional fields must be added to the table or an additional table created to contain phone numbers.

· Only one address will be kept for each Sale.

· Only one order number will be kept for each order received.

· Only one name will be kept for each order received.

**Data Structure:**

CUSTOMER (Customer ID​ (PK), Name, Address, Credit\_Card#, Order)​

**Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Customer ID | Number | Required, PK, Unique sequential ten-digit ID number, Indexed |
| Name | Text(14) | Required, name of the customer |
| Address | Text(20) | Required, address of the customer, must be valid U.S. zip code |
| Credit Card# | Number | Required, credit card number and information must be in format:  ####-####-####-####, MM/YY, ### |
| Order | Number | Required, ID of items in each order for customers must be in format:  #### |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 20 records per day | 20 updates per week, 50 queries per day | · Security to protect the personal information of the customer  · Applications: New Customer (C,R,U), Customer  Card Info (C,R,U,D), Customer ID (C,R) Customer  Address Info (C,R,U,D) |

Product​

· One Product can be in no Sold-Products or many Sold-Products, but a Sold-Product can have one and only one Product.

· A Product can have no Product-Calls or many Product-calls, but a Product-Call can be have by one and only one Product. · A Product can be in no Order-Products or many Order-Products, but an Order-Product can have one or many Products.

· A Product can have no Product-Calls or many Product-calls, but a Product-Call can be have by one and only one Product.

· Product\_ID is defined as the primary key for the Customer table.

· Only one phone number will be kept for each Customer. If additional phone numbers are needed, additional fields must be added to the table or an additional table created to contain phone numbers.

· Only one address will be kept for each Sale.

· Only one order number will be kept for each order received.

· Only one name will be kept for each order received.

**Data Structure:**

PRODUCT (Product\_ID​ (PK), Name, Description, Quantity\_Held, Order\_Quantity, Vendor\_Cost, Price\_Sold, Vendor\_ID(fk))​ **Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Product\_ID | Number | Required, PK, ID number given that is unique to the product offered |
| Name | Text (40) | Required, Name of the product offered |
| Description | Text (100) | Required, Description of the product offered |
| Quantity\_held | Number | Required, Number that represents how many of the items offered are in stock, available for purchase |
| Order\_Quantity | Number | Required, Number that represents how many of the items the customer purchased |
| Vendor\_Cost | Number | Required, Cost of the product that was bought from the vendor |
| Price\_Sold | Number | Required, the price that the product was sold at in the store |
| Vendor\_ID | Number | Required, fk, Unique sequential 4-digit ID number, Indexed |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 100 records per week | 100 updates per week, 75 queries per week | · Security for the customer and vendor’s private information  · Applications: New Product (C,R,U,D), Product Description  (R,U), Product Prices (C,R,U,D) Product Costs (R,U) |

**TABLE: Equipment** **Business Rules:**

· Equipment can be assigned to zero or many employees, but an employee can be assigned one or zero equipment. . Equipment ID is defined as the Primary Key for the Equipment Table **Data Structure:**

EQUIPMENT (Equipment\_ ID​ (​ PK), Equipment\_Name, Date\_of\_Use, Employee\_ID(fk))

**Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Equipment ID | Number | Required, Unique, PK, sequential 6-digit number, must be in format.#### |
| Equipment Name | Text(30) | Required, Name of equipment, text length could vary |
| Date of Use | Date/Time | Required, the date that the equipment was used by an employee. |
| Employee ID | Number | Required, fk, Unique, FK references Employee ID in the Employee table, sequential four-digit, must be in format: #### |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 25 records | 10 updates per day, 7 queries per day | · Security to protect equipment  · Applications: New Equipment (C,R,U) Broken Equipment (U,D) Employee Info (R,U,D) Date Used  (R,U) |

**TABLE: Sold\_Product** **Business Rules:**

* A Sold\_Product can have one and only one Product, but a Product can have none or many Sold\_Product
* A Sold\_Product an have one and only one Sale, but a Sale can have one or many Sold\_Product
* The Primary key for the Sold\_Product table is Sold\_Product\_ID
* The Two Foregin Keys that are visible in this Table are Sales\_ID and Product\_ID
* Quantity can be found in this table as well·

**Data Structure:**

SOLD\_PRODUCT (Sold\_Product\_ID ​ (​ PK), Quantity, Sales\_ID (fk), Product\_ID(fk))

**Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Sold Product ID | Number | Required, PK, product id of the sold item must be taken upon sale, sequential 3-digit number, must be in format.#### |
| Quantity | Number | Required, number of sold products per order must be taken before shipped, sequential 4-digit number, must be in format.#### |
| ProductName | varchar(50) | Name of the product |
| Product Description | varchar(50) | Description of Product |
| DatePurchased | Date | Date the product was purchased |
| EmployeeID | int | Employee who handled the sale |
| Salesman | varchar(50) | Name of Salesman |
| Name | varchar(50) | Name of Customer |
| Sales ID | Number | Required, fk, sales id must be taken before order is shipped, sequential 6-digit number, must be in format.#### |
| Product ID | Number | Required, fk, product id’s for all products within the order must be taken before shipped, sequential 6-digit number, must be in format.#### |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 20 records per day | 20 updates per day, 15 queries per day | · Security to protect customer information  · Applications: New Product Sold (C,R,U), Product Sold Info (C, R), Returned Product (R,U,D) Sale Info  (C,R,U) Quantity Info (R,U) |

**TABLE: Product\_Call** **Business Rules:**

* A Product\_Call can be apart of one and only one Call, but a Call can be associated with none or many Product\_Call
* A Product\_Call can be apart of one and only one Product, but a Product can be associated with none or many Product\_Call
* The Primary Key for this table is Product\_Call\_ID
* The Product\_Call Table has Foregin Keys of both Product\_ID and Call\_ID

·

**Data Structure:**

PRODUCT\_CALL (Product\_Call\_ ID​ (​ PK), Product\_ID (fk), Call\_ID(fk))

**Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Product Call ID | Number | Required, PK, each phone call that is related to a product will have its own 4 digit number and must be taken once the call is completed, sequential 5-digit number, must be in format.##-### |
| Product ID | Number | Required, fk, sequential 6-digit number, must be in format.###-### |
| Call ID | Number | Required, fk, Phone number from customer must be recorded, sequential 10-digit number, must be in format. ###-####-#### |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 25 records per day | 25 updates per day, 40 queries per day | · Security because it contains personal information of callers  · Applications: New Call (C,R), Call Directory  (R,U,D), Call Info (C,R,U), Customer Info (C,R,U),  Employee Info (C,R,U) |

**TABLE: Order\_Product**

**Business Rules:**

* + A Order\_Product can have one or more Orders, but an Order can have one and only one Order\_Product.
  + A Order\_Product can have zero or many Vendors, but a Vendor can have one and only one Order\_Product. ● Order\_Product\_ID is defined as the Primary Key for the Order\_Product table

·

**Data Structure:**

ORDER\_PRODUCT (Order\_Product\_ ID​ (​ PK), Quantity, Product\_ID (fk), Order\_#(fk))

**Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Order\_Product\_ ID | Number | Required, PK, Each order\_product\_ID will have its own number, sequential 4-digit number, must be in format.#-### |
| Quantity | Number | Required, How many products are in each order must be recorded, sequential 4-digit number, must be in format.#### |
| Product\_ID | Number | Required, fk, Product id must be recorded for each product within the order, sequential 6-digit number, must be in format.###-### |
| OrderID | Number | Required, fk, order number must be recorded before each order is shipped, sequential 6-digit number, must be in format. ####-##. |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 15 records per day | 15 updates per day, 10 queries per day | · Security of the customer’s private information  · Applications: New Order Product (C,R,U) Order  Returns (R,U,D) Product Info (R,U) Order Info (R,U)  Order Quantity (R,U) |

**TABLE: Order** **Business Rules:**

* + An Order can have zero or many Vendors, while a Vendor can only have one and only one Order.
  + An Order can have one and only one Order\_Product, while a Order\_Procuct can have one or more Order.
  + An Order can be assigned to zero or more Employees, but an Employee can be assigned one and only one Order.
  + Order # is defined as the Primary Key for the Order table.

