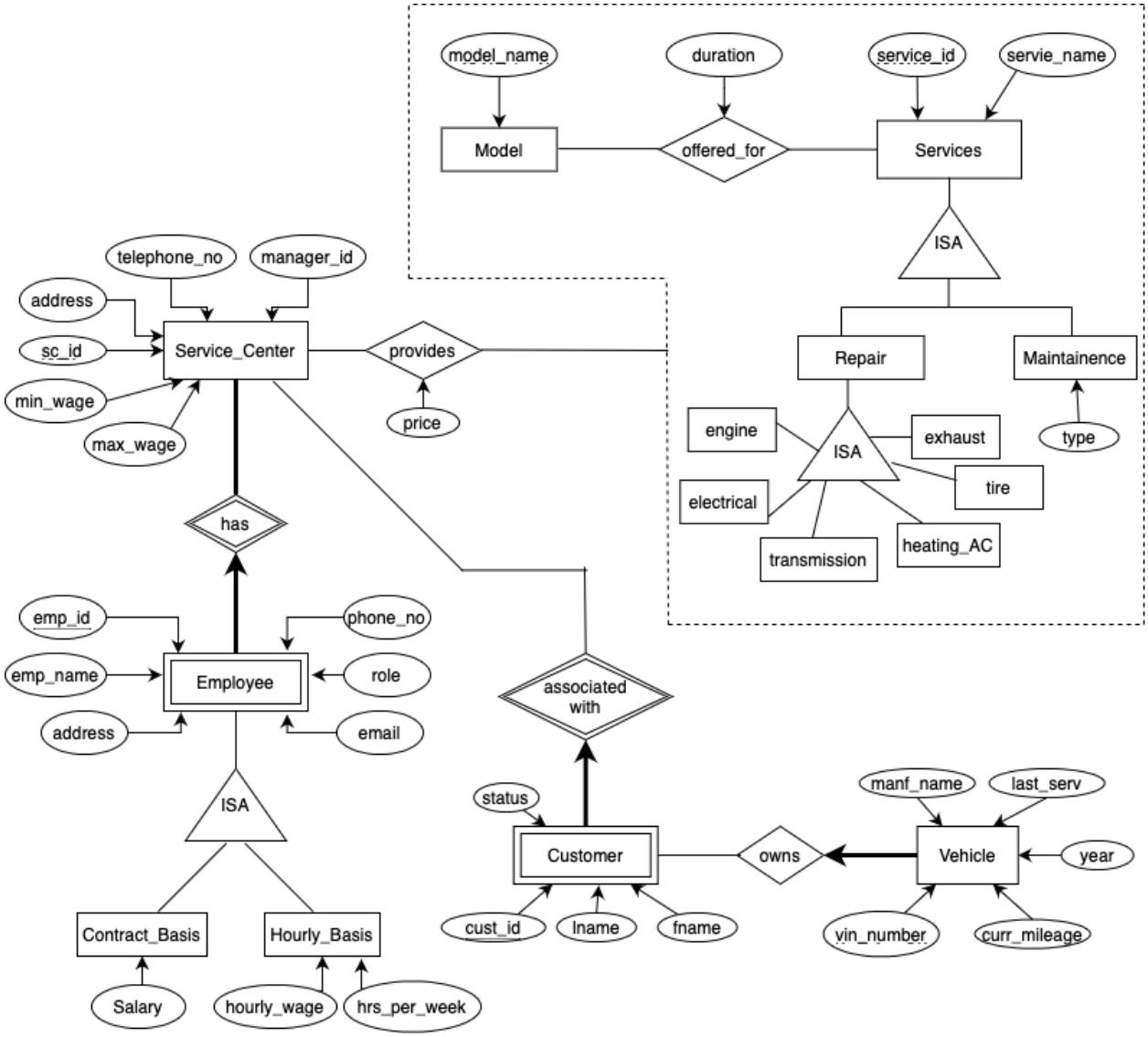
CSC-540 Database Management Concepts and System

Preliminary Report for Project-1

Team Members -

Aditi Salunkhe	Neel Shah	Purv Patel	Sourabh Wattamwar
acsalunk	npshah6	ppatel36	sswattam

1. Partial E-R model:



2. Textual description of partial E-R model:

→ Entity Service Center:

- {sc_id, address, telephone_no, manager_id, min_wage, max_wage}
- The attribute sc_id acts as primary key
- It is the parent entity for Employee entity and is linked to it by has relationship
- Each Service Center has 1 or more employees however each employee has one service center
- It is the parent entity for Customer entity and is linked to it by associated with relationship
- Each Service Center has 0 or more customers however each customer has one service center
- It has an aggregation relationship with offer relationship which is linked by provides relationship.

