Information Systems and Data Modeling – IT1090

Assignment 2

Title: Online Customer Support System	
Batch Number: 14.1	Group Number: MLB_14.01_05

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged.

We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

IT22320582	Jayasundara D.W.S	Wjayas ………………..
IT22305350	Rajapakshe P.H.Y.L	Yasas Hamina …………………
IT22337412	D.Rajapaksha	………………….
IT22341136	M.A.S Gunathilaka	Ashan. …………………..
IT22315632	R.N.D K.Rajapaksha	NRajapaksha ………………….

Submitted on: 01/06/2023

1) Hypothetical scenario

The use of online customer support system has increased in popularity in today's fast-paced, digitally-driven environment. These platforms provide an essential service for registered users to access help and information on a range of services. Users can use the system knowledge base or speak with specialized customer support agent directly for quick resolutions. They must purchase one of the several service plans according to access resources. When the user raises a ticket the system assigns a support agent to solve the issue. If that support agent fails to solve the tickets system redirect the user’s ticket to another department. Additionally, the user has the option to leave feedback, which customer support agent checks and assists to estimate user experiences and provides helpful data for the system's advancing improvement. The system manager is responsible for managing the departments. The manager generally reviews the reports generated by the system administrator to find new information and possible improvement areas. overall, customer support agent, system administrators and system managers work together to run the help desk online customer support system.

2) Requirements Analysis

An efficient online customer support system is designed and developed on the basis of the requirements analysis phase. It makes ensuring that the system addresses important functional and non-functional features while also being in line with business and consumer needs. An effective system implementation is eventually made possible by the analysis, which aids in prioritizing needs, identifying dependencies, and defining clear expectations for the development team.

a. Main Requirement of the system

The main requirement of an online customer support system is to provide efficient and effective support to customers, addressing their inquiries, issues, and concerns in a timely manner. The system should facilitate seamless communication and interaction between customers and support agents, enabling the resolution of problems and the provision of assistance. An efficient online customer support system is designed and developed on the basis of the requirements analysis phase. It makes ensuring that the system addresses important functional and non-functional features while also being in line with business and client needs.

b. Functional Requirements

The functional requirements of an online customer support system play a vital role in ensuring efficient and effective communication between customers and support agents. These requirements define the specific features and functionalities that the system must have to meet the needs and expectations of both customers and the support team. Our online customer support system is created based on functional requirements and this system has several main functional requirements.

- Unregistered user
- Registered user
- System administrator
- Customer support agent
- Manager
- Developer

Unregistered user requirements:

- User Registration: The system should provide an inbuilt and user-friendly registration process for unregistered users.
- Browsing for the services: Unregistered users should have access to the given knowledge base, FAQ, video tutorials and user guides part where they can look up answers to frequently asked questions or relevant articles.
- Giving feedback: User feedback is essential for enhancing the quality of services offered.

Registered user requirements:

- Registered user logs into the account.
- Registered user can request a service.
- Registered user can access the knowledge base.
- Registered user can receive regular updates.

System administrator requirements:

- System administrator logs into the admin account.
- System administrator can monitor the system performance.
- System administrator manages the knowledge base.
- System administrator manages user/staff accounts.
- System administrator can troubleshoot technical issues.

Customer support agent requirements

- Customer support agent should be able to log in to the system using agent user credentials.
- Customer support agent should have the access to the knowledgebase to retrieve information and the data.
- System should allow to customer support agents to handle received raised tickets.
- Customer support agent should be able to escalate tickets which want a higher level or another department support.
- Customer support agent should be able to review feedback sent by users.

Manager requirements

- The manager manages the customer support team.
- The manager should have the ability to manage user accounts.
- The manager should have the ability to performance track and report to the customer support team.
- The manager should have communication and collaboration with the customer support team.
- The manager can manage and update database.

Developer requirements

- The Developer should be able to design the system.
- The Developer should be able to develop the system.
- The Developer should be able to upgrade the system.

- The Developer should be able to maintain the system.
- The Developer should be able to add new features to the system.
- The Developer should be able to test the system.
- The Developer should be able to check the quality of the system.
- The Developer should be able to fix the bugs in the system.
- The Developer should be able to integrate the other system.

c) Non functional requirements

- Response Time:

There should not be any delay when the customer support agent provides request information to the user. The user should be receiving information within an acceptable time frame.

- Scalability:

The System should be able to handle multiple users and customer support agents' activities and support tickets without any without visibly breaking its performance.

- Throughput:

The system should be able to manage numerous transactions or interactions at once including handling incoming support tickets, assigning them to qualified agents, and ensuring effective dialogue between users and agents.

- Availability:

When a user needs to access information, the system should be accessible without any issues with the system services.

- Reliability:

The probability of errors, breakdowns, or interruptions should be kept to a minimum because of the system's dependability and stability.

- Database Performance:

  The system's database should operate at its maximum effectiveness, providing quick and effective data retrieval and storage. This should include handling large number of support tickets, system date at a time.

d) Data requirements of the system

Manager

- Manager id (MID)
- Manager first name (M_Fname)
- Manager last name (M_Lname)
- Manager email (M_Email)

Knowledgebase

- Article id (ArticalID)
- Article category (Category)
- Article title (A_Title)
- Article description (A_Description)

Service Plan

- Service plan id (Service_ID)
- Service plan name (SP_Name)

Feedback

- Feedback id (FID)
- Feedback type (F_Type)
- Feedback description (F_Description)

Department

- Department id (Department_ID)
- Managr id (MID)
- Department name (D_Name)

Support_Agent

- Agent id (Agent_ID)
- Department id (Department_ID)
- Artical id (ArticalID)
- Agent first name (A_Fname)
- Agent last name (A_Lname)
- Agent salary (Salary)

Check_Feedback

- Agent id (Agent_ID)
- Feedback id (FID)

Tele_No

- Agent id (Agent_ID)
- Agent phone number (P_Number)

Registered_User

- User id (UserID)
- User date of birth (DOB)
- Article id (ArticalID)
- Service plan id (Service_ID)

Support_Ticket

- Support ticket id (TicketID)
- Support ticket title (Title)
- Support ticket description (Description)
- Support ticket status (Status)
- User id (UserID)
- Agent id (Agent_ID)

SystemAdministrator

- Admin id (AdminID)
- Admin name (Adname)
- Article id (ArticalID)

Report

- Report id (ReportID)
- Report title (Report_title)
- Manager id (M_ID)
- Admin id (AdminID)

AdminEmail

- Admin id (AdminID)
- Admin email (Email)

Register_Contact

- User id (UserID)
- Agent id (Agent_ID)

