ER to SQL Translation

Unity IDs:

Saketh Vangala - svangal

Sukruthi Modem - smodem

Sasank Marabattula - smaraba

Varun Deepak Gudhe - vgudhe

```
CREATE TABLE SERVICE_CENTER(
SVC_ID number PRIMARY KEY,
phone_number string,
address string,
Saturday_flag Boolean,
Mech_Hourly_Wages Float,
STATE_ID string,
FOREIGN KEY STATE_ID REFERENCES STATE(STATE_ID) ON DELETE CASCADE
);

CREATE TABLE STATE
(
STATE_ID number PRIMARY KEY,
state_name string
);

CREATE TABLE EMPLOYEE_IN_SVC (
EMP_ID number,
SVC_ID number,
name string,
address string,
email string,
phone_no string,
role_at_svc string,
PRIMARY KEY (SVC_ID,EMP_ID),
FOREIGN KEY SVC_ID REFERENCES SERVICE_CENTER(SVC_ID) ON DELETE CASCADE
);

CREATE TABLE MANAGER
(
MANAGER_ID number DEFAULT 0000,
SVC_ID number,
Manager_ansal float,
PRIMARY KEY (SVC_ID,MANAGER_ID),
```

FOREIGN KEY SVC_ID REFERENCES SERVICE_CENTER(SVC_ID) ON DELETE CASCADE
FOREIGN KEY MANAGER_ID REFERENCES EMPLOYEE_IN_SVC (EMP_ID) ON DELETE CASCADE
);

CREATE TABLE RECEPTIONIST
(
RECEPTIONIST_ID number,
SVC_ID number,
MANAGER_ID number,
receptionist_ansal float,
PRIMARY KEY (SVC_ID,RECEPTIONIST_ID),
FOREIGN KEY SVC_ID REFERENCES SERVICE_CENTER(SVC_ID) ON DELETE CASCADE,
FOREIGN KEY MANAGER_ID REFERENCES MANAGER(MANAGER_ID) ON DELETE SET DEFAULT
0000,
FOREIGN KEY RECEPTIONIST_ID REFERENCES EMPLOYEE_IN_SVC (EMP_ID) ON DELETE CASCADE
);

CREATE TABLE MECHANIC
(
MECHANIC_ID number,
SVC_ID number,
MANAGER_ID number,
name string,
address string,
email string,
phone_no string,
receptionist_ansal float,
role_at_svc string,
hours_worked float,
time_off Boolean,
PRIMARY KEY (SVC_ID,MECHANIC_ID),
FOREIGN KEY SVC_ID REFERENCES SERVICE_CENTER(SVC_ID) ON DELETE CASCADE,
FOREIGN KEY MANAGER_ID REFERENCES MANAGER(MANAGER_ID) ON DELETE SET DEFAULT
0000,
FOREIGN KEY MECHANIC_ID REFERENCES EMPLOYEE_IN_SVC (EMP_ID) ON DELETE CASCADE
);

```
CREATE TABLE SERVICES
(
SERVICE_ID number PRIMARY KEY,
name string,
Est_time float
);

CREATE TABLE SERVICES_OFFERS
(
SERVICE_ID number,
SVC_ID number,
price float,
car_manf string,
PRIMARY KEY (SERVICE_ID,SVC_ID),
FOREIGN KEY SVC_ID REFERENCES SERVICE_CENTER(SVC_ID) ON DELETE NO ACTION ,
FOREIGN KEY SERVICE_ID REFERENCES SERVICE_ID(SERVICE_ID) ON DELETE NO ACTION
);

CREATE REPAIR_SERVICES
(
SERVICE_ID number PRIMARY KEY,
subcategory string,
individual_service string
FOREIGN KEY SERVICE_ID REFERENCES SERVICES(SERVICE_ID) ON DELETE CASCADE
);

CREATE TABLE MAINTENANCE_SERVICES
(
SERVICE_ID number PRIMARY KEY,
schedule_or_not boolean,
FOREIGN KEY SERVICE_ID REFERENCES SERVICES(SERVICE_ID) ON DELETE CASCADE
);

CREATE TABLE SCHEDULES
(
SERVICE_ID number PRIMARY KEY,
schedule_name string,
```

```sql
next_schedule string,
FOREIGN KEY SERVICE_ID REFERENCES MAINTENANCE_SERVICES(SERVICE_ID) ON DELETE
CASCADE
);


CREATE TABLE CUSTOMER_HAS_VEHICLE
(
CID number,
VIN number,
SVC_ID number,
first_name string,
last_name string,
status boolean,
total_amount_charged float,
total_amount_paid float,
car_manf string,
c_mil float,
year number,
last_schedule string,
PRIMARY KEY(CID,VIN,SVC_ID),
FOREIGN KEY CID REFERENCES CUSTOMER(CID) ON DELETE CASCADE,
FOREIGN KEY VIN REFERENCES VEHICLE(VIN) ON DELETE CASCADE,
FOREIGN KEY SVC_ID REFERENCES SERVICE_CENTER(SVC_ID) ON DELETE CASCADE
);

CREATE TABLE INVOICE
(
INV_ID number PRIMARY KEY,
CID number,
SVC_ID number,
VIN number,
date_of_service date,
status boolean,
ser_provided string,
ser_cost float,
mechanic string,
tot_cost float,
FOREIGN KEY CID REFERENCES CUSTOMER(CID) ON DELETE CASCADE,
FOREIGN KEY VIN REFERENCES VEHICLE(VIN) ON DELETE CASCADE,
FOREIGN KEY SVC_ID REFERENCES SERVICE_CENTER(SVC_ID) ON DELETE CASCADE
);
```

```
CREATE TABLE SLOTS
(
SID number PRIMARY KEY,
week_day_id number
);

CREATE TABLE WORKS_IN_SLOTS
(
SID number,
MECHANIC_ID number
PRIMARY KEY(sid,MECHANIC_ID ),
FOREIGN KEY SID REFERENCES SLOTS(SID) ON DELETE CASCADE,
FOREIGN KEY MECHANIC_ID REFERENCES MECHANIC(MECHANIC_ID) ON DELETE CASCADE
)
```

