**Phase -1 Project Report**

**CSCI 8856 DATABASE MANAGEMENT SYSTEMS**

**CUSTOMER SERVICE MANAGEMENT**

As the name suggests, the application is management of customer service center. A customer service center receives thousands of call a day. Some might be related to knowing more information about a service or a plan and some might be about a problem in their equipment which can be anything from a hand-held device to a machine that is used for a purpose. We can perform different types of queries on this application like how many callers has called the customer service about the problems faced for a type of error, calculating the efficiency of employees like operators and specialists and which are most number of problems that are resolved. This data can help in evaluating the service center and maintaining the center efficiently. The database system stores the information of every call made along with their name and email, operator who talked to them, different types of problems reported by callers, administrators who are managing the problems and specialists working to resolve the problems by category.

For evaluating an employee and tracking down the status of a request/problem every call that is made to the center is logged. Every call must be attended by an employee known as operator but not every operator needs to have caller on the line. Typically, the operator attends the call that is routed to her by the system providing information asked by the caller or starts a service request for the problems the caller faced in his equipment. This information is logged in logs which are then processed to know the request for service of the problem. Every operator needs to enter the information about the call she answered into the log and every log must be created by the operator only. Logs will have certain information with them such as caller id, operator id and problem id. Every problem must be logged along with certain information like id, Description, status, category id. Category gives the description about the problem, id of specialist assigned to tackle that problem and unique id to identify that category. The specialist is an employee who is specialized in a field and is assigned problems to solve them. One specialist will have only one problem assigned to him and each problem will have only one specialist. An employee known as administrator oversees every category i.e. each category has an administrator to supervise that departments activities. Every category must be managed by an administrator but administrator will have only one category under him. Every problem that has been resolved will be stored with details such as problem id, id of specialist who solved that problem, description about the solution along with a unique number to identify one from another.

There are different decisions made when modeling the application. First one is, decision to reduce the complexity of the E-R diagram. To achieve this I have not included two entities Manufacture and Problem type in the draft and in this phase-1 report. Second one is to include the specialization, Employee entity which won’t affect the Relational model. There is also a decision made to change the relations contains, assigned, manages and have from one to many/ many to one to one to one to include those relations in relational design. Why other constraints and cardinalities are placed between those entities and relations is explained above.

The following are list of entities that are participating in Entity Relationship diagram

* Caller

Name, Caller\_id(Primary key), Phone(multi-valued), Email

* Employee

Name, id(Primary key), phone(multi-valued), Password

* Operator

Name, id(Primary key), phone(multi-valued), Password, status

* Administrator

Name, id(Primary key), phone(multi-valued), Password

* Specialist

Name, id(Primary key), phone(multi-valued), Password, Reputation, Specialization, Catergory\_id

* Problem Category

Description, Specialist\_id, Id(Primary key)

* Resolved (Weak entity)