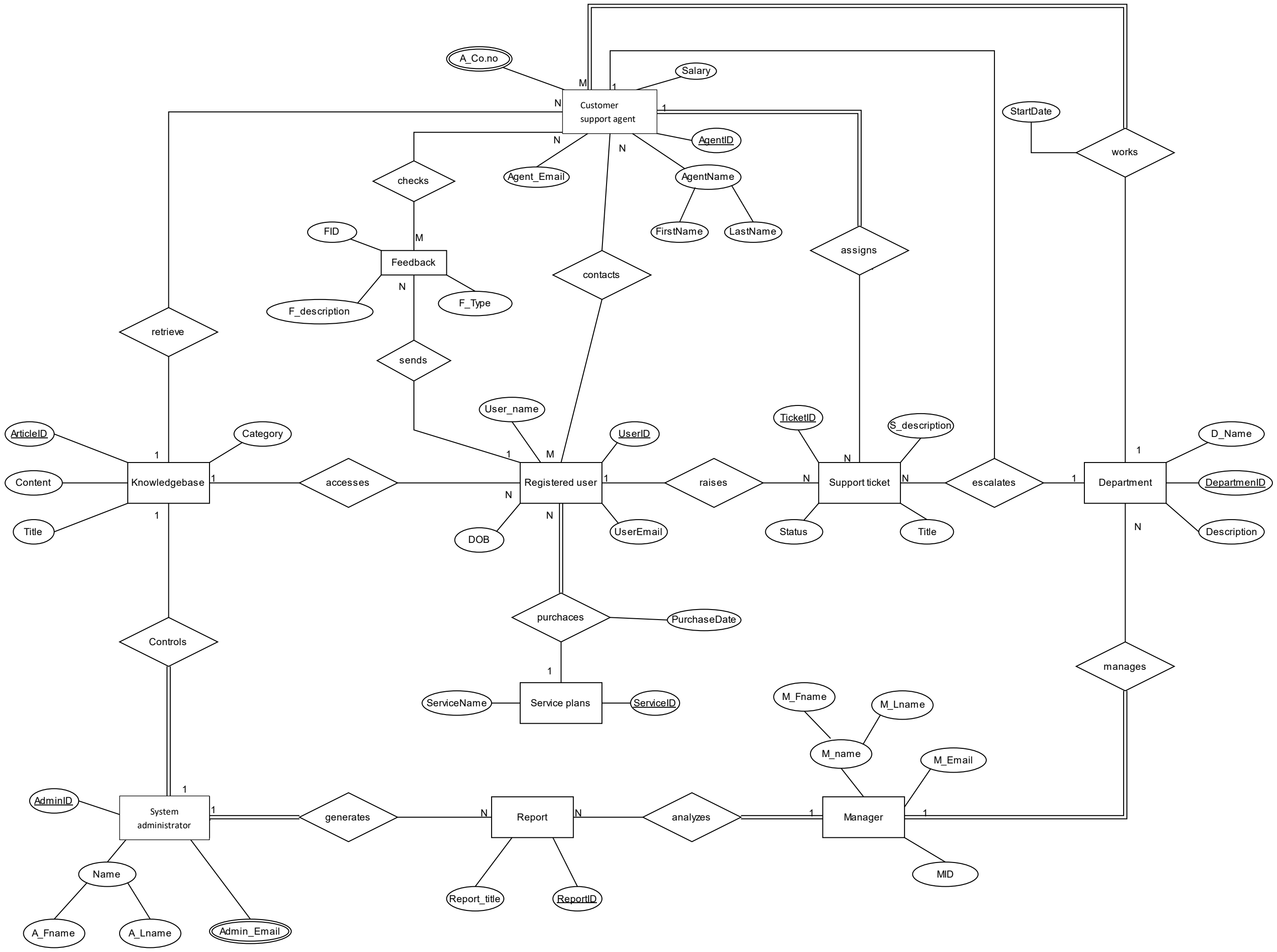
Agent_Worksin

- Agent id (AgentID)
- Department id (DepartmentID)
- Agent stat working date (StartDate)

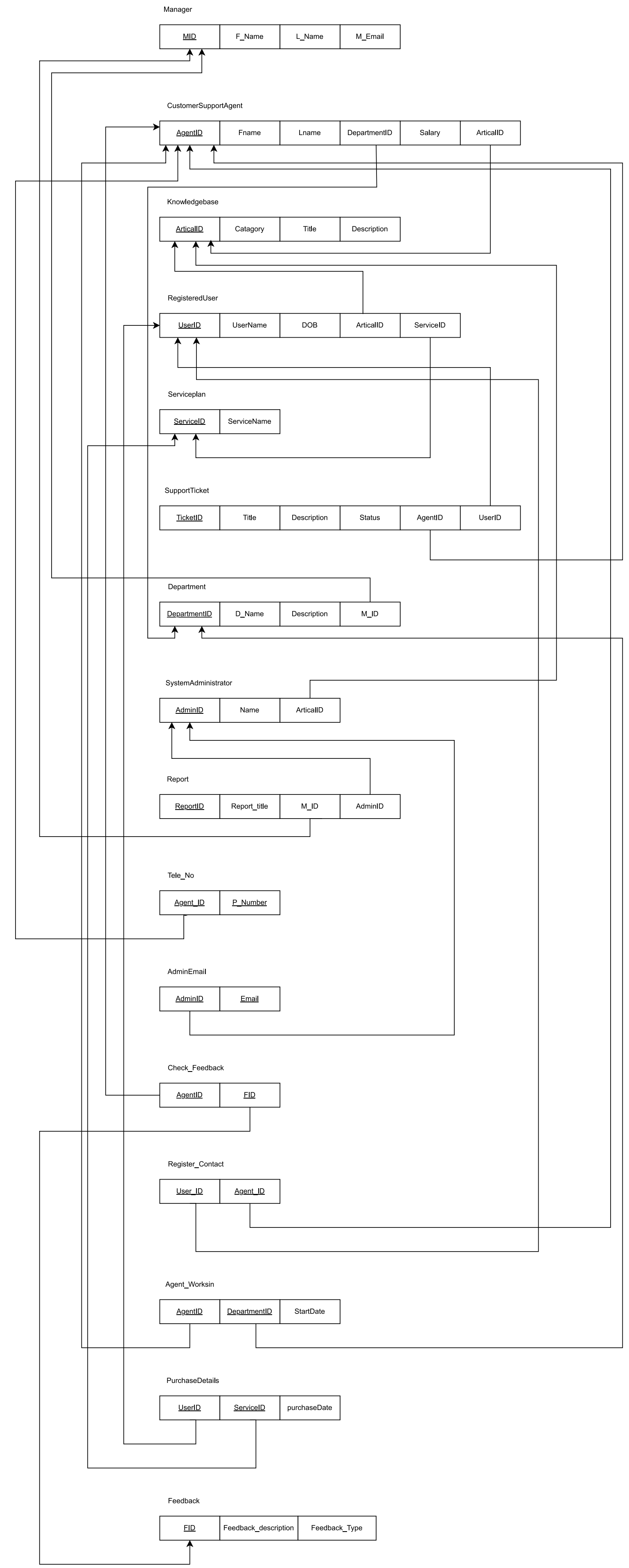
PurchaseDetails

- User id (UserID)
- Service plan id (ServiceID)
- Service plan purchase date (purchaseDate)

