# CS 6360.003 Database System
# Online Airport System
# Professor: Dr. Jalal Omer Sheikh

## Team Members

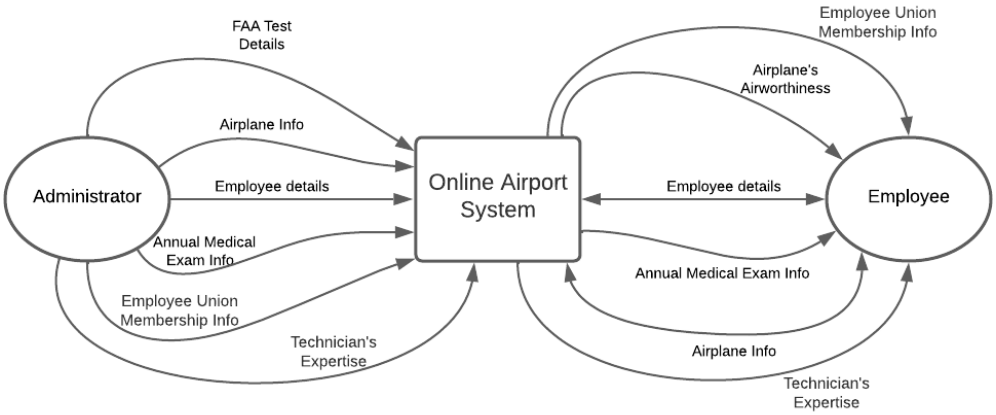

| | |
|---|---|
| Meghana Sai Bijivemula | MXB210011 |
| Milan Miriam George | MMG210001 |
| Wooseong Yang | WXY210000 |
| Yash Pratapwar | YXP200011 |
| Surekha Kumari | SXK210055 |

# Table Of Contents

## 1. System Description

Airport officials felt the need to maintain airport-related information and organize it into the **Online Airport System (OAS)**, so they hired us as experts. They provided us with information about the airplanes that are stationed and maintained at the airport to develop a system that suits their needs. **OAS** needs to use the registration number of airplanes, which is a unique number, to identify the plane. There are several airplane models in the airport, and each one is identified by a model number and has its own capacity and weight. The application will be used by officials and employees who wish to view or perform actions on airport functioning. The system also needs to manage employee details, and there are two types of employees. The first category of employees are technicians and they have specialized airplane models that they are experts in. Second category of employees are traffic controllers, and they undergo annual medical examination. The date of the most recent exam for each traffic controller must be stored, as they are responsible for the lives of many. All the airport employees, including technicians and traffic controllers, belong to unions. **OAS** will reflect these details about employees, officials and airplanes associated. This system also needs to manage information about the Federal Aviation Administration tests that are conducted by technicians periodically to ensure that airplanes are still airworthy. The system will keep track of each test done on the airplanes by storing the test date, number of hours spent by the technician to perform the test and the score.

## 2. Context Diagram



## 3. Functional Requirements

**Assumptions**

a. Details provided by an entity are stored in the Online Airport System.
b. Employees (Airport staff - technicians/traffic controllers/other employees) upload their details - name, SSN (Social Security Number), address, and phone number.
c. The airplane registration number and model and its details are stored in the system.

**A. Login Functional Requirements**

a. The system will allow the user to log in.
b. The system will verify the username and password.
c. The system will not allow the user to log in with an invalid username or password.
d. The system will be able to remember usernames and passwords.
e. The system will allow users to create accounts.
f. The system will enable users to log out of their accounts.

**B. Browsing Functional Requirements**

a. The airplane's registration number can be viewed.
b. The airplane's model can be viewed.
c. The airplane's model number can be viewed.
d. The capacity and weight of the airplane model can be viewed
e. The date of the traffic controller's medical examination can be viewed.
f. The Federal Aviation Administration (FAA) test number can be viewed.
g. The test's maximum possible score can be viewed.
h. The time of the test can be viewed.
i. The number of hours performed by said technician on a given test can be viewed.
j. The employee membership number can be viewed.
k. The name, SSN, address, phone number, and salary of each technician can be viewed.

**C. Administrator Functional Requirements**

a. The name, SSN, address, phone number, and salary of each technician can be changed/updated.
b. The new airplane model can be inserted into the database along with its capacity and weight.
c. The new employee/technician information can be added.
d. The capacity of the existing airplane model can be changed.
e. The weight of the existing airplane model can be included.
f. The Federal Aviation Administration (FAA) test number, a name, and a maximum possible score can be stored.
g. The time of the test can be stored.
h. The traffic controllers' annual examination record can be stored.
i. The union membership record can be maintained.
j. The expertise information of each employee can be stored.
k. The number of hours performed by a technician on a test can be stored.

## 4. Non-Functional Requirements

**a. Speed:**

a. The system will not lag due to optimized database.
b. The system will have a quick response time as data will be retrieved from the structured database.

**b. Security:**

a. Account creation:
    i. The system will allow a new user to create an account.
b. Password Generation:
    i. The system will allow all users to create a password to secure their account.
c. Login:
    i. The system will allow the user to log in with only the correct combination of username and password.

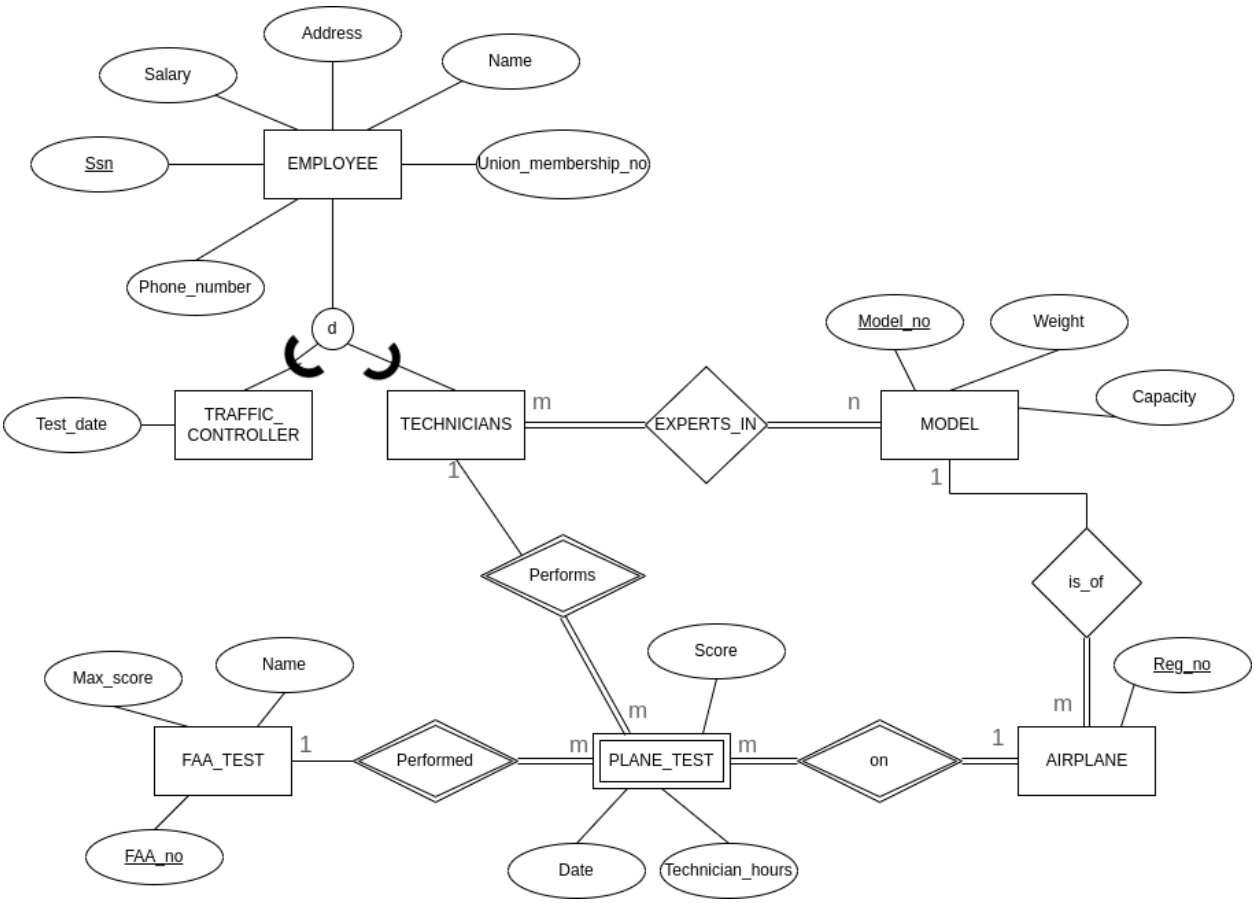
**c. Reliability:**

a. Real Time Data Update:
    i. The system will not show discrepancies in data and will have real-time data retrievals and updates.
b. Number of Critical Failures:
    i. The system will not have critical failures and will have a session-based login and logoff system.