Problem\_id, Id(), specialist\_id, Description

* Call Log

Call\_id(Primary Key), Caller\_id

* Problem

Id(Primary key), Description, Status

The relations participating in the ER diagram are

* Manages

The relationship is in between Problem Category and Administrator

* Logs

The relationship is between Operator and Call Log

* Contains

The relationship is between Call Log and Problem

* have

The relationship is between Problem and Resolved

* Talks

The relationship is between Caller and Operator

* Category

The relationship is between Problem and Problem Category

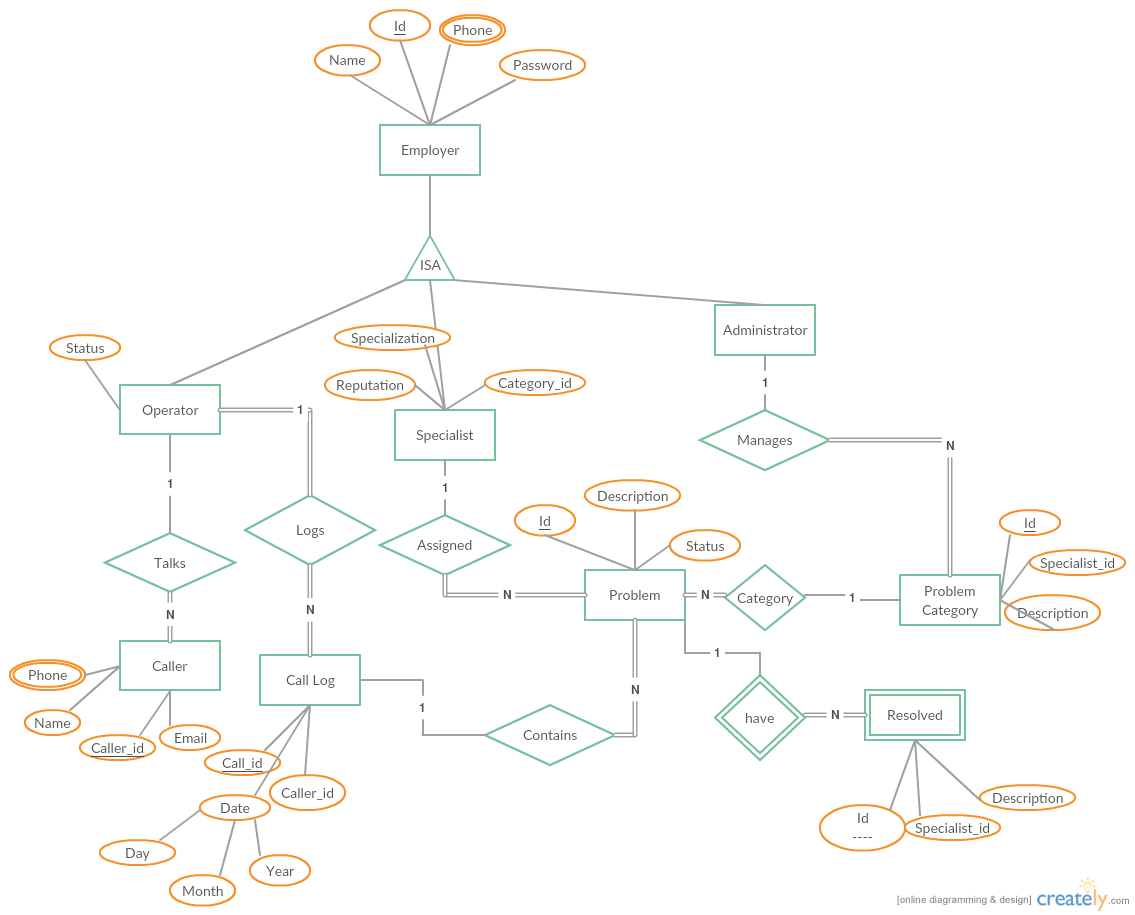
* Assigned

The relationship is between Problem and Specialist

* ISA

The relationship is specialization from employee to Operator, Administrator and Specialist

**Entity Relationship Diagram for Customer Service Management**



The queries given below create the relational design for the above E-R diagram. The total participation constraints cannot be enforced in MYSQL so they must be enforced using triggers during on commit. Some relations are not given as tables like talks and category because they have many and total participation.

**SQL Queries for creation of tables for Entities:**

CREATE TABLE `dbms`.`Specialist` ( `Name` VARCHAR(25) NOT NULL , `Id` INT(10) NOT NULL , `Password` VARCHAR(8) NOT NULL , `Reputation` INT(2) NOT NULL , `Specialization` VARCHAR(10) NOT NULL , `Category\_id` INT(10) NOT NULL , PRIMARY KEY (`Id`), FOREIGN KEY (`Category\_id`) REFERENCES `problem\_category`(`Id`));

CREATE TABLE `dbms`.`operator` ( `Name` VARCHAR(25) NOT NULL , `Id` INT(10) NOT NULL , `Password` VARCHAR(8) NOT NULL ,`Status` INT(1) NOT NULL , `Caller\_id` INT(15) NOT NULL , PRIMARY KEY (`Id`), FOREIGN KEY (`Caller\_id`) REFERENCES `caller`(`Caller\_id`));

CREATE TABLE `dbms`.`Caller` ( `Name` VARCHAR(25) NOT NULL , `Caller\_id` INT(15) NOT NULL , `Email` VARCHAR(25) NOT NULL, `Operator\_id` INT(10) NOT NULL , PRIMARY KEY (`Caller\_id`),FOREIGN KEY (`Operator\_id`) REFERENCES `operator`(`Id`));