→ Weak Entity Employee

- {emp_id, emp_name, address, phone_no, role, email}
- It is a weak entity whose parent entity is Service Center and linked by has relation.
- The attribute emp_id is locally unique and together with sc_id they form the primary key
- As it's a weak entity all employees are associated with exactly one Service Center which is indicated by the arrow from Employee to has relationship.
- All employees are classified into sub categories - Contract_Basis and Hourly_Basis. This is a total covering and non-overlapping ISA relationship.
- Contract_Basis has a salary attribute which denotes their yearly salary while Hourly_Basis has hourly_wage and hrs_per_week as attributes which denotes their hourly wage and number of hours worked per week respectively.

→ Entity Vehicle

- {vin_number, curr_mileage, year, last_serv, manf_name}
- The attribute vin_number acts as primary key
- It is linked with Customer entity by the relation owns.
- Each Vehicle is owned by exactly one customer which is denoted by the bold arrow from vehicle to owns relationship.

→ Weak Entity Customer

- {cust_id, lname, fname, status}
- It is a weak entity whose parent entity is Service Center and linked by associated with relation.
- The attribute cust_id is locally unique and together with sc_id they form the primary key.
- As it’s a weak entity all customers are associated with exactly one Service Center which is represented by the arrow from Customer to associated with relationship.
- The customer is further linked with the Vehicle entity by own relationship.
- Each customer is linked with zero or more vehicles.

→ Entity Services

- {service_id, service_name}
- The service_id attribute acts as the primary key

- All services are categorized into sub-categories - Repair and Maintenance. This is an overlapping and total covering ISA relationship.
- The Repair category is further divided into 6 subcategories - engine, electrical, transmission, heating_AC, tire and exhaust.
- The Maintenance category has type attribute which has either A,B or C value.
- Services entity is linked with Model entity by offer relationship.

→ Entity Model
- {model_name}
- The model_name attribute acts as the primary key

→ Relationship has
- It links the weak entity employee and its parent entity Service_Center
- This relationship ensures that each employee is associated with exactly one service center and each service center has 1 or more employees.

→ Relationship associated with
- It links the weak entity Customer and its parent entity Service_Center
- This relationship ensures that each customer is associated with exactly one service center.

→ Relationship owns
- It links the Vehicle entity with the weak entity Customer.
- This relationship ensures that each vehicle is associated with exactly one customer and there may be some customers without any vehicle.

→ Relationship offer
- It links the Services and Model entity
- This relationship has duration as its own attribute

→ Relationship provides
- It is an aggregation relationship between the Service Center and offer relationship
- It has its own attribute price which determines the price of a particular service of a particular model at a particular store.

3. A translation of partial E-R model into SQL:

1) SERVICE CENTER TABLE

```
CREATE TABLE SERVICE_CENTER(
sc_id number(4),
address varchar2(30) UNIQUE NOT NULL,
telephone_no number(10) UNIQUE NOT NULL,
manager_id number(9) UNIQUE NOT NULL,
min_wage number(2) NOT NULL,
max_wage number(2) NOT NULL,
constraint pk_sc primary key sc_id
)
```

2) CONTRACT BASIS TABLE

```
CREATE TABLE CONTRACT_BASIS(
emp_id number(9),
sc_id number(4) NOT NULL,
emp_name varchar2(10) NOT NULL,
address varchar2(30) NOT NULL,
phone_no number(10) NOT NULL,
email varchar2(10) NOT NULL,
emp_role varchar2(10) NOT NULL,
salary number(5) NOT NULL,
constraint pk_cb primary key (emp_id, sc_id),
constraint fk_sc foreign key sc_id references SERVICE_CENTER,
CHECK ( UNIQUE (sc_id where emp_role='receptionist')),
CHECK ( UNIQUE (sc_id where emp_role='manager'))
);
```