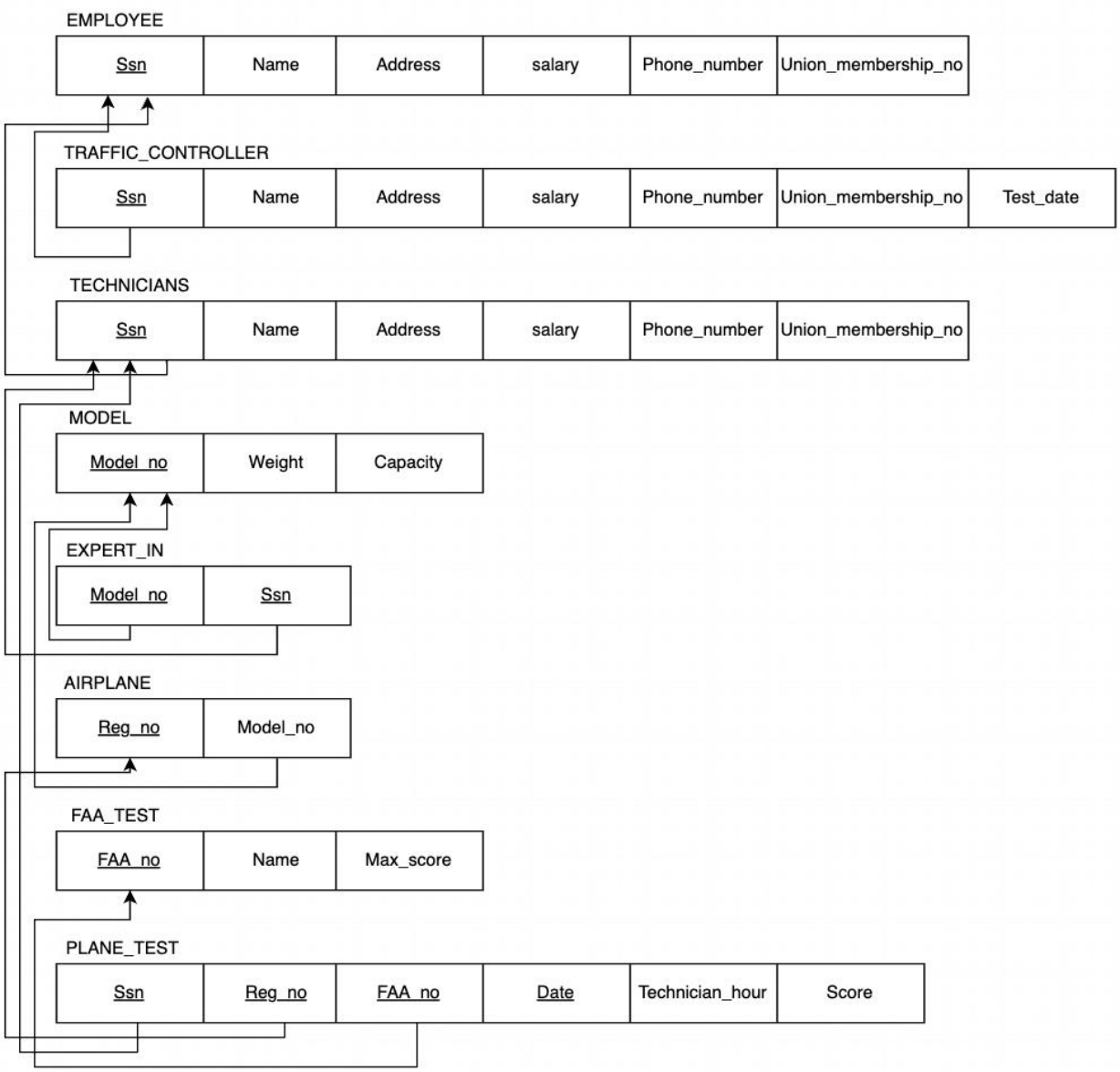
**d. Usability:**

a. Navigation:
    i. The user will be able to navigate through the system. They will be able to retrieve and view data as per their requirement.
b. Purpose of Feature
    i. The user will be able to determine the purpose of the feature.
c. Quality of Performance:
    i. The system will perform as expected.

## 5. Entity Relationship Diagram



## 6. Database Schema Diagram



## 7. Business Rules and Constraints

**Rules**

a. All employees must belong to the union.
b. Union membership details are private (users can only view their membership information).
c. Only the administrator can modify technician expertise details.
d. Traffic controllers cannot modify annual medical exam details.
e. Only the technicians can add FAA test details.
f. Employees cannot modify FAA test details.
g. Traffic controllers must pass the annual medical tests to continue in their position.

**Constraints**

a. Each technician can be an expert on airplane model(s).
b. A technician can have multiple expertise.
c. Each airplane belongs to a model, and many airplanes can be of same model.
d. Each technician can perform multiple FAA tests.
e. Each FAA test needs to be performed by a single technician.
f. Each technician can conduct tests on many airplanes.
g. Each airplane must undergo multiple tests by an expert technician.
h. The name, SSN, address, phone of an employee cannot be null.
i. An employee in a union is identified by a union membership number; it cannot be null.
j. The unique identifier can be obtained from the emploees' SSN; SSN is a primary key.
k. The salary and SSN of employees should be private.

## 8. Interface Requirements

a. Admin Interface allows administrators to create, modify or delete data from the database (CRUD operations).
b. The login page allows all employees along with administrator to sign into the system.
c. Technician page allows technicians to view their expertise, information regarding FAA tests performed on airplanes.
d. Traffic controller page allows traffic controllers to view their annual medical exam information.
**e.** Employee page allows employees to view their union membership details and update their personal details.
f. Airplane Information page will allow the technician to view the details of airplanes along with its airworthy status.

## 9. Functional Dependencies

a. [EMPLOYEE]
   i. SSN -> Name, Address, Salary, Phone Number, salary, union_membership_no
   ii. Union_membership_no -> Ssn, Name, Address, Salary, Phone Number, salary

b. [TRAFFIC_CONTROLLER]
   i. SSN -> Name, Address, Salary, Phone Number, salary, union_membership_no, test data
   ii. Union_membership_no -> SSN, Name, Address, Salary, Phone Number, salary, test data

c. [TECHNICIAN]
   i. SSN -> Name, Address, Salary, Phone Number, salary, union_membership_no.
   ii. Union_membership_no. -> SSN, Name, Address, Salary, Phone Number, salary

d. [MODEL]
   i. Model_no -> weight, capacity

e. [EXPERT_IN]
   i. SSN ->-> model_no

f. [AIRPLANE]
   i. Reg_no. -> model_no

g. [FAA_TEST]
   i. FAA_No. -> Name, Max_score

h. [PLANE_TEST]
   i. {SSN, Reg_no., FAA_No. , Date} -> {Technician_hour, score}

## 10. Normalization

**First Normal Form**

All the above tables are in First Normal Form because they do not contain composite attributes, multivalued attributes, or nested relations.