Entity Relationship Diagram



Relational Schema



Manager

MID	F_Name	L_Name	M_Email

CustomerSupportAgent

AgentID	Fname	Lname	DepartmentID	Salary	ArticalID

Knowledgebase

ArticalID	Catagory	Title	Description

RegisteredUser

UserID	UserName	DOB	ArticalID	ServiceID

Serviceplan

ServiceID	ServiceName

SupportTicket

TicketID	Title	Description	Status	AgentID	UserID

Department

DepartmentID	D_Name	Description	M_ID

SystemAdministrator

AdminID	Name	ArticalID

Report

ReportID	Report_title	M_ID	AdminID

Tele_No

Agent_ID	P_Number

AdminEmail

AdminID	Email

Check_Feedback

AgentID	FID

Register_Contact

User_ID	Agent_ID

Agent_Worksin

AgentID	DepartmentID	StartDate

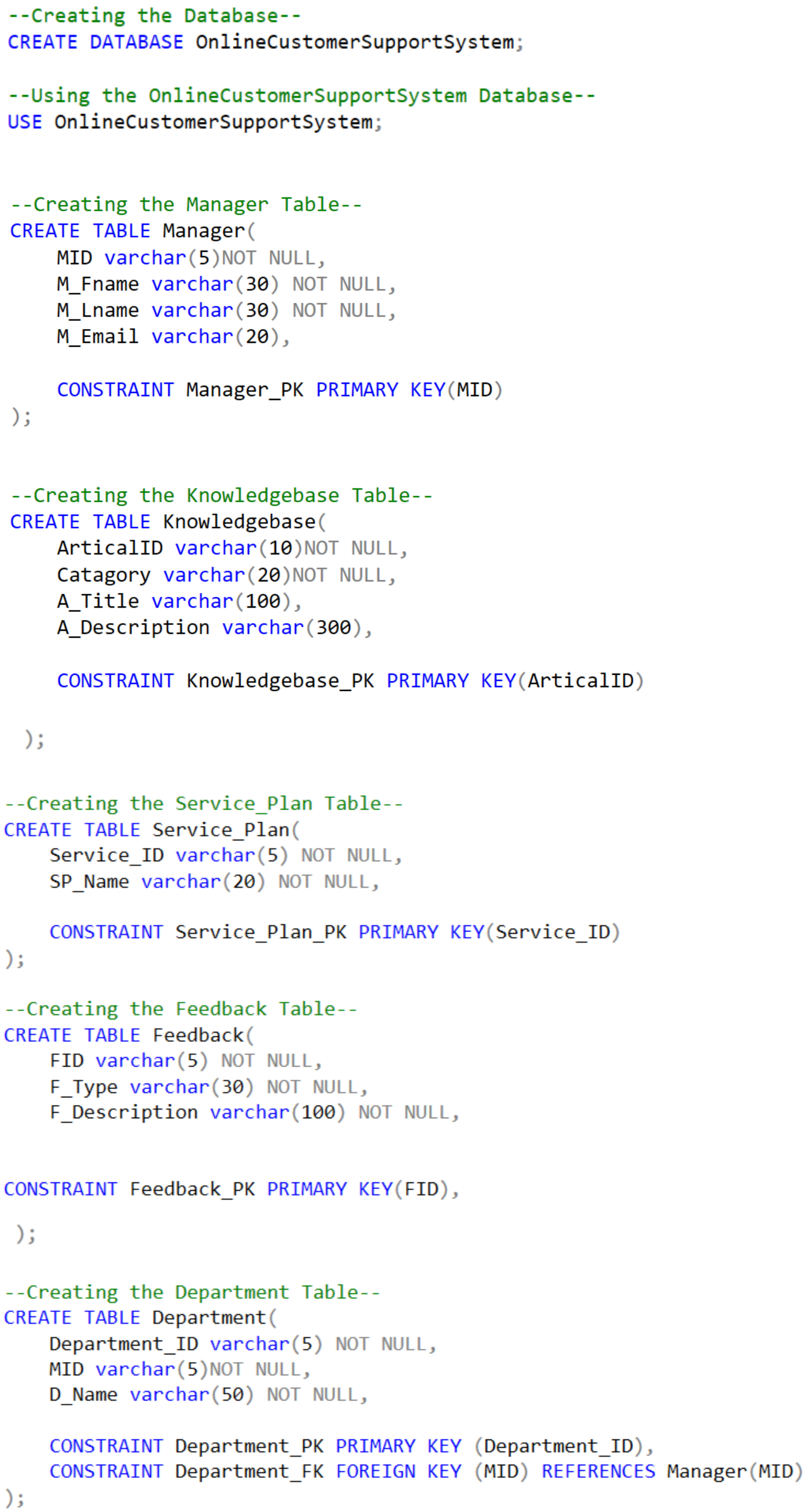
PurchaseDetails

UserID	ServiceID	purchaseDate

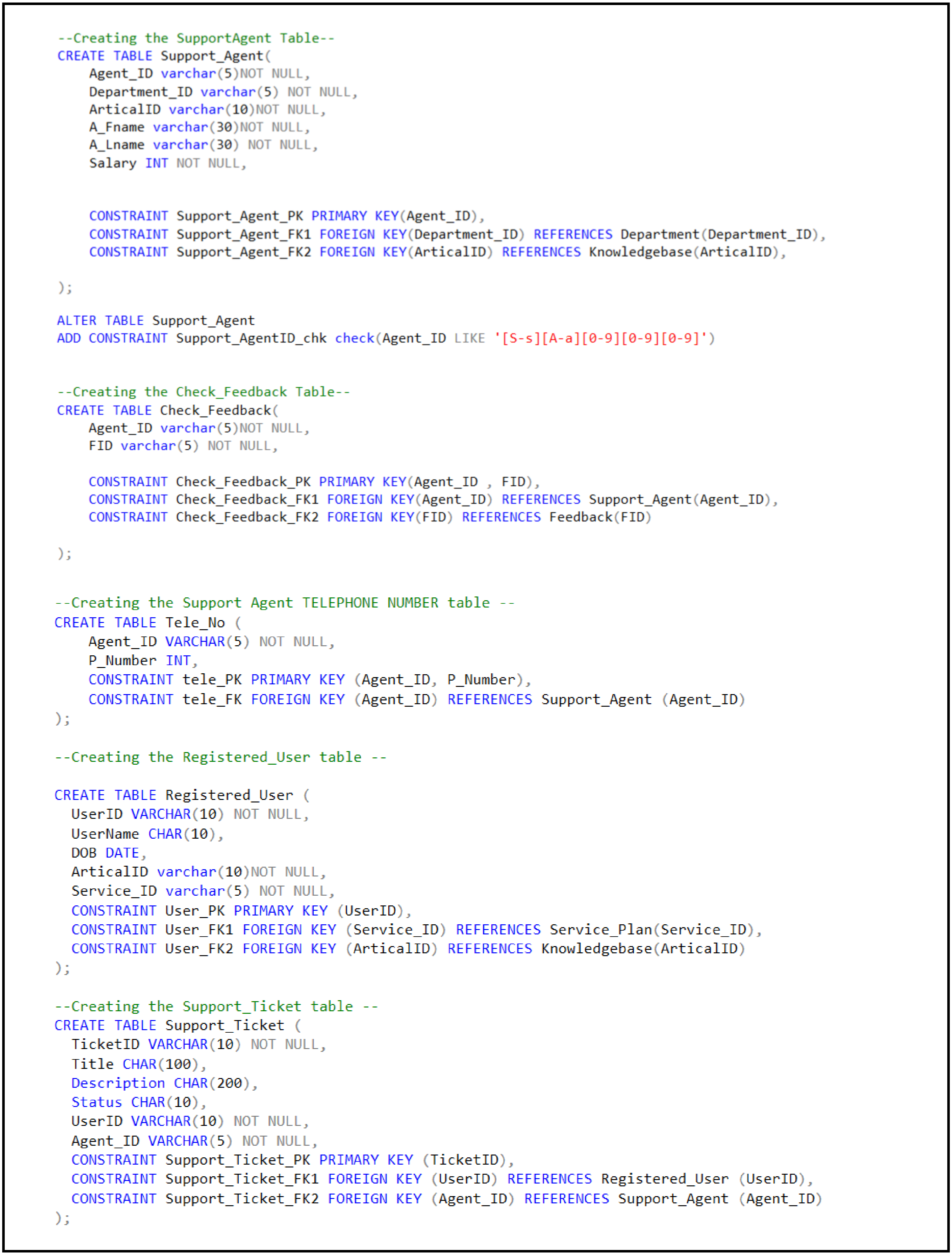
Feedback

FID	Feedback_description	Feedback_Type

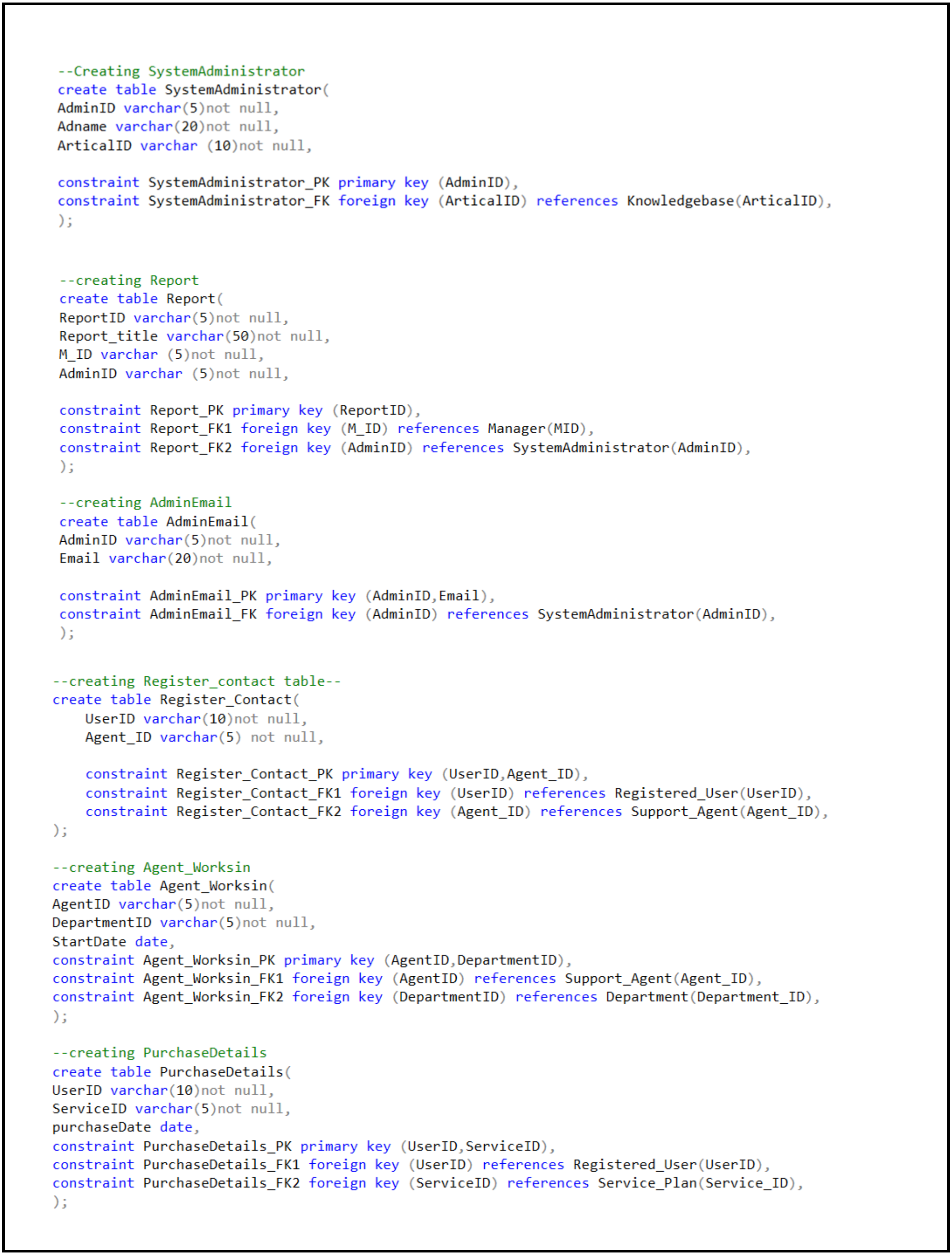
Database Script



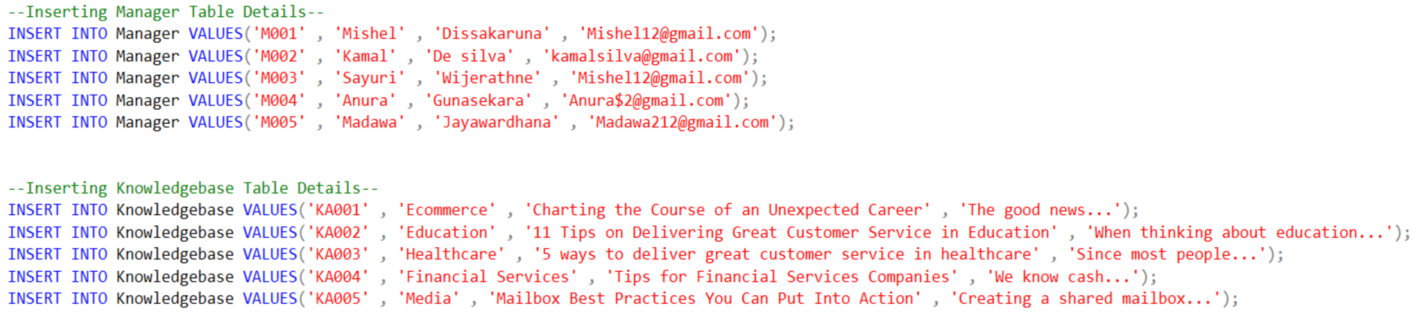
```
--Creating the Database--
CREATE DATABASE OnlineCustomerSupportSystem;

--Using the OnlineCustomerSupportSystem Database--
USE OnlineCustomerSupportSystem;


--Creating the Manager Table--