·

**Data Structure:**

ORDER (OrderID​ (​ PK), Name, Address, Employee\_ID(fk), Vendor\_ID(fk))

**Attribute Description**:​

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Order ID | Number | Required, PK, Order number must be recorded, sequential 6-digit number, must be in format.####-## |
| Name | Text (40) | Required, Customers First and Last name must be recorded |
| Address | Text (100) | Required, Address of where the order will be delivered to must be recorded |
| Employee ID | Number | Required, fk, Employee id of employee who is creating order before it is shipped must be recorded, sequential 6-digit number, must be in format.##-##-## |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 15 records per day | 15 updates per day, 10 queries per day | · Security of the customer’s private information  · Applications: New Order Product (C,R,U) Order  Returns (R,U,D) Product Info (R,U) Order Info (R,U)  Order Quantity (R,U) |

**TABLE: Departments** **Business Rules:**

* A Department can have one or many Employees, but an Employee can be apart of none or one Department
* An Employee can supervise one or no Department, but a Department may have one and only one employee supervise them.
* The Primary Key for this table is Department\_ID
* There are no foregin keys located in this table as there is not a many relationship located to departments - All the data from department will be kept inside department

·

**Data Structure:**

DEPARTMENTS (​Dep ID​(PK), Dep\_ Phone#, Dep\_Name) **Attribute Description**​:

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Comments\*: |
| Department ID | Number | Required, Unique, PK, number that relates to specific department, sequential 4-digit number, must be in format.##-## |
| Dep\_Phone# | Number | Required, phone number pertaining to each department, sequential 6-digit number, must be in format.###-####-#### |
| Dep\_Name | Text(40) | Required, Name of each department |

**Usage and Performance Issues:**

|  |  |  |
| --- | --- | --- |
| Data Volume | Data Volatility & Throughput | Additional Considerations |
| 10 records | 10 updates per day, 25 queries per day | · Security to protect employee information in each department  · Applications: New Department (C,R,U) Department  Downsizing (R,U,D) Department Directory (C,R,U,D) |

TABLES

|  |
| --- |
| VENDOR |
|  |

|  |
| --- |
| EMPLOYEE |
|  |

|  |
| --- |
| CALL |
|  |

|  |
| --- |
| CUSTOMER |
|  |

|  |
| --- |
| PRODUCT |
|  |

|  |
| --- |
| SALES |
|  |

|  |
| --- |
| Equipment |
|  |

|  |
| --- |
| ProductCall |
|  |

|  |
| --- |
| SoldProduct |
|  |

|  |
| --- |
| ORDER\_PRODUCT |
|  |

|  |
| --- |
| ORDERS |
|  |

|  |
| --- |
| DEPARTMENTS |
|  |

Queries

|  |
| --- |
| (**Query**​ )​  1. List all fields from table Employee. Sort by employee last name |
| (**SQL Statement**​ )​  SELECT \*  FROM EMPLOYEE  ORDER BY (LastName); |
| (​**Results**​) |

|  |
| --- |
| (**Query**​ )​  2. List all products provided by vendor *XXX.*​​(*XXX*​ corresponds to the name of one of your vendors). |
| (​**SQL Statement**​)  SELECT​ ​\*  FROM​ [QUINNIPIAC\msdonahue]​.​[Product]  WHERE​ [Vendor\_ID] ​=​ 11 |
| (​**Results**​) |

|  |
| --- |
| (**Query**​ )​  3. List all sales. For each sale, list the products sold (name and description of product), quantity, the employee name who handled the transaction and the customer who made the purchase. The date of the sale should also be retrieved. List all purchases made prior to today. (you may have to use many joins) |
| (**SQL Statement**​ )​  SELECT SOLD\_PRODCUT​.​Salesman ​AS EmployeeName​,  [QUINNIPIAC\lrmacmillan]​.​[SOLD\_PRODCUT]​.​Name​ ​AS​ CustomerName​,  [QUINNIPIAC\lrmacmillan]​.​[SOLD\_PRODCUT]​.​DatePurchased ​AS​ SaleDate​,  [QUINNIPIAC\lrmacmillan]​.​[SOLD\_PRODCUT]​.​SalesID​,  [QUINNIPIAC\lrmacmillan]​.​[SOLD\_PRODCUT]​.​ProductName​,  [QUINNIPIAC\lrmacmillan]​.​[SOLD\_PRODCUT]​.​ProductDescripition​,  [QUINNIPIAC\lrmacmillan]​.​[SOLD\_PRODCUT]​.​ProductID​,  [QUINNIPIAC\lrmacmillan]​.​[SOLD\_PRODCUT]​.​Quantity  FROM​ [QUINNIPIAC\lrmacmillan].​ [​ SOLD\_PRODCUT]  ​LEFT​ ​JOIN​ [QUINNIPIAC\msdonahue]​.​[Sales]  ON [SOLD\_PRODCUT]​.​[Salesman] ​= [Sales].​​[Salesman] LEFT​ ​JOIN  [QUINNIPIAC\msdonahue]​.​[Customer]  ON​ [SOLD\_PRODCUT]​.​[Name] ​=​ [Customer]​.​[Name]  WHERE​ [QUINNIPIAC\lrmacmillan]​.​[SOLD\_PRODCUT]​.​DatePurchased ​<​ ​GETDATE​(); |
| (​**Results**​) |