**Second Normal Form**

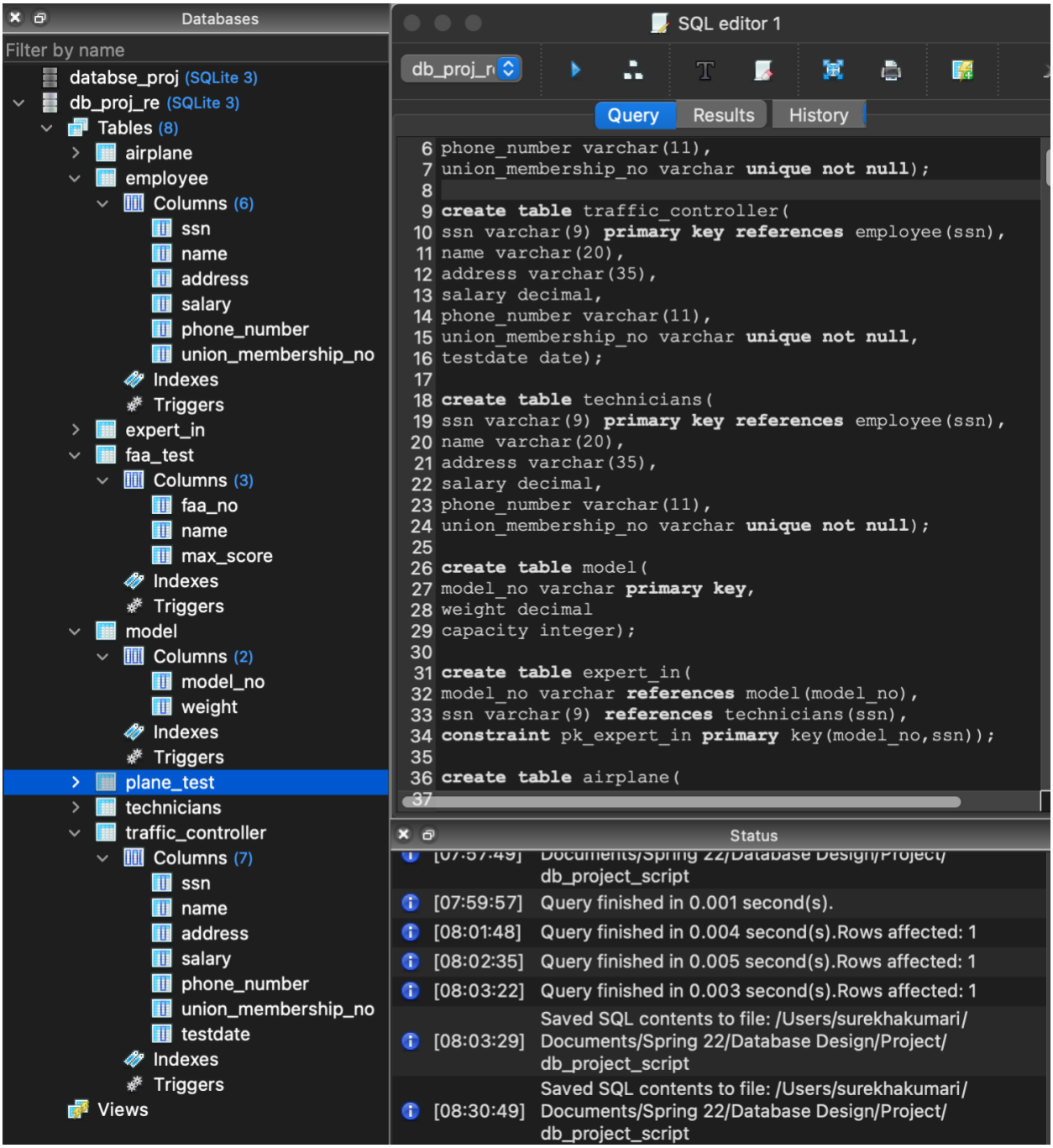
i. EMPLOYEE Table – Already in Second Normal Form because it does not contain partial functional dependency
ii. TRAFFIC_CONTROLLER Table – Already in Second Normal Form because it does not contain partial functional dependency

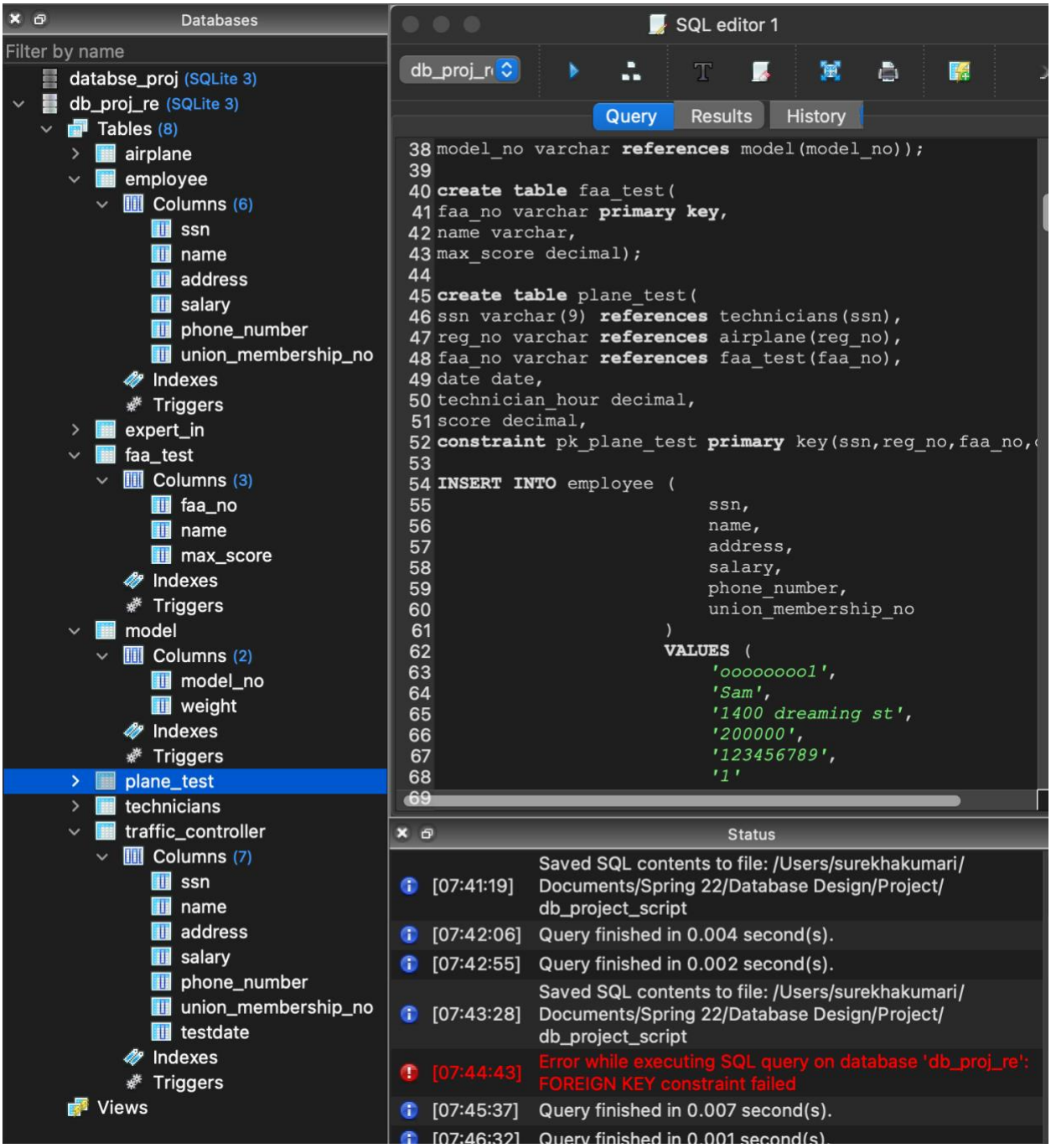
iii. TECHNICIANS Table – Already in Second Normal Form because it does not contain partial functional dependency
iv. MODEL Table – Already in Second Normal Form because it does not contain partial functional dependency
v. EXPERT_IN Table – Already in Second Normal Form because it does not contain partial functional dependency
vi. AIRPLANE Table – Already in Second Normal Form because it does not contain partial functional dependency
vii. FAA_TEST Table – Already in Second Normal Form because it does not contain partial functional dependency
viii. PLANE Table – Already in Second Normal Form because it does not contain partial functional dependency

