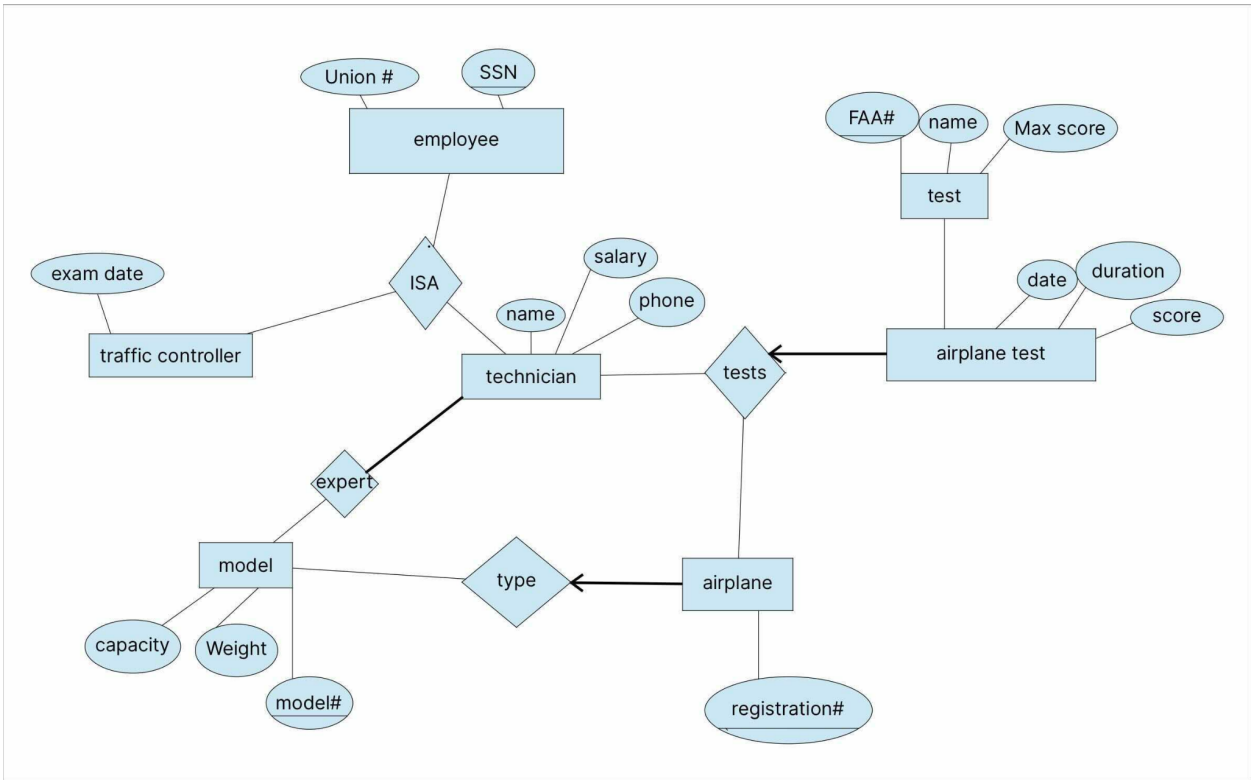
Homework 1 / 5707 Spring 2021 Team 12
Bela Demchouk (derga002), Kelsey Neis (neis), Ashwin Sridhar (sridh052)

2.6.1



Assumptions:

- Technician and traffic controller do not overlap, since employees only belong to one union.
- A single technician can do the same test on the same exact plane multiple times per day. Thus the only thing differing would be start time.

Explanation of ER components:
Thick edge: FULL PARTICIPATION
Thin edge: PARTIAL PARTICIPATION
Arrow (->) 1-1
Line only 1-to-Many

2.6.2 A constraint that checks that there is a tuple in expert that has the model of the airplane being tested and the technician conducting the test.

3.8.1 The Works table references ‘did’, the primary key of Dept, as a foreign key. Therefore, a user cannot insert a value for ‘did’ in the Works table if it does not already exist in the Dept table. If a user then deletes a tuple in the Dept table, then the tuple in the Works table that has the same ‘did’ will either be deleted(cascade) or the deletion will not be allowed (no action).