CREATE TABLE Manager(
    MID varchar(5)NOT NULL,
    M_Fname varchar(30) NOT NULL,
    M_Lname varchar(30) NOT NULL,
    M_Email varchar(20),

    CONSTRAINT Manager_PK PRIMARY KEY(MID)
);


--Creating the Knowledgebase Table--
CREATE TABLE Knowledgebase(
    ArticalID varchar(10)NOT NULL,
    Catagory varchar(20)NOT NULL,
    A_Title varchar(100),
    A_Description varchar(300),

    CONSTRAINT Knowledgebase_PK PRIMARY KEY(ArticalID)

 );

--Creating the Service_Plan Table--
CREATE TABLE Service_Plan(
    Service_ID varchar(5) NOT NULL,
    SP_Name varchar(20) NOT NULL,

    CONSTRAINT Service_Plan_PK PRIMARY KEY(Service_ID)
);

--Creating the Feedback Table--
CREATE TABLE Feedback(
    FID varchar(5) NOT NULL,
    F_Type varchar(30) NOT NULL,
    F_Description varchar(100) NOT NULL,


CONSTRAINT Feedback_PK PRIMARY KEY(FID),

 );

--Creating the Department Table--
CREATE TABLE Department(
    Department_ID varchar(5) NOT NULL,
    MID varchar(5)NOT NULL,
    D_Name varchar(50) NOT NULL,

    CONSTRAINT Department_PK PRIMARY KEY (Department_ID),
    CONSTRAINT Department_FK FOREIGN KEY (MID) REFERENCES Manager(MID)
);
```