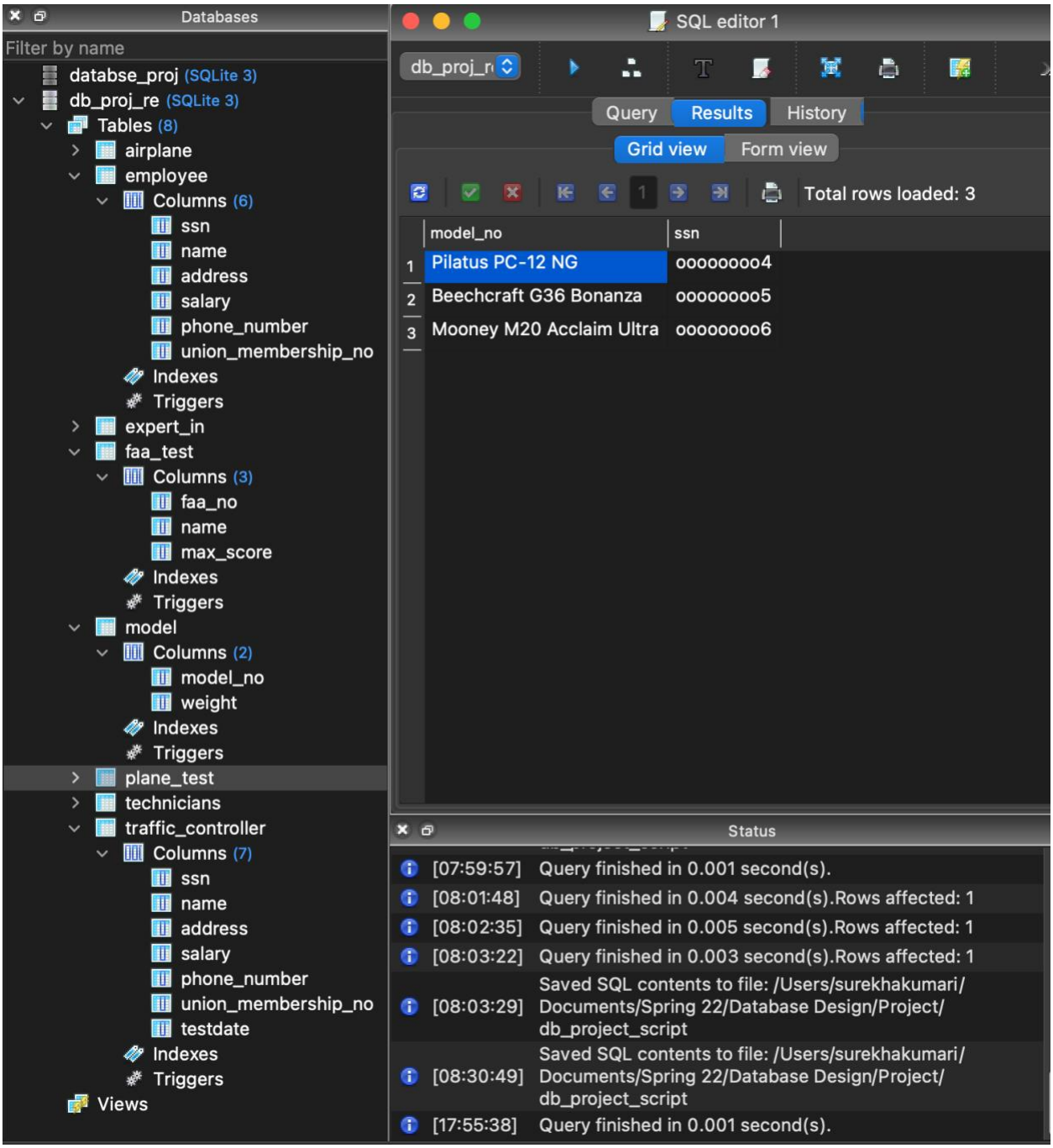
**Third Normal Form**

i. EMPLOYEE Table – Already in Third Normal Form because it does not contain transitive functional dependency
ii. TRAFFIC_CONTROLLER Table – Already in Third Normal Form because it does not contain transitive functional dependency
iii. TECHNICIANS Table – Already in Third Normal Form because it does not contain transitive functional dependency
iv. MODEL Table – Already in Third Normal Form because it does not contain transitive functional dependency
v. EXPERT_IN Table – Already in Third Normal Form because it does not contain transitive functional dependency
vi. AIRPLANE Table – Already in Third Normal Form because it does not contain transitive functional dependency
vii. FAA_TEST Table – Already in Third Normal Form because it does not contain transitive functional dependency
viii. PLANE Table – Already in Third Normal Form because it does not contain transitive functional dependency

# 11. Database Implementation




Databases
Filter by name
databse_proj (SQLite 3)
db_proj_re (SQLite 3)
Tables (8)
airplane
employee
Columns (6)
ssn
name
address
salary
phone_number
union_membership_no
Indexes
Triggers
expert_in
faa_test
Columns (3)
faa_no
name
max_score
Indexes
Triggers
model
Columns (2)
model_no
weight
Indexes
Triggers
plane_test
technicians
traffic_controller
Columns (7)
ssn
name
address
salary
phone_number
union_membership_no
testdate
Indexes
Triggers
Views
SQL editor 1
db_proj_r
Query
Results
History
38 model_no varchar references model(model_no));
39
40 create table faa_test(
41 faa_no varchar primary key,
42 name varchar,
43 max_score decimal);
44
45 create table plane_test(
46 ssn varchar(9) references technicians(ssn),
47 reg_no varchar references airplane(reg_no),
48 faa_no varchar references faa_test(faa_no),
49 date date,
50 technician_hour decimal,
51 score decimal,
52 constraint pk_plane_test primary key(ssn,reg_no,faa_no,
53
54 INSERT INTO employee (
55 ssn,
56 name,
57 address,
58 salary,
59 phone_number,
60 union_membership_no
61 )
62 VALUES (
63 '00000001',
64 'Sam',
65 '1400 dreaming st',
66 '200000',
67 '123456789',
68 '1'
69
Status
[07:41:19] Saved SQL contents to file: /Users/surekhakumari/Documents/Spring 22/Database Design/Project/db_project_script
[07:42:06] Query finished in 0.004 second(s).
[07:42:55] Query finished in 0.002 second(s).
[07:43:28] Saved SQL contents to file: /Users/surekhakumari/Documents/Spring 22/Database Design/Project/db_project_script
[07:44:43] Error while executing SQL query on database 'db_proj_re': FOREIGN KEY constraint failed
[07:45:37] Query finished in 0.007 second(s).
[07:46:32] Query finished in 0.001 second(s).


Databases
Filter by name
databse_proj (SQLite 3)
db_proj_re (SQLite 3)
Tables (8)
airplane
employee
Columns (6)
ssn
name
address
salary
phone_number
union_membership_no
Indexes
Triggers
expert_in
faa_test
Columns (3)
faa_no
name
max_score
Indexes
Triggers
model
Columns (2)
model_no
weight
Indexes
Triggers
plane_test
technicians
traffic_controller
Columns (7)
ssn
name
address
salary
phone_number
union_membership_no
testdate
Indexes
Triggers
Views
SQL editor 1
db_proj_r
Query
Results
History
Grid view
Form view
Total rows loaded: 3
model_no
ssn
1 Pilatus PC-12 NG 000000004
2 Beechcraft G36 Bonanza 000000005
3 Mooney M20 Acclaim Ultra 000000006
Status
[07:59:57] Query finished in 0.001 second(s).
[08:01:48] Query finished in 0.004 second(s).Rows affected: 1
[08:02:35] Query finished in 0.005 second(s).Rows affected: 1
[08:03:22] Query finished in 0.003 second(s).Rows affected: 1
[08:03:29] Saved SQL contents to file: /Users/surekhakumari/Documents/Spring 22/Database Design/Project/db_project_script
[08:30:49] Saved SQL contents to file: /Users/surekhakumari/Documents/Spring 22/Database Design/Project/db_project_script
[17:55:38] Query finished in 0.001 second(s).

