

STEP 1: Topic/Domain Selection

Topic/Application:

"Vital Ring Healthcare Case Management" system is the case management application that serves the patients who have issues/questions/complaints/requests regarding the healthcare product Vital Ring which measures the vital characteristics by wearing a ring.

Introduction:

The "Vital Ring Healthcare Case Management" system is designed to manage and streamline healthcare services. It focuses on organizing patient data, case management, and service quality assurance. The system integrates various entities like patient information, service requests, capture service orders and feedback mechanisms. This integration enables organization to efficiently manage patient cases, track service requests, and ensure high-quality care. The system's design aims to improve healthcare delivery by providing a structured and comprehensive approach to case management, enhancing both patient experience and operational efficiency in healthcare settings.

Advantages:

The "Vital Ring Healthcare Case Management" system offers several advantages:

1. Enhanced Data Organization: The system facilitates better organization and accessibility of patient and case information, making it easier for healthcare providers to access and manage crucial data efficiently.
2. Improved Service Efficiency: By streamlining case management processes, the system leads to quicker response times, enhancing the overall efficiency and effectiveness of patient services.
3. Quality Control and Feedback: Incorporating feedback mechanisms enables continuous improvement in healthcare services, ensuring that patient care is consistently evolving and adapting to meet changing needs.

Use Cases:

The following industries and areas use similar database system to "Vital Ring Healthcare Case Management" system:

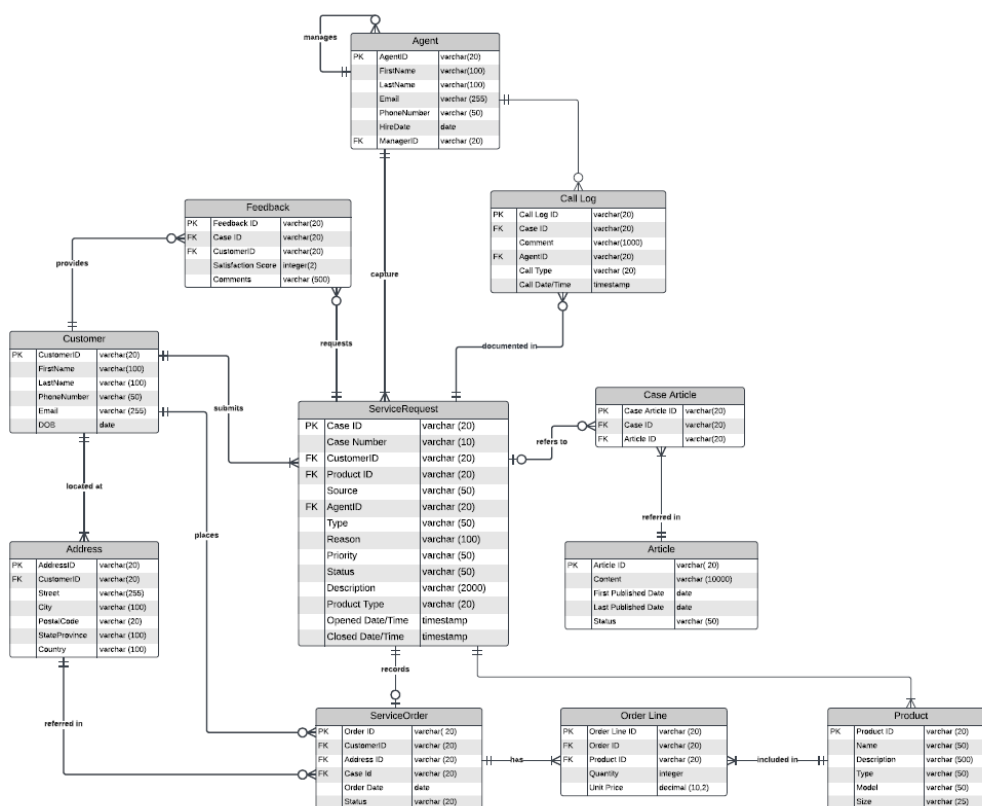
1. Telemedicine Services: They have a similar type of database system, which is ideal for coordinating remote medical consultations and effectively managing electronic health records, enhancing the delivery of healthcare services over distance.
2. Insurance Companies: A similar database system can be used in streamlining the processing of healthcare claims and improving the efficiency of customer case management.
3. Public Health Departments: The system supports the management of community health initiatives and the tracking of patient demographics, crucial for public health monitoring and intervention.

4. Pharmaceutical Companies: Useful in managing patient support programs and facilitating drug efficacy studies, the system can aid in tracking patient responses and outcomes.
5. Elderly Care Facilities: It's well-suited for managing care plans for the elderly and tracking health interventions, ensuring personalized and continuous care for residents.