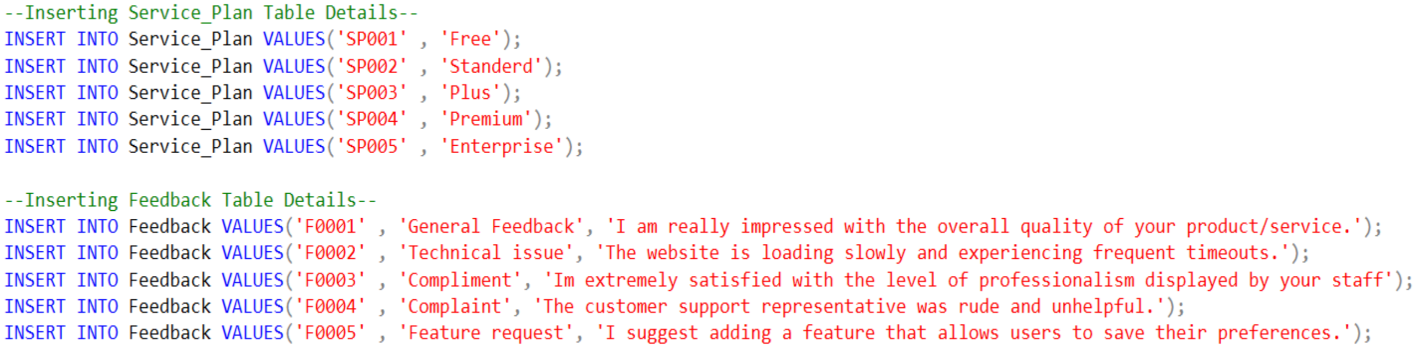
```sql
--Creating the SupportAgent Table--
CREATE TABLE Support_Agent(
    Agent_ID varchar(5)NOT NULL,
    Department_ID varchar(5) NOT NULL,
    ArticalID varchar(10)NOT NULL,
    A_Fname varchar(30)NOT NULL,
    A_Lname varchar(30) NOT NULL,
    Salary INT NOT NULL,


    CONSTRAINT Support_Agent_PK PRIMARY KEY(Agent_ID),
    CONSTRAINT Support_Agent_FK1 FOREIGN KEY(Department_ID) REFERENCES Department(Department_ID),
    CONSTRAINT Support_Agent_FK2 FOREIGN KEY(ArticalID) REFERENCES Knowledgebase(ArticalID),

);

ALTER TABLE Support_Agent
ADD CONSTRAINT Support_AgentID_chk check(Agent_ID LIKE '[S-s][A-a][0-9][0-9][0-9]')


--Creating the Check_Feedback Table--
CREATE TABLE Check_Feedback(
    Agent_ID varchar(5)NOT NULL,
    FID varchar(5) NOT NULL,

    CONSTRAINT Check_Feedback_PK PRIMARY KEY(Agent_ID , FID),
    CONSTRAINT Check_Feedback_FK1 FOREIGN KEY(Agent_ID) REFERENCES Support_Agent(Agent_ID),
    CONSTRAINT Check_Feedback_FK2 FOREIGN KEY(FID) REFERENCES Feedback(FID)

);


--Creating the Support Agent TELEPHONE NUMBER table --
CREATE TABLE Tele_No (
    Agent_ID VARCHAR(5) NOT NULL,
    P_Number INT,
    CONSTRAINT tele_PK PRIMARY KEY (Agent_ID, P_Number),
    CONSTRAINT tele_FK FOREIGN KEY (Agent_ID) REFERENCES Support_Agent (Agent_ID)
);

--Creating the Registered_User table --

CREATE TABLE Registered_User (
  UserID VARCHAR(10) NOT NULL,
  UserName CHAR(10),
  DOB DATE,
  ArticalID varchar(10)NOT NULL,
  Service_ID varchar(5) NOT NULL,
  CONSTRAINT User_PK PRIMARY KEY (UserID),
  CONSTRAINT User_FK1 FOREIGN KEY (Service_ID) REFERENCES Service_Plan(Service_ID),
  CONSTRAINT User_FK2 FOREIGN KEY (ArticalID) REFERENCES Knowledgebase(ArticalID)
);

--Creating the Support_Ticket table --
CREATE TABLE Support_Ticket (
  TicketID VARCHAR(10) NOT NULL,
  Title CHAR(100),
  Description CHAR(200),
  Status CHAR(10),
  UserID VARCHAR(10) NOT NULL,
  Agent_ID VARCHAR(5) NOT NULL,
  CONSTRAINT Support_Ticket_PK PRIMARY KEY (TicketID),
  CONSTRAINT Support_Ticket_FK1 FOREIGN KEY (UserID) REFERENCES Registered_User (UserID),
  CONSTRAINT Support_Ticket_FK2 FOREIGN KEY (Agent_ID) REFERENCES Support_Agent (Agent_ID)
);
```