## 12. SQL Statements for DB Construction and Population

```
Create table employee(
ssn varchar(9) primary key,
name varchar(20),
address varchar(35),
salary decimal,
phone_number varchar(11),
union_membership_no varchar unique not null);

create table traffic_controller(
ssn varchar(9) primary key references employee(ssn),
name varchar(20),
address varchar(35),
salary decimal,
phone_number varchar(11),
union_membership_no varchar unique not null,
testdate date);

create table technicians(
ssn varchar(9) primary key references employee(ssn),
name varchar(20),
address varchar(35),
salary decimal,
phone_number varchar(11),
union_membership_no varchar unique not null);

create table model(
model_no varchar primary key,
weight decimal
capacity integer);

create table expert_in(
model_no varchar references model(model_no),
ssn varchar(9) references technicians(ssn),
constraint pk_expert_in primary key(model_no,ssn));

create table airplane(
reg_no varchar primary key,
model_no varchar references model(model_no));

create table faa_test(
faa_no varchar primary key,
name varchar,
max_score decimal);
```

```
create table plane_test(
ssn varchar(9) references technicians(ssn),
reg_no varchar references airplane(reg_no),
faa_no varchar references faa_test(faa_no),
date date,
technician_hour decimal,
score decimal,
constraint pk_plane_test primary key(ssn,reg_no,faa_no,date));

INSERT INTO employee (
ssn,
name,
address,
salary,
phone_number,
union_membership_no
)
VALUES (
'oooooooo1',
'Sam',
'1400 dreaming st',
'200000',
'123456789',
'1'
);

INSERT INTO traffic_controller (
ssn,
name,
address,
salary,
phone_number,
union_membership_no,
testdate
)
VALUES (
'oooooooo1',
'Sam',
'1400 dreaming st',
'200000',
'123456789',
'1',
'2021-09-21'
);

INSERT INTO employee (
```

```
ssn,
name,
address,
salary,
phone_number,
union_membership_no
)
VALUES (
'oooooooo2',
'Tricia',
'1500 logic st',
'400000',
'123456799',
'2'
);

INSERT INTO traffic_controller (
ssn,
name,
address,
salary,
phone_number,
union_membership_no,
testdate
)
VALUES (
'oooooooo2',
'Tricia',
'1500 logic st',
'400000',
'123456799',
'2',
'2021-08-26'
);

INSERT INTO employee (
ssn,
name,
address,
salary,
phone_number,
union_membership_no
)
VALUES (
'oooooooo3',
'Rita',
```

```
'1400 dreaming st',
'300000',
'123456989',
'3'
);

INSERT INTO traffic_controller (
ssn,
name,
address,
salary,
phone_number,
union_membership_no,
testdate
)
VALUES (
'oooooooo3',
'Rita',
'1400 dreaming st',
'300000',
'123456989',
'3',
'2021-07-26'
);

INSERT INTO employee (
ssn,
name,
address,
salary,
phone_number,
union_membership_no
)
VALUES (
'oooooooo4',
'Shakira',
'1890 waka waka st',
'800000',
'123456999',
'4'
);

INSERT INTO technicians (
ssn,
name,
address,
```

```
salary,
phone_number,
union_membership_no
)
VALUES (
'oooooooo4',
'Shakira',
'1890 waka waka st',
'800000',
'123456999',
'4'
);

INSERT INTO employee (
ssn,
name,
address,
salary,
phone_number,
union_membership_no
)
VALUES (
'oooooooo5',
'Ed Sheeran',
'400 perfect st',
'700000',
'123459999',
'5'
);

INSERT INTO technicians (
ssn,
name,
address,
salary,
phone_number,
union_membership_no
)
VALUES (
'oooooooo5',
'Ed Sheeran',
'400 perfect st',
'700000',
'123459999',
'5'
);
```

```
INSERT INTO employee (
ssn,
name,
address,
salary,
phone_number,
union_membership_no
)
VALUES (
'ooooooo6',
'Elvis P',
'200 ghetto st',
'80000',
'123456000',
'6'
);

INSERT INTO technicians (
ssn,
name,
address,
salary,
phone_number,
union_membership_no
)
VALUES (
'ooooooo6',
'Elvis P',
'200 ghetto st',
'80000',
'123456000',
'6'
);

INSERT INTO employee (
ssn,
name,
address,
salary,
phone_number,
union_membership_no
)
VALUES (
'ooooooo7',
'F Sinatra',
```

```
'300 fly moon st',
'70000',
'123456001',
'7'
);

INSERT INTO model (
model_no,
weight
)
VALUES (
'Beechcraft G36 Bonanza',
'12500'
);

INSERT INTO model (
model_no,
weight
)
VALUES (
'Mooney M20 Acclaim Ultra',
'11700'
);

INSERT INTO model (
model_no,
weight
)
VALUES (
'Pilatus PC-12 NG',
'11230'
);

INSERT INTO expert_in (
model_no,
ssn
)
VALUES (
'Pilatus PC-12 NG',
'oooooooo4'
);

INSERT INTO expert_in (
model_no,
ssn
)
```

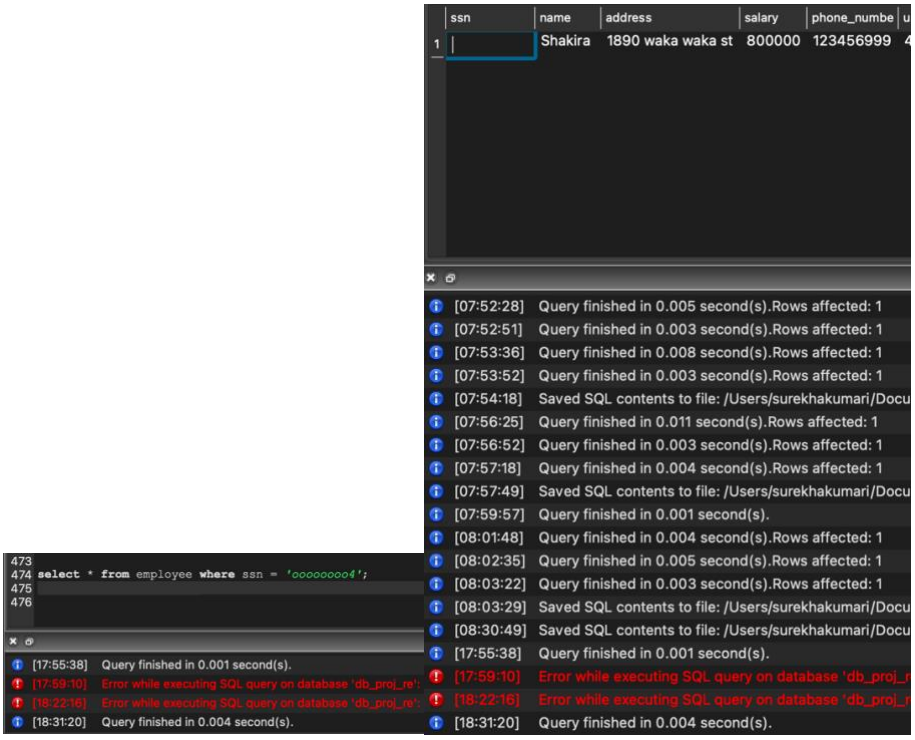
```
VALUES (
'Beechcraft G36 Bonanza',
'oooooooo5'
);

INSERT INTO expert_in (
model_no,
ssn
)
VALUES (
'Mooney M20 Acclaim Ultra',
'oooooooo6'
);

INSERT INTO airplane (
reg_no,
model_no
)
VALUES (
'1',
'Pilatus PC-12 NG'
);

INSERT INTO airplane (
reg_no,
model_no
)
VALUES (
'2',
'Pilatus PC-12 NG'
);

INSERT INTO airplane (
reg_no,
model_no
)
VALUES (
'3',
'Beechcraft G36 Bonanza'
);

INSERT INTO airplane (
reg_no,
model_no
)
VALUES (
```