CREATE TABLE `dbms`.`Administrator` ( `Name` VARCHAR(25) NOT NULL , `Id` INT(10) NOT NULL , `Password` VARCHAR(8) NOT NULL , PRIMARY KEY (`Id`));

CREATE TABLE `dbms`.`Problem\_Category` (`Id` INT(10) NOT NULL , `Password` VARCHAR(25) NOT NULL , `Description` INT(10) NOT NULL , PRIMARY KEY (`Id`));

CREATE TABLE `dbms`.`Call\_Log` ( `Call\_id` INT(10) NOT NULL , `Caller\_id` INT(10) NOT NULL , `Day` INT(2) NOT NULL , `Month` INT(2) NOT NULL , `Year` INT(4) NOT NULL , PRIMARY KEY (`Call\_id`), PRIMARY KEY (`Caller\_id`),FOREIGN KEY (`Caller\_id`) REFERENCES `caller`(`Caller\_Id`));

CREATE TABLE `dbms`.`Resolved` ( `Id` INT(10) NOT NULL , `Specialist\_id` INT(10) NOT NULL , `Description` VARCHAR(25) NOT NULL , `Problem\_id` INT(10) NOT NULL , PRIMARY KEY (`Id`), FOREIGN KEY (`Specialist\_id`) REFERENCES `specialist`(`Id`), FOREIGN KEY (`Problem\_id`) REFERENCES `problem`(` Id`) );

CREATE TABLE `dbms`.`Problem` ( `Id` INT(10) NOT NULL , `Description` VARCHAR(25) NOT NULL , `Status` INT(1) NOT NULL , `Category\_id` INT NOT NULL , PRIMARY KEY (`Id`) , FOREIGN KEY (`Category\_id`) REFERENCES `problem\_category`(`Id`));

CREATE TABLE `dbms`.`phone\_cus` ( `Id` INT(10) NOT NULL , `Phone` INT(10) NOT NULL , PRIMARY KEY (`Id`, `Phone`),FOREIGN KEY (`Id`) REFERENCES `caller`(`Caller\_id`) );

CREATE TABLE `dbms`.`phone\_emp` ( `Id` INT(10) NOT NULL , `phone` INT(10) NOT NULL , PRIMARY KEY (`Id`, `phone`),FOREIGN KEY (`Id`) REFERENCES `administrator`(`Id`),FOREIGN KEY (`Id`) REFERENCES `specialist`(`Id`),FOREIGN KEY (`Id`) REFERENCES `operator`(`Id`));

**SQL Queries for creation of tables for Relations:**

CREATE TABLE `dbms`.`manages` ( `Admin\_id` INT(10) NOT NULL , `Category\_id` INT(10) NOT NULL , PRIMARY KEY (`Admin\_id`, `Category\_id`), FOREIGN KEY (`Category\_id`) REFERENCES `problem\_category`(`Id`), FOREIGN KEY (`Admin\_id`) REFERENCES `administrator`(`Id`));

CREATE TABLE `dbms`.`Logs` ( `Call\_id` INT(10) NOT NULL , `Operator\_id` INT(10) NOT NULL , PRIMARY KEY (`Call\_id`), FOREIGN KEY (`Call\_id`) REFERENCES `call\_log`(`Call\_id`), FOREIGN KEY (`Operator\_id`) REFERENCES `operator`(`Id`));

CREATE TABLE `dbms`.`Assigned` ( `Specialist\_id` INT(10) NOT NULL , `Problem\_id` INT(10) NOT NULL , PRIMARY KEY (`Specialist\_id`, `Problem\_id`) , FOREIGN KEY (`Problem\_id`) REFERENCES `problem`(`Id`), FOREIGN KEY (`Specialist\_id`) REFERENCES `Specialist`(`Id`));

CREATE TABLE `dbms`.`Contains` ( `Call\_id` INT(10) NOT NULL , `Problem\_id` INT(10) NOT NULL , PRIMARY KEY (`Call\_id`, `Problem\_id`), FOREIGN KEY (`Call\_id`) REFERENCES `call\_log`(`Call\_id`), FOREIGN KEY (`Problem\_id`) REFERENCES `problem`(`Id`));