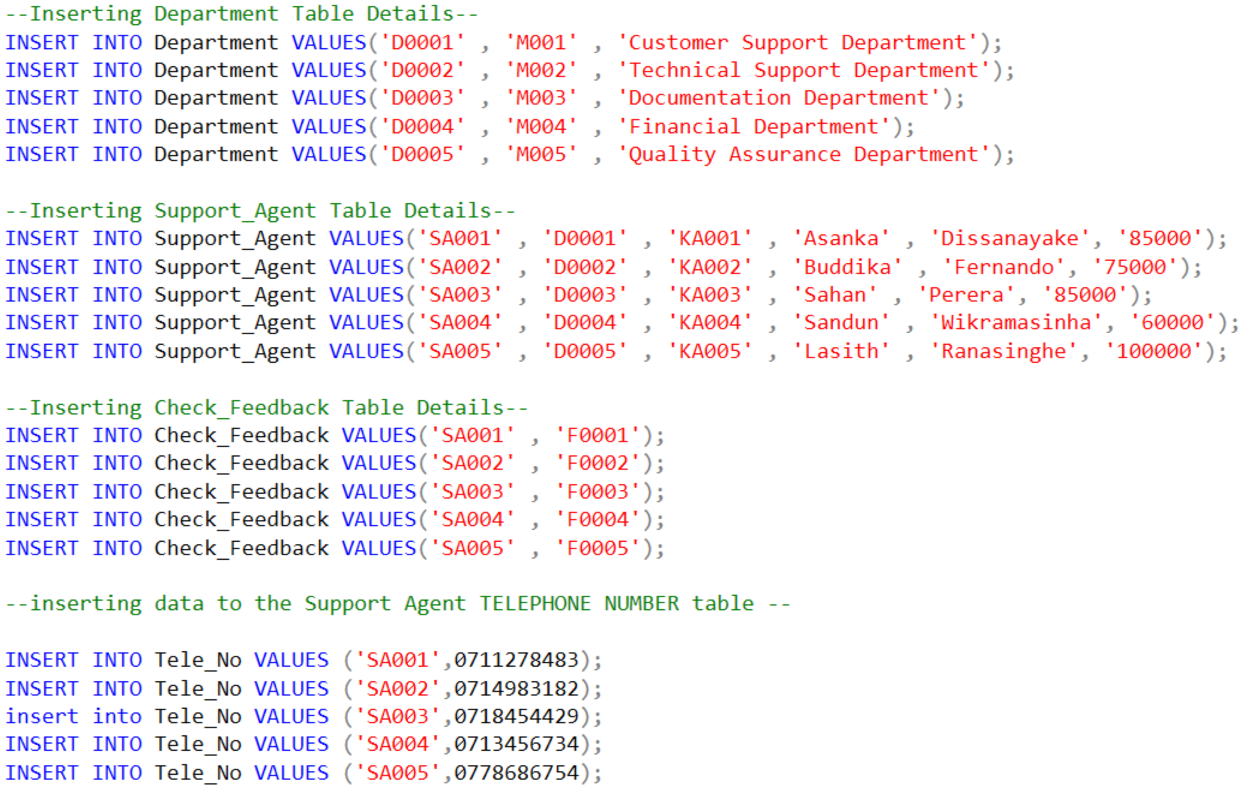
```
--Creating SystemAdministrator
create table SystemAdministrator(
AdminID varchar(5)not null,
Adname varchar(20)not null,
ArticalID varchar (10)not null,

constraint SystemAdministrator_PK primary key (AdminID),
constraint SystemAdministrator_FK foreign key (ArticalID) references Knowledgebase(ArticalID),
);


--creating Report
create table Report(
ReportID varchar(5)not null,
Report_title varchar(50)not null,
M_ID varchar (5)not null,
AdminID varchar (5)not null,

constraint Report_PK primary key (ReportID),
constraint Report_FK1 foreign key (M_ID) references Manager(MID),
constraint Report_FK2 foreign key (AdminID) references SystemAdministrator(AdminID),
);

--creating AdminEmail
create table AdminEmail(
AdminID varchar(5)not null,
Email varchar(20)not null,

constraint AdminEmail_PK primary key (AdminID,Email),
constraint AdminEmail_FK foreign key (AdminID) references SystemAdministrator(AdminID),
);


--creating Register_contact table--
create table Register_Contact(
    UserID varchar(10)not null,
    Agent_ID varchar(5) not null,

    constraint Register_Contact_PK primary key (UserID,Agent_ID),
    constraint Register_Contact_FK1 foreign key (UserID) references Registered_User(UserID),
    constraint Register_Contact_FK2 foreign key (Agent_ID) references Support_Agent(Agent_ID),
);

--creating Agent_Worksin
create table Agent_Worksin(
AgentID varchar(5)not null,
DepartmentID varchar(5)not null,
StartDate date,
constraint Agent_Worksin_PK primary key (AgentID,DepartmentID),
constraint Agent_Worksin_FK1 foreign key (AgentID) references Support_Agent(Agent_ID),
constraint Agent_Worksin_FK2 foreign key (DepartmentID) references Department(Department_ID),
);

--creating PurchaseDetails
create table PurchaseDetails(
UserID varchar(10)not null,
ServiceID varchar(5)not null,
purchaseDate date,
constraint PurchaseDetails_PK primary key (UserID,ServiceID),
constraint PurchaseDetails_FK1 foreign key (UserID) references Registered_User(UserID),
constraint PurchaseDetails_FK2 foreign key (ServiceID) references Service_Plan(Service_ID),
);
```



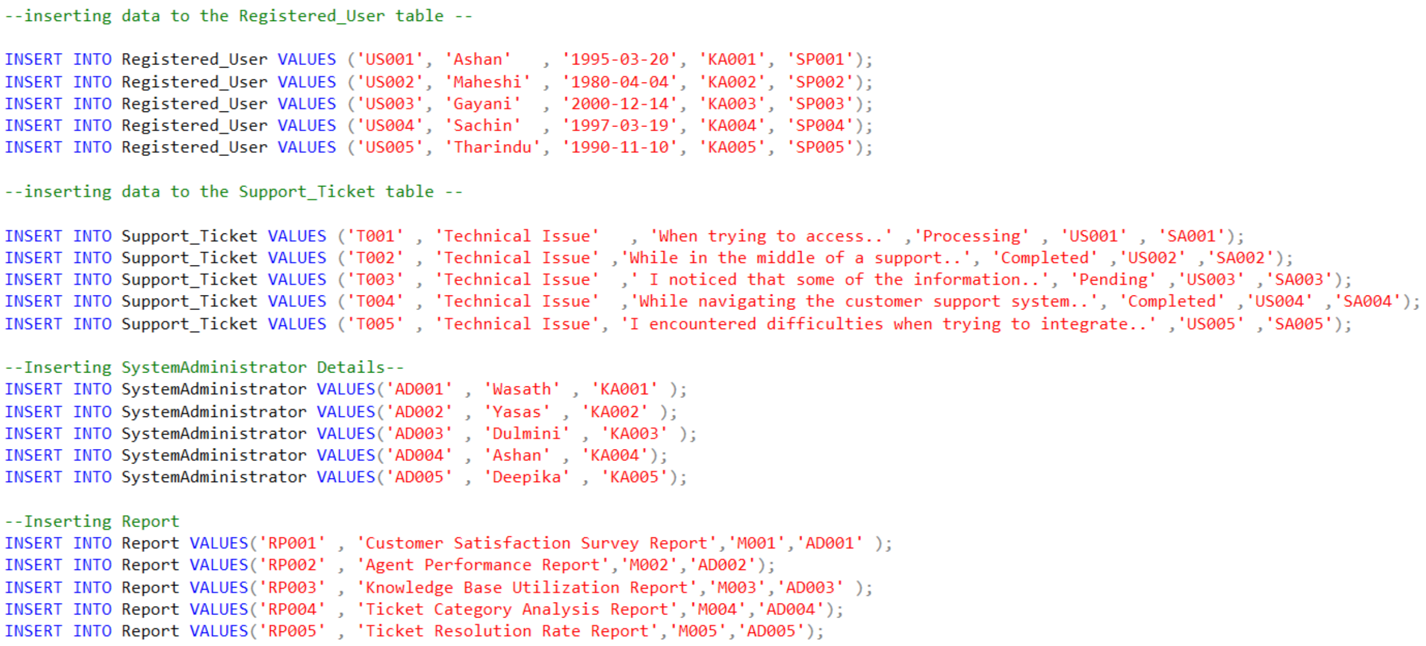
```
--Inserting Manager Table Details--
INSERT INTO Manager VALUES('M001' , 'Mishel' , 'Dissakaruna' , 'Mishel12@gmail.com');
INSERT INTO Manager VALUES('M002' , 'Kamal' , 'De silva' , 'kamalsilva@gmail.com');
INSERT INTO Manager VALUES('M003' , 'Sayuri' , 'Wijerathne' , 'Mishel12@gmail.com');
INSERT INTO Manager VALUES('M004' , 'Anura' , 'Gunasekara' , 'Anura$2@gmail.com');
INSERT INTO Manager VALUES('M005' , 'Madawa' , 'Jayawardhana' , 'Madawa212@gmail.com');


--Inserting Knowledgebase Table Details--
INSERT INTO Knowledgebase VALUES('KA001' , 'Ecommerce' , 'Charting the Course of an Unexpected Career' , 'The good news...');
INSERT INTO Knowledgebase VALUES('KA002' , 'Education' , '11 Tips on Delivering Great Customer Service in Education' , 'When thinking about education...');
INSERT INTO Knowledgebase VALUES('KA003' , 'Healthcare' , '5 ways to deliver great customer service in healthcare' , 'Since most people...');
INSERT INTO Knowledgebase VALUES('KA004' , 'Financial Services' , 'Tips for Financial Services Companies' , 'We know cash...');
INSERT INTO Knowledgebase VALUES('KA005' , 'Media' , 'Mailbox Best Practices You Can Put Into Action' , 'Creating a shared mailbox...');
```