3.8.2. CREATE TABLE Emp (
eid int,
ename char(20),
age int,
salary real,
PRIMARY KEY(eid)

```
        );
CREATE TABLE Works (
        eid int,
        did int,
        pcttime int,
        FOREIGN KEY(did) REFERENCES Dept,
        FOREIGN KEY(eid) REFERENCES Emp,
        PRIMARY KEY(eid, did)
);

CREATE TABLE Dept (
                did int,
                dname char(20),
                budget real,
                managerid int,
                PRIMARY KEY(did)
        );
```

3.8.3 CREATE TABLE Dept (

```
                did int,
                dname string,
                budget real,
                managerid int NOT NULL,
                PRIMARY KEY(did)
        );
```

3.8.4 INSERT INTO Emp(eid,ename,age,salary)

```
        VALUES (101,"John Doe",32,15000)
```

3.8.5 UPDATE TABLE Emp SET Salary = Salary * 1.10

3.8.6 DELETE FROM Dept where dname = "Toy"

CASE 1: Chosen referential shema is:

```
CREATE TABLE Works (
        eid Int Not Null,
        did Int Not Null,
        pcttime Int Not Null,
        FOREIGN KEY(eid) REFERENCES Emp,
        FOREIGN KEY(did) REFERENCES Dept,
        PRIMARY KEY(eid, did),
        );
```

Default behavior is "NO ACTION" if FOREIGN KEY into Dept exists. Thus in this case NOTHING will happen. (The "Toy" department will stay in the db)

CASE 2: Chosen referential shema is:

```
CREATE TABLE Works (
        eid Int,
        did Int,
        pcttime Int NOT NULL,
        FOREIGN KEY(did) REFERENCES Dept ON DELETE CASCADE,
```

```
        FOREIGN KEY(eid) REFERENCES Emp,
        PRIMARY KEY(eid, did)
        );
```

Since the row containing “Toy” is deleted, all the rows in the Works table with the corresponding ‘did’ are deleted as well (CASCADE action).

3.16)

```
CREATE TABLE Employee (
        ssn VARCHAR(11),
        union_number VARCHAR NOT NULL,
        PRIMARY KEY(ssn)
);
CREATE TABLE Traffic_controller (
        exam_date DATE NOT NULL,
        ssn VARCHAR(11),
        FOREIGN KEY(ssn) REFERENCES Employee,
        PRIMARY KEY(ssn)
);
CREATE TABLE Technician (
        ssn VARCHAR(11),
        name CHAR(25) NOT NULL,
        salary INT NOT NULL,
        address VARCHAR(100) NOT NULL,
        phone_number VARCHAR(12) NOT NULL,
        FOREIGN KEY (ssn) REFERENCES Employee ON DELETE CASCADE,
        PRIMARY KEY(ssn)
);
CREATE TABLE Expert (
        ssn varchar(11),
        Model_number varchar(20),
        FOREIGN KEY(model_number) REFERENCES Model,
        FOREIGN KEY(ssn) REFERENCES Technician,
        PRIMARY KEY(model_number, ssn)
);

CREATE TABLE Model (
        model_number varchar(20) NOT NULL,
        weight int NOT NULL,
        capacity int NOT NULL,
        PRIMARY KEY(model_number)
);

CREATE TABLE Airplane (
        registration_number int NOT NULL,
        model_number varchar(20) NOT NULL,
        FOREIGN KEY(model_number) REFERENCES Model,
        PRIMARY KEY(registration_number)
```

```
);

CREATE TABLE Test (
        faa_number int NOT NULL,
        name char(20) NOT NULL,
        max_score int NOT NULL,
        PRIMARY_KEY(faa_number)
);

CREATE TABLE Plane_Test (
        faa_number int NOT NULL,
        test_date datetime NOT NULL,
        duration int NOT NULL,
        score int NOT NULL,
        tech_ssn varchar(11) NOT NULL,
        plane_reg_num int NOT NULL,
        PRIMARY KEY(plane_reg_num, tech_ssn, test_date, faa_number),
        FOREIGN KEY(tech_ssn) REFERENCES Technician,
        FOREIGN KEY(plane_reg_num) REFERENCES Airplane,
        FOREIGN KEY(faa_number) REFERENCES Test
);
```

Expert constraint (2.6.2)
Assumption: Table already exists, new FAA rules come in dictating that only model experts perform plane tests.
In English: Query the Airplane table using plane_reg_num to look up the model. Then use the model number result in conjunction with tech_ssn to query the Expert table for the exact row matching both tech and model expertise.
IN Query Language:

```
CREATE ASSERTION Model_Expert
CHECK (TRUE = (
                SELECT ALL tech_ssn
                FROM Airplane_Test
                INNER JOIN Airplane ON Airplane_Test.plane_reg_num = Airplane.registration_num
                INNER JOIN Expert ON Airplane.model_number = Expert.model_number
                WHERE Expert.ssn = Airplane_Test.tech_ssn
        )
  );
```

4.4

1. Find the list of supplier names that supply red parts costing less than $100
2. Will not return anything, because the projection of sid will remove the sname column.
3. A list of supplier names that supply both red and green parts costing less than $100
4. A list of supplier sids that supply both red and green parts costing less than $100
5. A list of supplier names that supply both red and green parts costing less than $100