3) HOURLY BASIS TABLE

```
CREATE TABLE HOURLY_BASIS(
emp_id number(9),
sc_id number(4) NOT NULL,
emp_name varchar2(10) NOT NULL,
address varchar2(30) NOT NULL,
phone_no number(10) NOT NULL,
email varchar2(10) NOT NULL,
emp_role varchar2(10) NOT NULL,
hourly_wage number(2) NOT NULL,
hrs_per_week number(2) NOT NULL,
constraint pk_cb primary key (emp_id, sc_id),
constraint fk_sc foreign key sc_id references SERVICE_CENTER,
CHECK( hrs_per_week < 50)
);
```

4) Condition to check hourly_wage between min_wage and max_wage

```
CREATE FUNCTION dbo.MIN_MAX_CHECK(@sc_id number(4), @hourly_wage number(2))
RETURNS varchar2(1)
BEGIN
    DECLARE @min_wage AS number(2)
    DECLARE @max_wage AS number(2)

    SElECT @min_wage = min_wage
    FROM SERVICE_CENTER
    WHERE sc_id = @sc_id

    SElECT @max_wage = max_wage
    FROM SERVICE_CENTER AS S
    WHERE S.sc_id = @sc_id

    IF (@hourly_wage > @min_wage) AND (@hourly_wage < @max_wage)
    BEGIN
        SELECT ISNULL(NULL, '1')
    END
    ELSE
    BEGIN
        SELECT ISNULL(NULL, '0')
    END
END

ALTER TABLE HOURLY_BASIS(
ADD CHECK(dbo.MIN_MAX_CHECK(sc_id, hourly_wage) = '1')
);
```

5) CAR MODEL

```
CREATE TABLE CAR_MODEL(
model_name varchar2(10),
constraint pk_cm primary key model_name
)
```

6) OFFERED FOR TABLE

```
CREATE TABLE OFFERED_FOR(
model_name varchar2(10) NOT NULL,
service_id number(5) NOT NULL,
service_duration number(2) NOT NULL,
constraint pk_of primary key (model_name, service_id)
)
```

7) REPAIR SERVICES

```
CREATE TABLE REPAIR_ENGINE(
service_id number(5),
service_name varchar2(20) NOT NULL,
constraint pk_re primary key service_id
);

CREATE TABLE REPAIR_ELECTRICAL(
service_id number(5),
service_name varchar2(20) NOT NULL,
constraint pk_rel primary key service_id
);

CREATE TABLE REPAIR_TRANSMISSION(
service_id number(5),
service_name varchar2(20) NOT NULL,
constraint pk_rt primary key service_id
);

CREATE TABLE REPAIR_HEATING_AC(
service_id number(5),
service_name varchar2(20) NOT NULL,
constraint pk_rh primary key service_id
);

CREATE TABLE REPAIR_TIRE(
service_id number(5),
service_name varchar2(20) NOT NULL,
constraint pk_rtr primary key service_id
);

CREATE TABLE REPAIR_EXHAUST(
service_id number(5),
service_name varchar2(20) NOT NULL,
constraint pk_rex primary key service_id
);
```

8) MAINTENANCE TABLE

```
CREATE TABLE MAINTENANCE(
service_id number(5),
service_name varchar2(20) NOT NULL,
main_type varchar2(1) NOT NULL,
constraint pk_mt primary key service_id
);
```

9) PROVIDES TABLE

```
CREATE TABLE PROVIDES(
service_id NOT NULL,
sc_id NOT NULL,
model_name NOT NULL,
price number(5) NOT NULL,
constraint pk_p primary key (service_id, sc_id, model_name),
constraint fk_sc foreign key sc_id references SERVICE_CENTER,
constraint fk_m foreign key model_name references CAR_MODEL,
constraint fk_ser foreign key service_id references SERVICE
);
```

10) CUSTOMER TABLE

```
CREATE TABLE CUSTOMER(
cust_id number(5),
sc_id number(4) NOT NULL,
lname varchar2(10) NOT NULL,
fname varchar2(10) NOT NULL,
balance_status bit default(0),
constraint pk_cust primary key (cust_id, sc_id),
constraint fk_sc foreign key sc_id references SERVICE_CENTER
);
```