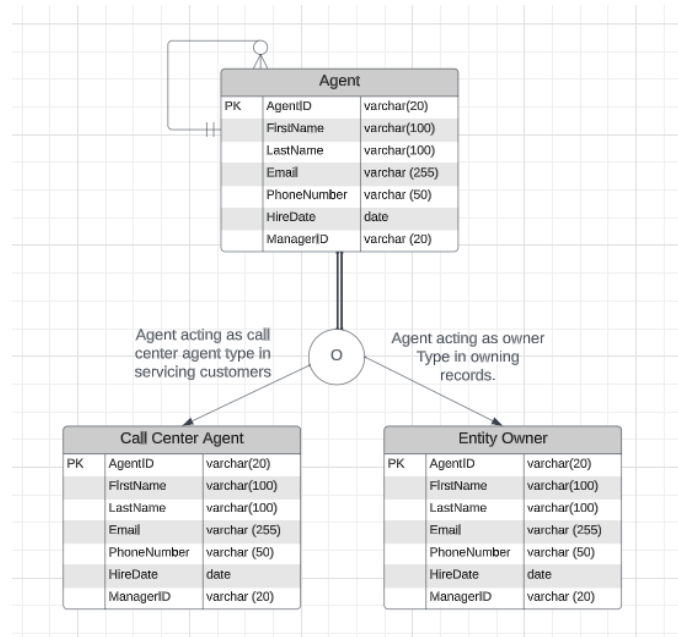
Similar system might be used in any other industry requiring detailed case management, such as legal case management, social services, or customer relationship management in various sectors.

STEP 2: Topic/Domain Selection

ER Diagram with relevant business rules in 3rd normalization



EER – Agent is an employee who will act as a call center agent and act as an owner for a certain task. They perform 2 different functions but belong to one persona.



STEP 3: Database Implementation

Attached the scripts that were executed in a separate file.

STEP 4,5: Deploy Database on AWS Cloud

Our database was deployed to AWS Cloud. Required details were included in the python colab file that is shared separately.

STEP 5: App-Database Interaction

Below are the possible interactions we executed against our database

1. What are the projects with the most associated cases?

```

SELECT
P.ProductID,
P.Name AS ProductName,
COUNT(C.CaseID) AS AssociatedCases
FROM
Product P
JOIN
VitalRingService.ServiceRequest C ON P.ProductID = C.ProductID
GROUP BY

```

P.ProductID, P.Name
ORDER BY
AssociatedCases DESC
LIMIT 1;

2. Order by status count

SELECT
Status,
COUNT(*) AS StatusCount
FROM
VitalRingService.ServiceOrder
GROUP BY
Status
ORDER BY
StatusCount DESC;

3. Query all orders including the products associated to the order grouped by customer

SELECT
C.CustomerID,
C.FirstName,
C.LastName,
O.OrderID,
O.OrderDate,
P.ProductID,
P.Name AS ProductName,
OL.Quantity,
OL.UnitPrice
FROM
Customer C
JOIN
VitalRingService.ServiceOrder O ON C.CustomerID = O.CustomerID
JOIN
OrderLine OL ON O.OrderID = OL.OrderID
JOIN
Product P ON OL.ProductID = P.ProductID
ORDER BY
C.CustomerID, O.OrderID, P.ProductID;

4. Order agents by the number of cases they solved

```
SELECT Agent.AgentID, FirstName, LastName, t.NumberOfCases
FROM Agent
JOIN
(
SELECT AgentID, COUNT(CaseID) AS NumberOfCases
FROM ServiceRequest
GROUP BY AgentID
) AS t ON Agent.AgentID=t.AgentID
ORDER BY t.NumberOfCases Desc
```

5. Determine the most successful team, that handled the most cases.

We have 2 managers each managing a team of agents and the third group consists of cases handled by managers themselves.

```
SELECT COALESCE(CONCAT( Manager.FirstName, ' ', Manager.LastName, "'s
team'),'Handled by manager') AS Team, SUM(t.NumberOfCases) AS
NumberOfCases
FROM Agent
LEFT JOIN
(
SELECT AgentID, COUNT(CaseID) AS NumberOfCases
FROM ServiceRequest
GROUP BY AgentID
) AS t ON Agent.AgentID=t.AgentID
Left Join Agent as Manager On Agent.ManagerID=Manager.AgentID
GROUP BY 1
```

6. List all customers including the case count for each customer

```
SELECT
Cust.CustomerID,
Cust.FirstName,
Cust.LastName,
COUNT(sr.CaseID) AS CaseCount
```

```
FROM
Customer Cust
LEFT JOIN ServiceRequest sr ON Cust.CustomerID = sr.CustomerID
GROUP BY
Cust.CustomerID, Cust.FirstName, Cust.LastName
ORDER BY
Cust.CustomerID;

SELECT sr.CaseID, COUNT(cl.CallLogID) AS CallLogCount
```