# The following data is copy of the report generated using PhpMyAdmin using export option and file type as MS Word

# Database dbms

## Table structure for table administrator

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| Name | varchar(25) | No |  |
| ***Id*** | int(10) | No |  |
| Password | varchar(8) | No |  |

## Dumping data for table administrator

|  |  |  |
| --- | --- | --- |
| Kristofer Braodbent | 432081046 | kristof |
| Jennell Crane | 997773734 | Jennell |

## 

## Table structure for table assigned

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***Specialist\_id*** | int(10) | No |  |
| ***Problem\_id*** | int(10) | No |  |

## Dumping data for table assigned

|  |  |
| --- | --- |
| 164666188 | 2 |
| 227657699 | 1 |

## Table structure for table caller

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| Name | varchar(25) | No |  |
| ***Caller\_id*** | int(15) | No |  |
| Email | varchar(25) | No |  |
| Operator\_id | int(10) | No |  |

## Dumping data for table caller

|  |  |  |  |
| --- | --- | --- | --- |
| William Kusko | 127961616 | wk@gmail.com | 197616018 |
| Alisha Sergi | 949700884 | al@gmail.com | 237490166 |

## Table structure for table call\_log

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***Call\_id*** | int(10) | No |  |
| Caller\_id | int(10) | No |  |
| Day | int(2) | No |  |
| Month | int(2) | No |  |
| Year | int(4) | No |  |

## Dumping data for table call\_log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 127961616 | 2 | 3 | 2017 |
| 2 | 949700884 | 7 | 1 | 2017 |

## Table structure for table contains

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***Call\_id*** | int(10) | No |  |
| ***Problem\_id*** | int(10) | No |  |

## Dumping data for table contains

|  |  |
| --- | --- |
| 1 | 1 |
| 2 | 2 |

## Table structure for table logs

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***Call\_id*** | int(10) | No |  |
| Operator\_id | int(10) | No |  |

## Dumping data for table logs

|  |  |
| --- | --- |
| 1 | 197616018 |
| 2 | 237490166 |

## Table structure for table manages

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***Admin\_id*** | int(10) | No |  |
| ***Category\_id*** | int(10) | No |  |

## Dumping data for table manages

|  |  |
| --- | --- |
| 997773734 | 1 |
| 432081046 | 2 |

## Table structure for table operator

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| Name | varchar(25) | No |  |
| ***Id*** | int(10) | No |  |
| Password | varchar(8) | No |  |
| Status | int(1) | No |  |

## Dumping data for table operator

|  |  |  |  |
| --- | --- | --- | --- |
| Frankie Van | 197616018 | Frankie | 1 |
| Lanell Colpa | 237490166 | Lanell | 1 |
| Dorsey Stewart | 739583429 | Dorsey | 0 |

## Table structure for table problem

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***Id*** | int(10) | No |  |
| Description | varchar(25) | No |  |
| Status | int(1) | No |  |
| Category\_id | int(11) | No |  |

## Dumping data for table problem

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | OS Error | 0 | 1 |
| 2 | App Error | 1 | 2 |

## Table structure for table problem\_category

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***Id*** | int(10) | No |  |
| Description | varchar(25) | No |  |

## Dumping data for table problem\_category

|  |  |
| --- | --- |
| 1 | OS Error |
| 2 | App Error |

## Table structure for table resolved

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| ***Id*** | int(10) | No |  |
| Specialist\_id | int(10) | No |  |
| Description | varchar(25) | No |  |
| Problem\_id | int(10) | No |  |

## Dumping data for table resolved

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 164666188 | Updated App/Code | 2 |

## Table structure for table specialist

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** |
| Name | varchar(25) | No |  |
| ***Id*** | int(10) | No |  |
| Password | varchar(8) | No |  |
| Reputation | int(2) | No |  |
| Specialization | varchar(10) | No |  |
| Category\_id | int(10) | No |  |

## Dumping data for table specialist

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mirta Mallet | 164666188 | Mirta | 9 | App | 2 |
| Haydee Denooyer | 227657699 | Haydee | 7 | OS | 1 |