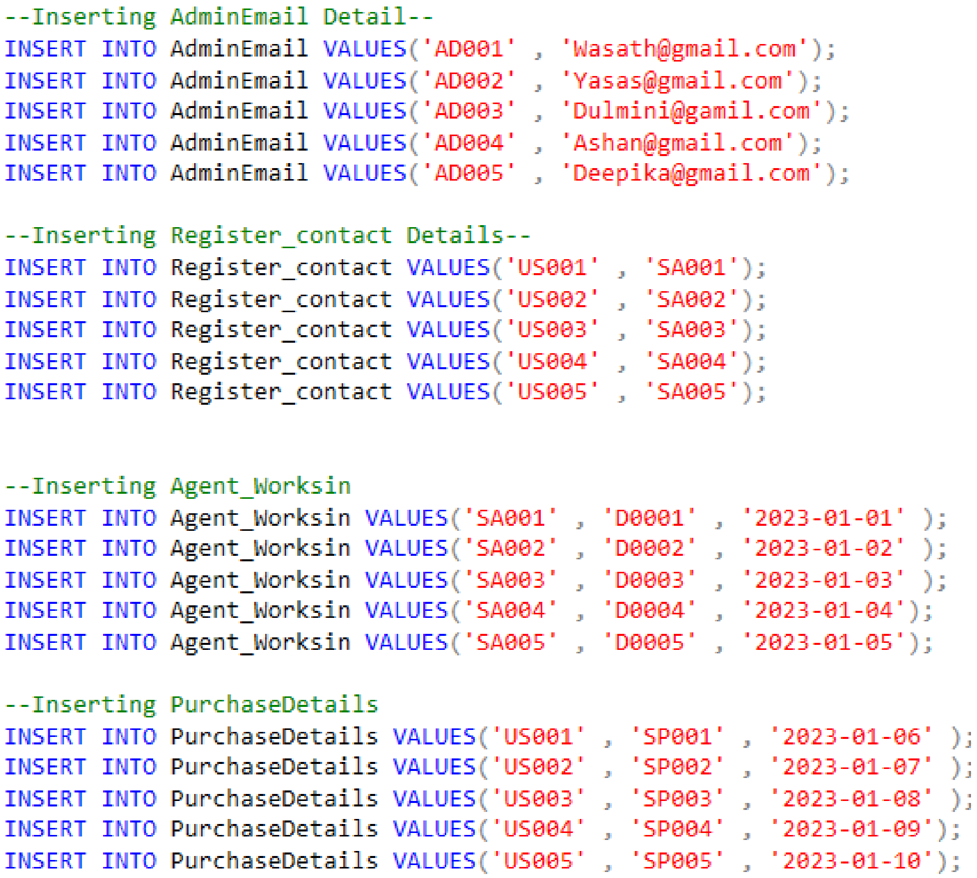
```
--Inserting Service_Plan Table Details--
INSERT INTO Service_Plan VALUES('SP001' , 'Free');
INSERT INTO Service_Plan VALUES('SP002' , 'Standerd');
INSERT INTO Service_Plan VALUES('SP003' , 'Plus');
INSERT INTO Service_Plan VALUES('SP004' , 'Premium');
INSERT INTO Service_Plan VALUES('SP005' , 'Enterprise');

--Inserting Feedback Table Details--
INSERT INTO Feedback VALUES('F0001' , 'General Feedback', 'I am really impressed with the overall quality of your product/service.');
INSERT INTO Feedback VALUES('F0002' , 'Technical issue', 'The website is loading slowly and experiencing frequent timeouts.');
INSERT INTO Feedback VALUES('F0003' , 'Compliment', 'Im extremely satisfied with the level of professionalism displayed by your staff');
INSERT INTO Feedback VALUES('F0004' , 'Complaint', 'The customer support representative was rude and unhelpful.');
INSERT INTO Feedback VALUES('F0005' , 'Feature request', 'I suggest adding a feature that allows users to save their preferences.');
```



```
--Inserting Department Table Details--
INSERT INTO Department VALUES('D0001' , 'M001' , 'Customer Support Department');
INSERT INTO Department VALUES('D0002' , 'M002' , 'Technical Support Department');
INSERT INTO Department VALUES('D0003' , 'M003' , 'Documentation Department');
INSERT INTO Department VALUES('D0004' , 'M004' , 'Financial Department');
INSERT INTO Department VALUES('D0005' , 'M005' , 'Quality Assurance Department');

--Inserting Support_Agent Table Details--
INSERT INTO Support_Agent VALUES('SA001' , 'D0001' , 'KA001' , 'Asanka' , 'Dissanayake', '85000');
INSERT INTO Support_Agent VALUES('SA002' , 'D0002' , 'KA002' , 'Buddika' , 'Fernando', '75000');
INSERT INTO Support_Agent VALUES('SA003' , 'D0003' , 'KA003' , 'Sahan' , 'Perera', '85000');
INSERT INTO Support_Agent VALUES('SA004' , 'D0004' , 'KA004' , 'Sandun' , 'Wikramasinha', '60000');
INSERT INTO Support_Agent VALUES('SA005' , 'D0005' , 'KA005' , 'Lasith' , 'Ranasinghe', '100000');

--Inserting Check_Feedback Table Details--
INSERT INTO Check_Feedback VALUES('SA001' , 'F0001');
INSERT INTO Check_Feedback VALUES('SA002' , 'F0002');
INSERT INTO Check_Feedback VALUES('SA003' , 'F0003');
INSERT INTO Check_Feedback VALUES('SA004' , 'F0004');
INSERT INTO Check_Feedback VALUES('SA005' , 'F0005');

--inserting data to the Support Agent TELEPHONE NUMBER table --

INSERT INTO Tele_No VALUES ('SA001',0711278483);
INSERT INTO Tele_No VALUES ('SA002',0714983182);
insert into Tele_No VALUES ('SA003',0718454429);
INSERT INTO Tele_No VALUES ('SA004',0713456734);
INSERT INTO Tele_No VALUES ('SA005',0778686754);
```



```sql
--inserting data to the Registered_User table --

INSERT INTO Registered_User VALUES ('US001', 'Ashan'   , '1995-03-20', 'KA001', 'SP001');
INSERT INTO Registered_User VALUES ('US002', 'Maheshi' , '1980-04-04', 'KA002', 'SP002');
INSERT INTO Registered_User VALUES ('US003', 'Gayani'  , '2000-12-14', 'KA003', 'SP003');
INSERT INTO Registered_User VALUES ('US004', 'Sachin'  , '1997-03-19', 'KA004', 'SP004');
INSERT INTO Registered_User VALUES ('US005', 'Tharindu', '1990-11-10', 'KA005', 'SP005');

--inserting data to the Support_Ticket table --

INSERT INTO Support_Ticket VALUES ('T001' , 'Technical Issue'   , 'When trying to access..' ,'Processing' , 'US001' , 'SA001');
INSERT INTO Support_Ticket VALUES ('T002' , 'Technical Issue' ,'While in the middle of a support..', 'Completed' ,'US002' ,'SA002');
INSERT INTO Support_Ticket VALUES ('T003' , 'Technical Issue'  ,' I noticed that some of the information..', 'Pending' ,'US003' ,'SA003');
INSERT INTO Support_Ticket VALUES ('T004' , 'Technical Issue'  ,'While navigating the customer support system..', 'Completed' ,'US004' ,'SA004');
INSERT INTO Support_Ticket VALUES ('T005' , 'Technical Issue', 'I encountered difficulties when trying to integrate..' ,'US005' ,'SA005');

--Inserting SystemAdministrator Details--
INSERT INTO SystemAdministrator VALUES('AD001' , 'Wasath' , 'KA001' );