Description Document phrase	How Model's captured the constraint
Each service center is identified by a globally unique ID	Added SVC_ID as Primary Key to SERVICE_CENTER entity
Maintenance services are usually done in a rotational manner i.e. after a Schedule A maintenance service, the next maintenance will be Schedule B, then Schedule C. After Schedule C, it restarts at Schedule A.	Added an attribute "next_schedule" that shows the next schedule for "schedule_name" (which stores A,B,C) attributes. Ex: If schedule_name='A', then we show next_schedule='B'
For each employee, we store their locally unique 9-digit employee ID, name, address, email address, phone number, service center ID,	We added EMP_ID as the primary key for employees but as it is dependent on the service center, so the employee table becomes a weak entity and the EMP_ID becomes a partial key. So, now the primary key for employee table will be SVC_ID, EMP_ID
Each employee is associated with only one service center. Each employee can only play one role at a time	Total participation constraint is added from employee to service center. Non overlap constraint is added to ISA for employees table as one employee can do only one role at a time. Total participation constraint is added from service center to manager as every service center has one manager One to one relation is there from service to manager as each manager is associated with only one service center and each service center has only one manager

Individual services have a globally unique number and name.	Added SERVICE_ID as primary key to table services
A *Customer* has *id* (an integer) that is unique with respect to a specific center	Added CID as the primary key to the customer table but as its unique with respect to a specific center, customer is a weak entity and therefore CID is the partial key. So, the primary key for customer table will be SVC_ID, CID
A customer is associated with at least one vehicle which is identified by globally unique *vin number*	Full participation constraint is applied to the customer which is associated with the vehicle and vehicle is a weak entity and also the customer is weak entity. We created one table CUSTOMER_HAS_VEHICLE where the primary key is SVC_ID, CID,VIN. So instead of customer table we made only one table for both custom and vehicle CUSTOMER_HAS_VEHICLE
The general employee structure in each center has a *manager* who manages all employees, a *receptionist,* and several *mechanics*.	As every employee is managed by only one manager. We added a full participation constraint to employee associated with manager and many to one participation from employee to manager
Every *customer* is associated with *one* specific *service store.*	As every customer is associated with one service store, we added full participation constraint to the customer associated with that service store and many to one participation constraint from customer to service store.
Each center operates 5 days a week (M-F) from 8 AM to 8 PM. *Some (not all)* are also open on *Saturdays* from 9am - 1pm.	Added *saturday_flag* attribute in service center entity, which indicates whether the service center is open on saturdays.
Each employee is associated with only one service center. Each employee can only play one role at a time (for example, a mechanic cannot also be a receptionist).	Added *IS A* relationship from employee to receptionist, manager and mechanics with covering and non overlapping constraints.

Each center has its own hourly rate for mechanics.	Added an attribute *mech_hourly wages* to the entity service center.
For each service event, an invoice with unique invoice id.	Added *invoice_id* as a primary key to the entity Invoice.

FUNCTIONAL DEPENDENCIES

SVC_ID in table (SERVICE_CENTER),
SERVICE_ID in table (SERVICES),
CID in table (CUSTOMER_HAS_VEHICLE)

SVC_ID -> mech_hourly_wages
SVC_ID,EMP_ID -> role_at_svc
SERVICE_ID -> name, estimated_time
SVC_ID -> saturday_flag
SVC_ID,EMP_ID -> emp_name
SVC_ID, SERVICE_ID, car_manf -> price
SVC_ID, CID -> status
VIN -> last_schedule
INV_ID -> total_cost