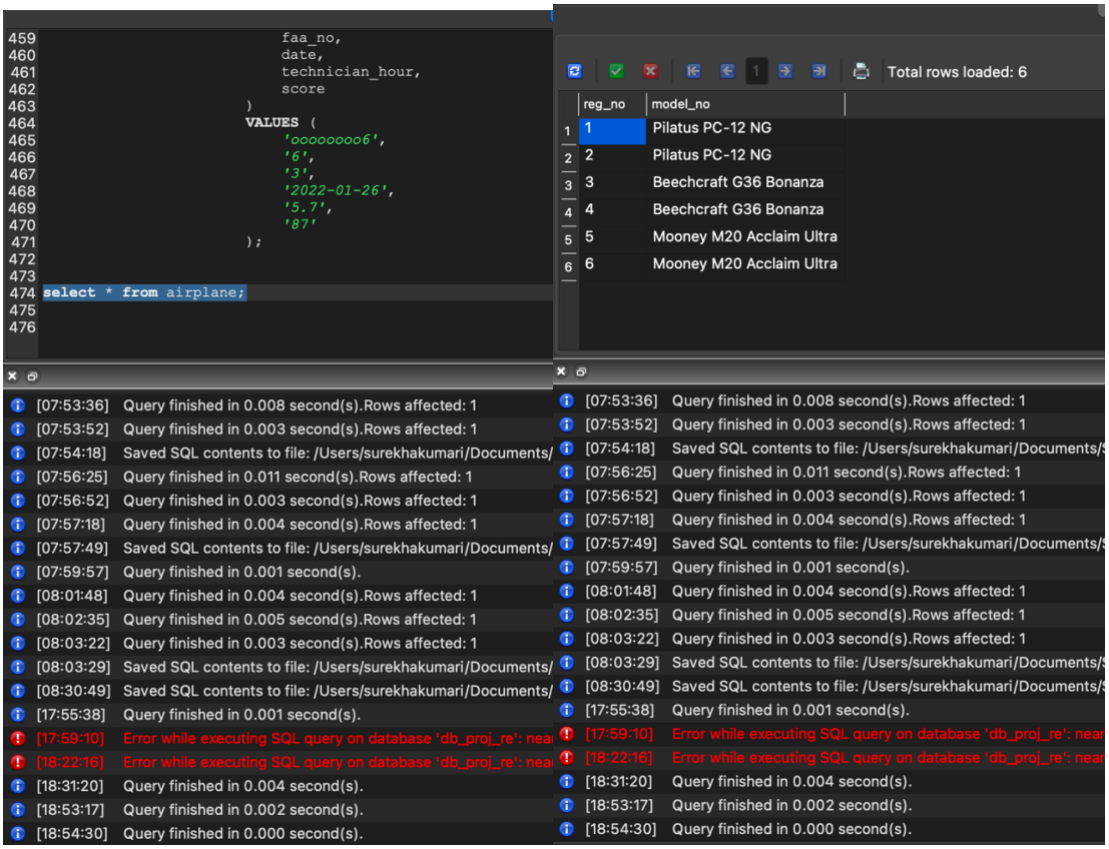
```
'4',
'Beechcraft G36 Bonanza'
);

INSERT INTO airplane (
reg_no,
model_no
)
VALUES (
'5',
'Mooney M20 Acclaim Ultra'
);

INSERT INTO airplane (
reg_no,
model_no
)
VALUES (
'6',
'Mooney M20 Acclaim Ultra'
);

INSERT INTO faa_test (
faa_no,
name,
max_score
)
VALUES (
'1',
'AMA',
'100'
);

INSERT INTO faa_test (
faa_no,
name,
max_score
)
VALUES (
'2',
'AMG',
'100'
);

INSERT INTO faa_test (
faa_no,
```

```
name,
max_score
)
VALUES (
'3',
'IAR',
'100'
);

INSERT INTO plane_test (
ssn,
reg_no,
faa_no,
date,
technician_hour,
score
)
VALUES (
'oooooooo4',
'2',
'1',
'2022-03-30',
'2.5',
'86'
);

INSERT INTO plane_test (
ssn,
reg_no,
faa_no,
date,
technician_hour,
score
)
VALUES (
'oooooooo5',
'3',
'2',
'2022-02-26',
'4.1',
'79'
);

INSERT INTO plane_test (
ssn,
reg_no,
```

```
faa_no,
date,
technician_hour,
score
)
VALUES (
'oooooooo6',
'6',
'3',
'2022-01-26',
'5.7',
'87'
);
```

# 13. Demonstration



```
                    faa_no,
                    date,
                    technician_hour,
                    score
                )
                VALUES (
                    '000000006',
                    '6',
                    '3',
                    '2022-01-26',
                    '5.7',
                    '87'
                );


select * from employee;
```

Total rows loaded: 7

| | ssn | name | address | salary | phone_numbe | union_men |
|---|---|---|---|---|---|---|
| 1 | 0... | Sam | 1400 dreaming st | 200000 | 123456789 | 1 |
| 2 | 0... | Tricia | 1500 logic st | 400000 | 123456799 | 2 |
| 3 | 0... | Rita | 1400 dreaming st | 300000 | 123456989 | 3 |
| 4 | 0... | Shakira | 1890 waka waka st | 800000 | 123456999 | 4 |
| 5 | 0... | Ed Sheeran | 400 perfect st | 700000 | 123459999 | 5 |
| 6 | 0... | Elvis P | 200 ghetto st | 80000 | 123456000 | 6 |
| 7 | 0... | F Sinatra | 300 fly moon st | 70000 | 123456001 | 7 |

```
[07:53:52] Query finished in 0.003 second(s).Rows affected: 1
[07:54:18] Saved SQL contents to file: /Users/surekhakumari/Doc
[07:56:25] Query finished in 0.011 second(s).Rows affected: 1
[07:56:52] Query finished in 0.003 second(s).Rows affected: 1
[07:57:18] Query finished in 0.004 second(s).Rows affected: 1
[07:57:49] Saved SQL contents to file: /Users/surekhakumari/Doc
[07:59:57] Query finished in 0.001 second(s).
[08:01:48] Query finished in 0.004 second(s).Rows affected: 1
[08:02:35] Query finished in 0.005 second(s).Rows affected: 1
[08:03:22] Query finished in 0.003 second(s).Rows affected: 1
[08:03:29] Saved SQL contents to file: /Users/surekhakumari/Doc
[08:30:49] Saved SQL contents to file: /Users/surekhakumari/Doc
[17:55:38] Query finished in 0.001 second(s).
[17:59:10] Error while executing SQL query on database 'db_proj_
[18:22:16] Error while executing SQL query on database 'db_proj_
[18:31:20] Query finished in 0.004 second(s).
[18:53:17] Query finished in 0.002 second(s).
[18:54:30] Query finished in 0.000 second(s).
[18:56:23] Query finished in 0.002 second(s).
```

```
[07:53:52] Query finished in 0.003 second(s).Rows affected: 1
[07:54:18] Saved SQL contents to file: /Users/surekhakumari/Documents/Sprin
[07:56:25] Query finished in 0.011 second(s).Rows affected: 1
[07:56:52] Query finished in 0.003 second(s).Rows affected: 1
[07:57:18] Query finished in 0.004 second(s).Rows affected: 1
[07:57:49] Saved SQL contents to file: /Users/surekhakumari/Documents/Sprin
[07:59:57] Query finished in 0.001 second(s).
[08:01:48] Query finished in 0.004 second(s).Rows affected: 1
[08:02:35] Query finished in 0.005 second(s).Rows affected: 1
[08:03:22] Query finished in 0.003 second(s).Rows affected: 1
[08:03:29] Saved SQL contents to file: /Users/surekhakumari/Documents/Sprin
[08:30:49] Saved SQL contents to file: /Users/surekhakumari/Documents/Sprin
[17:55:38] Query finished in 0.001 second(s).
[17:59:10] Error while executing SQL query on database 'db_proj_re': near "AS
[18:22:16] Error while executing SQL query on database 'db_proj_re': near "co
[18:31:20] Query finished in 0.004 second(s).
[18:53:17] Query finished in 0.002 second(s).
[18:54:30] Query finished in 0.000 second(s).
[18:56:23] Query finished in 0.002 second(s).
```

```
                    reg_no,
                    faa_no,
                    date,
                    technician_hour,
                    score
                )
                VALUES (
                    '000000006',
                    '6',
                    '3',
                    '2022-01-26',
                    '5.7',
                    '87'
                );


select * from expert_in;
```