INSERT INTO SystemAdministrator VALUES('AD002' , 'Yasas' , 'KA002' );
INSERT INTO SystemAdministrator VALUES('AD003' , 'Dulmini' , 'KA003' );
INSERT INTO SystemAdministrator VALUES('AD004' , 'Ashan' , 'KA004');
INSERT INTO SystemAdministrator VALUES('AD005' , 'Deepika' , 'KA005');

--Inserting Report
INSERT INTO Report VALUES('RP001' , 'Customer Satisfaction Survey Report','M001','AD001' );
INSERT INTO Report VALUES('RP002' , 'Agent Performance Report','M002','AD002');
INSERT INTO Report VALUES('RP003' , 'Knowledge Base Utilization Report','M003','AD003' );
INSERT INTO Report VALUES('RP004' , 'Ticket Category Analysis Report','M004','AD004');
INSERT INTO Report VALUES('RP005' , 'Ticket Resolution Rate Report','M005','AD005');
```



```sql
--Inserting AdminEmail Detail--
INSERT INTO AdminEmail VALUES('AD001' , 'Wasath@gmail.com');
INSERT INTO AdminEmail VALUES('AD002' , 'Yasas@gmail.com');
INSERT INTO AdminEmail VALUES('AD003' , 'Dulmini@gamil.com');
INSERT INTO AdminEmail VALUES('AD004' , 'Ashan@gmail.com');
INSERT INTO AdminEmail VALUES('AD005' , 'Deepika@gmail.com');

--Inserting Register_contact Details--
INSERT INTO Register_contact VALUES('US001' , 'SA001');
INSERT INTO Register_contact VALUES('US002' , 'SA002');
INSERT INTO Register_contact VALUES('US003' , 'SA003');
INSERT INTO Register_contact VALUES('US004' , 'SA004');
INSERT INTO Register_contact VALUES('US005' , 'SA005');


--Inserting Agent_Worksin
INSERT INTO Agent_Worksin VALUES('SA001' , 'D0001' , '2023-01-01' );
INSERT INTO Agent_Worksin VALUES('SA002' , 'D0002' , '2023-01-02' );
INSERT INTO Agent_Worksin VALUES('SA003' , 'D0003' , '2023-01-03' );
INSERT INTO Agent_Worksin VALUES('SA004' , 'D0004' , '2023-01-04');
INSERT INTO Agent_Worksin VALUES('SA005' , 'D0005' , '2023-01-05');

--Inserting PurchaseDetails
INSERT INTO PurchaseDetails VALUES('US001' , 'SP001' , '2023-01-06' );
INSERT INTO PurchaseDetails VALUES('US002' , 'SP002' , '2023-01-07' );
INSERT INTO PurchaseDetails VALUES('US003' , 'SP003' , '2023-01-08' );
INSERT INTO PurchaseDetails VALUES('US004' , 'SP004' , '2023-01-09');
INSERT INTO PurchaseDetails VALUES('US005' , 'SP005' , '2023-01-10');
```

Performance Considerations:

- The system should be highly available and accessible to users whenever they need support.
- User interface should be responsive, smooth and user friendly.
- The system should be able to handle the increasing user traffic without any performance breakdown.
- The system should quickly response to the customer support queries and tickets. Such as within 12 hours or less.
- Reports should be generated at a scheduled time.
- Users can access the system anywhere with suitable hardware devices.
- Login details verification should be done within a few seconds by the system.
- In real time communication with customer support agents, the system should have low latency and fast response time.
- Support tickets assigned quickly according to the availability of support agents.
- The system quickly sends the confirmation mail to user for assigning ticket in few seconds.
- The system automatically updates raised ticket progress and user can check the progress without any delay.

Security Requirements:

- User account username should be unique, and password must be strong and valid.
- Sensitive user data such as login credentials and personal information should be encrypted.
- System should use secure authentication methods such as two-factor authentication.
- Only admin has access to the admin dashboard of system.
- System should have a backup and recovery option to avoid data losses,
- Developer should give regularly updates and patches for the system to avoid security breaches.
- The privacy of the user raised tickets or inquiries should be encrypted by the system.
- System should provide secure communication channels to communicate between user and the support agent.
- Payment information should be encrypted by the system.