7. Call Log Count by Cases

```
FROM ServiceRequest sr
JOIN CallLog cl ON sr.CaseID = cl.CaseID
GROUP BY sr.CaseID
ORDER BY CallLogCount DESC;
```

8. A complex query that retrieves all customers, the number of cases they've opened, the number of calls logged for them, and the average satisfaction score from their feedback.

```
SELECT
Cust.CustomerID, Cust.FirstName, Cust.LastName,
COUNT(DISTINCT sr.CaseID) AS NumberOfCases,
COUNT(DISTINCT cl.CallLogID) AS NumberOfCalls,
AVG(Fdbk.SatisfactionScore) AS AverageSatisfactionScore
FROM
Customer AS Cust
LEFT JOIN ServiceRequest AS sr ON Cust.CustomerID = sr.CustomerID
LEFT JOIN CallLog AS cl ON sr.CaseID = cl.CaseID
LEFT JOIN Feedback AS Fdbk ON Cust.CustomerID = Fdbk.CustomerID
GROUP BY
Cust.CustomerID, Cust.FirstName, Cust.LastName;
```

9. Articles Count by Case Usage

```
SELECT
a.ArticleID as articleId, a.Content as content, count(*) as usage_count
FROM CaseArticle ca, Article a, ServiceRequest sr
WHERE ca.CaseID = sr.CaseID and
```

```
ca.ArticleID = a.ArticleID  
GROUP BY ca.ArticleID having count(*) > 1
```

10. Row Count For All Tables

```
SELECT 'Agent' AS TableName, COUNT(*) AS RowCount FROM Agent  
UNION ALL  
SELECT 'Feedback', COUNT(*) FROM Feedback  
UNION ALL  
SELECT 'Customer', COUNT(*) FROM Customer  
UNION ALL  
SELECT 'Address', COUNT(*) FROM Address  
UNION ALL  
SELECT 'ServiceRequest', COUNT(*) FROM ServiceRequest  
UNION ALL  
SELECT 'ServiceOrder', COUNT(*) FROM ServiceOrder  
UNION ALL  
SELECT 'OrderLine', COUNT(*) FROM OrderLine  
UNION ALL  
SELECT 'Product', COUNT(*) FROM Product  
UNION ALL  
SELECT 'Article', COUNT(*) FROM Article  
UNION ALL  
SELECT 'CaseArticle', COUNT(*) FROM CaseArticle  
UNION ALL  
SELECT 'CallLog', COUNT(*) FROM CallLog;
```

Business Rules For the Vital Ring Service

Case Creation:

Rule: A new case is created when a healthcare device requires attention or maintenance.

Conditions: The creation must include essential information such as device type, serial number, reported issue, and contact details of the person reporting the problem.

Case Assignment:

Rule: Cases are assigned to the appropriate technician or support personnel.

Conditions: Assignment is based on the type of device, technician expertise, workload, and geographical location.

Case Prioritization:

Rule: Cases are prioritized based on the severity of the issue and its impact on patient care.

Conditions: Severity levels are defined (e.g., critical, high, medium, low), and prioritization is automatic based on these levels.

Communication with Stakeholders:

Rule: Regular updates are provided to stakeholders, including the person who reported the issue and relevant healthcare professionals.

Conditions: Updates may include case status, expected resolution time, and any additional information required from the stakeholders.

Device Inspection and Diagnosis:

Rule: Technicians perform a thorough inspection and diagnosis of the healthcare device to identify the root cause.

Conditions: Diagnostic procedures are defined for each device type, and technicians have access to historical data and maintenance records.

Inventory Management:

Rule: If a replacement part is required, the system checks the inventory for availability.

Conditions: If the required part is not in stock, an order is automatically placed with the supplier, and the case status is updated accordingly.

Resolution and Documentation:

Rule: Cases are marked as resolved once the issue is fixed, and detailed documentation is updated.

Conditions: Documentation includes the actions taken, replacement parts used, and any recommendations for future maintenance.

Quality Assurance:

Rule: A quality check is performed on resolved cases to ensure the effectiveness of the solution.

Conditions: Random sampling may be used to select cases for quality assurance, and feedback is provided to technicians for continuous improvement.

Escalation Procedures:

Rule: Cases with prolonged resolution times or unresolved critical issues are escalated to higher-level support or management.

Conditions: Escalation criteria are defined, and notifications are sent to the appropriate personnel when these criteria are met.

Reporting and Analytics:

Rule: Regular reports and analytics are generated to track case resolution times, common issues, and technician performance.

Conditions: Reports are accessible to management for strategic decision-making and process improvement.

These business process rules provide a foundation for developing the SQL database schema, stored procedures, and triggers to automate and manage the healthcare devices case management system effectively.