Total rows loaded: 3

| | model_no | ssn |
|---|---|---|
| 1 | Pilatus PC-12 NG | 000000004 |
| 2 | Beechcraft G36 Bonanza | 000000005 |
| 3 | Mooney M20 Acclaim Ultra | 000000006 |

```
[07:54:18] Saved SQL contents to file: /Users/surekhakumari/Documents/S
[07:56:25] Query finished in 0.011 second(s).Rows affected: 1
[07:56:52] Query finished in 0.003 second(s).Rows affected: 1
[07:57:18] Query finished in 0.004 second(s).Rows affected: 1
[07:57:49] Saved SQL contents to file: /Users/surekhakumari/Documents/S
[07:59:57] Query finished in 0.001 second(s).
[08:01:48] Query finished in 0.004 second(s).Rows affected: 1
[08:02:35] Query finished in 0.005 second(s).Rows affected: 1
[08:03:22] Query finished in 0.003 second(s).Rows affected: 1
[08:03:29] Saved SQL contents to file: /Users/surekhakumari/Documents/S
[08:30:49] Saved SQL contents to file: /Users/surekhakumari/Documents/S
[17:55:38] Query finished in 0.001 second(s).
[17:59:10] Error while executing SQL query on database 'db_proj_re': near '
[18:22:16] Error while executing SQL query on database 'db_proj_re': near '
[18:31:20] Query finished in 0.004 second(s).
[18:53:17] Query finished in 0.002 second(s).
[18:54:30] Query finished in 0.000 second(s).
[18:56:23] Query finished in 0.002 second(s).
[18:57:03] Query finished in 0.001 second(s).
```

```
[07:54:18] Saved SQL contents to file: /Users/surekhakumari/Documents/Spring
[07:56:25] Query finished in 0.011 second(s).Rows affected: 1
[07:56:52] Query finished in 0.003 second(s).Rows affected: 1
[07:57:18] Query finished in 0.004 second(s).Rows affected: 1
[07:57:49] Saved SQL contents to file: /Users/surekhakumari/Documents/Spring
[07:59:57] Query finished in 0.001 second(s).
[08:01:48] Query finished in 0.004 second(s).Rows affected: 1
[08:02:35] Query finished in 0.005 second(s).Rows affected: 1
[08:03:22] Query finished in 0.003 second(s).Rows affected: 1
[08:03:29] Saved SQL contents to file: /Users/surekhakumari/Documents/Spring
[08:30:49] Saved SQL contents to file: /Users/surekhakumari/Documents/Spring
[17:55:38] Query finished in 0.001 second(s).
[17:59:10] Error while executing SQL query on database 'db_proj_re': near "ASS
[18:22:16] Error while executing SQL query on database 'db_proj_re': near "con
[18:31:20] Query finished in 0.004 second(s).
[18:53:17] Query finished in 0.002 second(s).
[18:54:30] Query finished in 0.000 second(s).
[18:56:23] Query finished in 0.002 second(s).
[18:57:03] Query finished in 0.001 second(s).
```

```
458                     reg_no,
459                     faa_no,
460                     date,
461                     technician_hour,
462                     score
463                 )
464                 VALUES (
465                     '000000006',
466                     '6',
467                     '3',
468                     '2022-01-26',
469                     '5.7',
470                     '87'
471                 );
472
473
474 select * from faa_test;
475
476
```

Total rows loaded: 3

| | faa_no | name | max_score |
|---|---|---|---|
| 1 | 1 | AMA | 100 |
| 2 | 2 | AMG | 100 |
| 3 | 3 | IAR | 100 |

```
[07:56:25]  Query finished in 0.011 second(s).Rows affected: 1
[07:56:52]  Query finished in 0.003 second(s).Rows affected: 1
[07:57:18]  Query finished in 0.004 second(s).Rows affected: 1
[07:57:49]  Saved SQL contents to file: /Users/surekhakumari/Documents/Spring 22
[07:59:57]  Query finished in 0.001 second(s).
[08:01:48]  Query finished in 0.004 second(s).Rows affected: 1
[08:02:35]  Query finished in 0.005 second(s).Rows affected: 1
[08:03:22]  Query finished in 0.003 second(s).Rows affected: 1
[08:03:29]  Saved SQL contents to file: /Users/surekhakumari/Documents/Spring 22
[08:30:49]  Saved SQL contents to file: /Users/surekhakumari/Documents/Spring 22
[17:55:38]  Query finished in 0.001 second(s).
[17:59:10]  Error while executing SQL query on database 'db_proj_re': near "ASSERT
[18:22:16]  Error while executing SQL query on database 'db_proj_re': near "constra
[18:31:20]  Query finished in 0.004 second(s).
[18:53:17]  Query finished in 0.002 second(s).
[18:54:30]  Query finished in 0.000 second(s).
[18:56:23]  Query finished in 0.002 second(s).
[18:57:03]  Query finished in 0.001 second(s).
[18:57:46]  Cannot execute empty query.
```

```
[07:56:52]  Query finished in 0.003 second(s).Rows affected: 1
[07:57:18]  Query finished in 0.004 second(s).Rows affected: 1
[07:57:49]  Saved SQL contents to file: /Users/surekhakumari/Documer
[07:59:57]  Query finished in 0.001 second(s).
[08:01:48]  Query finished in 0.004 second(s).Rows affected: 1
[08:02:35]  Query finished in 0.005 second(s).Rows affected: 1
[08:03:22]  Query finished in 0.003 second(s).Rows affected: 1
[08:03:29]  Saved SQL contents to file: /Users/surekhakumari/Documer
[08:30:49]  Saved SQL contents to file: /Users/surekhakumari/Documer
[17:55:38]  Query finished in 0.001 second(s).
[17:59:10]  Error while executing SQL query on database 'db_proj_re': r
[18:22:16]  Error while executing SQL query on database 'db_proj_re': r
[18:31:20]  Query finished in 0.004 second(s).
[18:53:17]  Query finished in 0.002 second(s).
[18:54:30]  Query finished in 0.000 second(s).
[18:56:23]  Query finished in 0.002 second(s).
[18:57:03]  Query finished in 0.001 second(s).
[18:57:46]  Cannot execute empty query.
[18:59:54]  Query finished in 0.001 second(s).
```

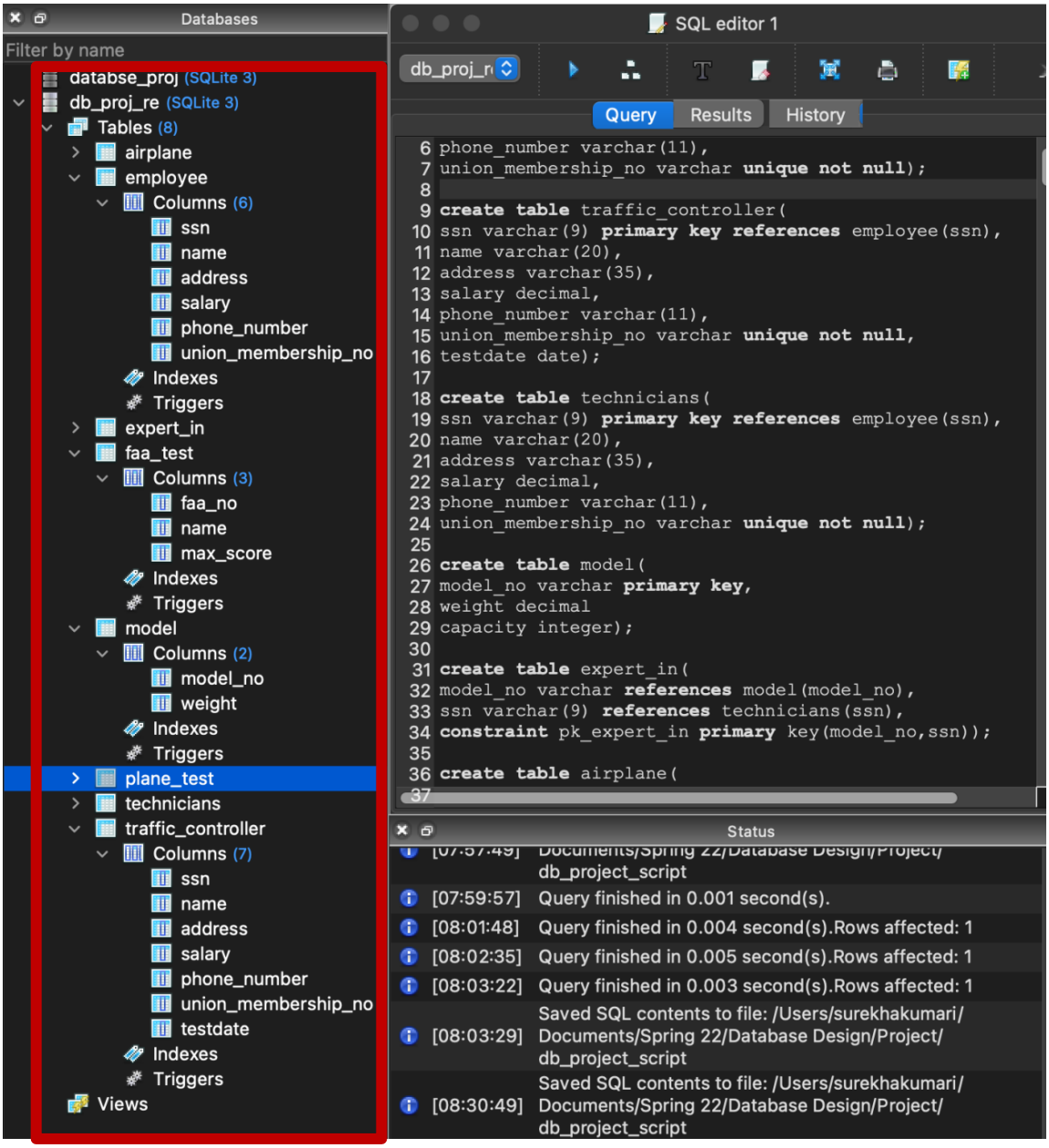
```
458                     reg_no,
459                     faa_no,
460                     date,
461                     technician_hour,
462                     score
463                 )
464                 VALUES (
465                     '000000006',
466                     '6',
467                     '3',
468                     '2022-01-26',
469                     '5.7',
470                     '87'
471                 );
472
473
474 select * from model;
475
476
```