|  |
| --- |
| (**Query**​ )​  4. List all different departments and the count of employees assigned to each department. |
| (**SQL Statement**​ )​  SELECT[​ QUINNIPIAC\jozarzano].[EMPLOYEE].[DepartmentID],  [Dep\_Name] , Count​ (​ EMPLOYEE.DepartmentID) as​ CountEmployees​  From [QUINNIPIAC\akfeuilly].[DEPARTMENTS] RIGHT JOIN​  [QUINNIPIAC\jozarzano].[EMPLOYEE] ON​  [QUINNIPIAC\akfeuilly].[DEPARTMENTS].DepartmentID =  EMPLOYEE.DepartmentID  Group By [QUINNIPIAC\jozarzano].[EMPLOYEE].[DepartmentID], Dep\_Name ; |
| (​**Results**​) |

|  |
| --- |
| (**Query**​ )​  5. Insert department named ‘R&D’ to the department table |
| (​**SQL Statement**​)  INSERT INTO​ DEPARTMENTS (DepartmentID, Dep\_Phone#, Dep\_NAME)  VALUES​ (​‘3427’​, ​‘2039471873’,​ ​‘R & D’​); |
| (​**Results**​) |

|  |
| --- |
| (**Query**​ )​  6. Delete R&D from the department Table. |
| (​**SQL Statement**​)  DELETE FROM​ ​DEPARTMENTS  WHERE​ Dep\_Name='​R & D'​; |
| (​**Results**​) |

|  |
| --- |
| (**Query**​ )​  7. List all the average prices for your products. |
| (​**SQL Statement**​)  SELECT​ Name, ​AVG​(Price\_Sold\_PerItem) ​as​ AvgPrice  From Product  Group By Name; |
| (​**Results**​) |

|  |
| --- |
| (**Query**​ )​  8. Update the cost of one of your products from $$ to $$$. |
| (​**SQL Statement**​)  UPDATE​ Product  SET​ Price\_Sold\_PerItem = 200  WHERE​ Product\_ID= 10002; |
| (​**Results**​) |

|  |
| --- |
| (**Query**​ )​  9. Create a query using the following requirements:   1. Joining two tables 2. Using an outer join (right or left) |
| (**SQL Statement**​ )​  SELECT \*​  FROM [QUINNIPIAC\jozarzano].[EMPLOYEE] RIGHT​ JOIN​ [QUINNIPIAC\jozarzano].[CALL]  ON [QUINNIPIAC\jozarzano].[EMPLOYEE].EmployeeID =  [QUINNIPIAC\jozarzano].[CALL].EmployeeID |
| (​**Results**​) |

|  |
| --- |
| (**Query**​ )​  10. Create a query using the following requirements:   1. Joining two tables or more 2. Using an aggregate functions (SUM, AVG, MAX, MIN, COUNT) c. Using a filter |
| (​**SQL Statement**​)  SELECT Product.Name, Product.Descriptioon, Product.Quantity\_Held,  SUM​([Vendor\_Cost\_Total])  FROM​ [QUINNIPIAC\msdonahue].[Product] ​INNER JOIN  [QUINNIPIAC\lrmacmillan].[SOLD\_PRODCUT]  ON [QUINNIPIAC\msdonahue].[Product].Product\_ID =  [QUINNIPIAC\lrmacmillan].[SOLD\_PRODCUT].[ProductID]  Group By​ Product.​Name​, Descriptioon, Quantity\_Held Having​ Quantity\_Held >'​10​' |
| (​**Results**​) |