11) VEHICLE TABLE

```
CREATE TABLE VEHICLE(
vin_number number(8),
cust_id number(5) NOT NULL,
sc_id number(4) NOT NULL,
manf_name varchar2(10) NOT NULL,
last_serv varchar2(1) NOT NULL,
vehicle_year number(4) NOT NULL,
curr_mileage number(5) NOT NULL,
constraint pk_v primary key vin_number,
constraint fk_cust foreign key (cust_id, sc_id) references CUSTOMER
);
```

12) Adding active status in CUSTOMER TABLE

```
CREATE FUNCTION dbo.GET_ACTIVE_STATUS(@sc_id number(4), @cust_id number(2))
RETURNS varchar2(1)
BEGIN
    SELECT vin_number FROM VEHICLE AS V WHERE V.sc_id=@sc_id AND V.cust_id=@cust_id ISNULL(NULL, '0')
END

ALTER TABLE CUSTOMER(
ADD active_status varchar2(1) AS dbo.GET_ACTIVE_STATUS(sc_id, cust_id) default(1)
);
```

4. Constraints captured in our partial E-R model:

Description	Constraints
Each service center is identified by a globally unique ID, address, and a telephone number.	Unique Keys
Each service center is identified by a globally unique ID.	Primary Key
The general employee structure in each center has a manager	Not Null in Service Center Table (manager_id)
There should be only one receptionist per store.	Check (Unique (sc_id where role='receptionist'))
For each employee, we store their locally unique 9-digit employee ID, name, address, email address, phone number, service center ID, and the role at the service center they work at.	Not Null values
Each employee is associated with only one service center.	{emp_id, sc_id} as Primary Key
While the manager and receptionist are contract employees with fixed annual salary, mechanics are hourly paid workers.	Total Covering, Non-overlapping ISA relationship
Some services like Brake Repair are both maintenance and repair services while others like Oil Change are exclusively maintenance services.	Overlapping ISA relationship
Individual services belong to only one of the 6 repair service subcategories	Total Covering, Non-overlapping ISA relationship
Individual services have a globally unique number and name.	Unique attributes (service_name)
Individual services have a globally unique number.	Primary Key (service_id)
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.	Check Constraint
Each mechanic works no more than 50hours a week.	CHECK(hrs_per_week < 50)

A mechanic may ask for and may be given time off, as long as there are always at least 3 mechanics present at any given time.	Check Constraint
A Customer has id (an integer) that is unique with respect to a specific center,	{cust_Id, sc_id} as Primary Key
Vehicle which is identified by globally unique vin number (8 alphanumeric characters)	Primary Key (vin_number)
Vehicle which is identified by globally unique vin number (8 alphanumeric characters), car manufacturer e.g. Honda, and current mileage (integer), year, the last scheduled maintenance service class performed denoted by a single character ‘A’, ‘B’, or ‘C’.	Not Null values
Customers become “inactive” if their profiles no longer include any vehicles	View SQL QUERY in 3rd PART
For each store, store id (either auto generated or given), address and store manager information (firstname, last name, username, password, salary, and employeeid) and the store’s minimum and maximum hourly wage for mechanics	Not Null values
Salaries must fall within the min and max salaries set up for the stores	View SQL QUERY in 3rd PART
They should be allowed to only register cars that belong to the list of manufacturers that Downtown Auto works on	Check Constraint
When a customer sells a car they can delete a car from their profile. If the number of cars drops to zero, the customer’s status becomes inactive.	Check Constraint

5. Functional Dependencies:

1. In Service_Center Entity -
    1. address → {manager_id, telephone_no, min_wage, max_wage}
    2. telephone_no → {manager_id, address, min_wage, max_wage}
    3. manager_id → {address, telephone_no, min_wage, max_wage}
2. In Employee Entity -
    1. {emp_id, sc_id} → {emp_name, address, phone_no, email, role}
3. In Services Entity -
    1. service_name → {service_id}
4. In Customer Entity -
    1. {cust_id, sc_id} → {fname, lname, status}
5. In Vehicle Entity -
    1. vin_number → {manf_name, last_serv, curr_milage, year}