Total rows loaded: 3

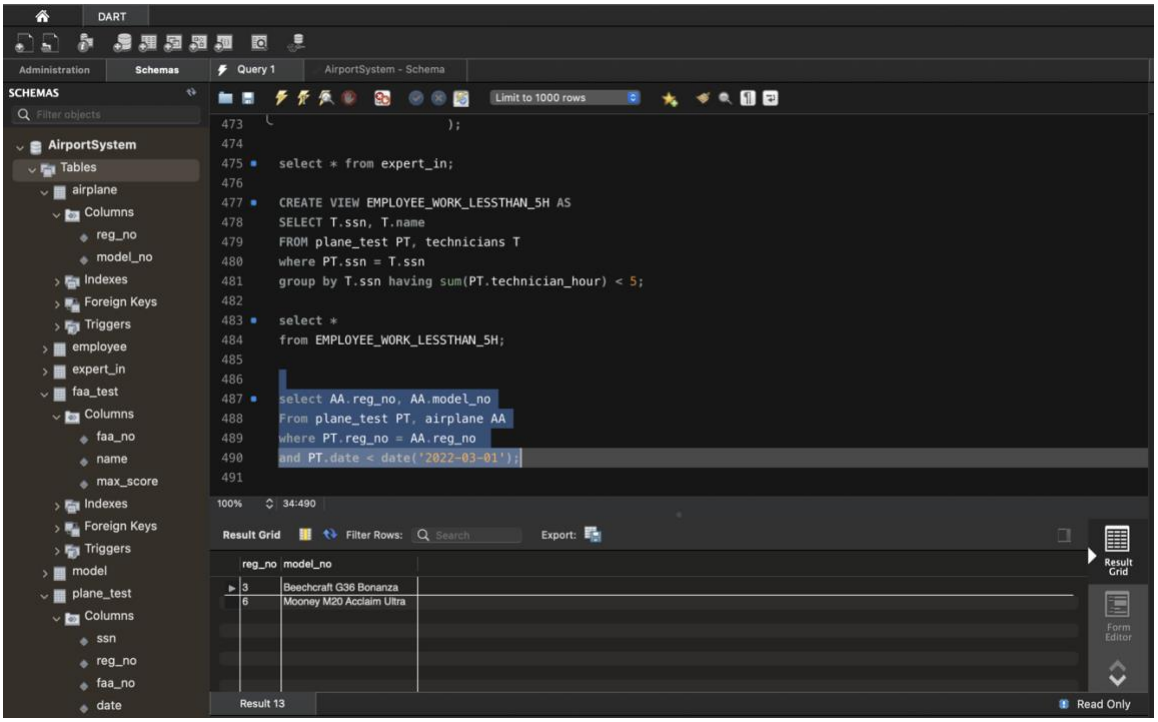
| | model_no | weight |
|---|---|---|
| 1 | Beechcraft G36 Bonanza | 12500 |
| 2 | Mooney M20 Acclaim Ultra | 11700 |
| 3 | Pilatus PC-12 NG | 11230 |

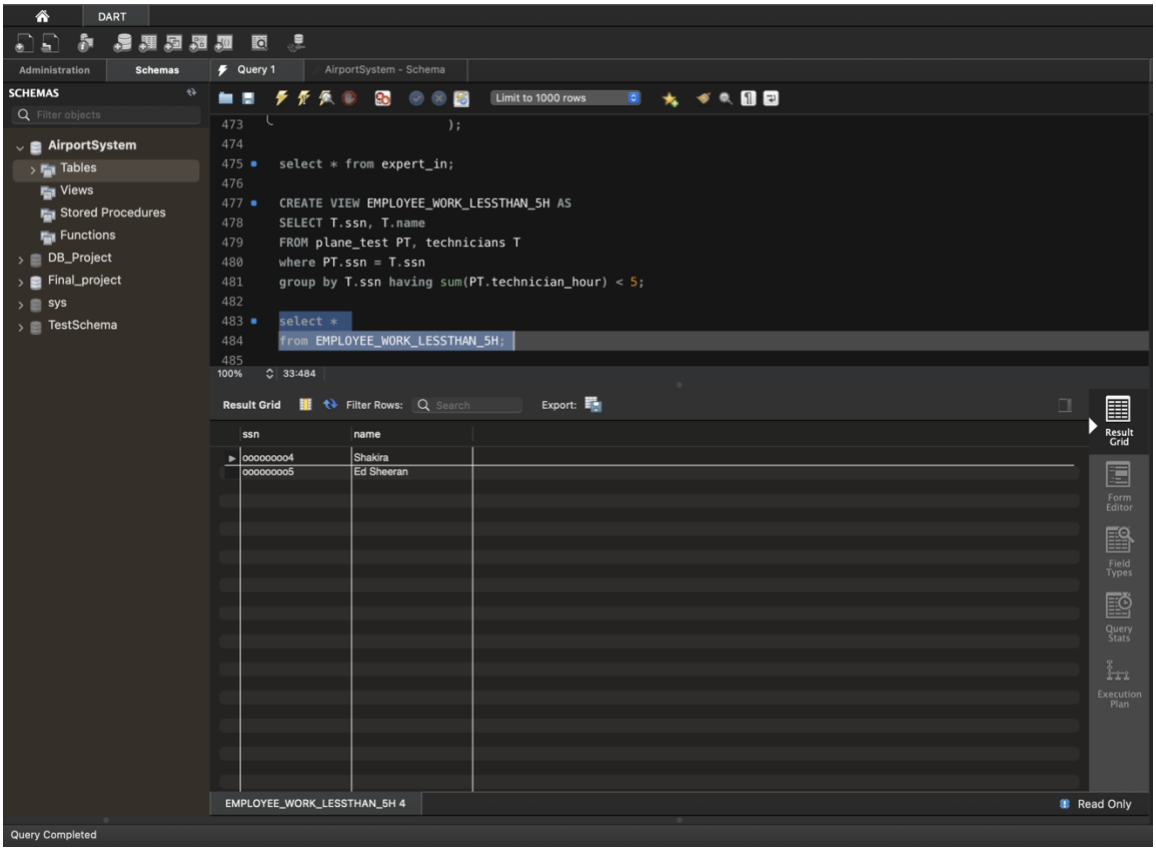
```
[07:57:18]  Query finished in 0.004 second(s).Rows affected: 1
[07:57:49]  Saved SQL contents to file: /Users/surekhakumari/Docum
[07:59:57]  Query finished in 0.001 second(s).
[08:01:48]  Query finished in 0.004 second(s).Rows affected: 1
[08:02:35]  Query finished in 0.005 second(s).Rows affected: 1
[08:03:22]  Query finished in 0.003 second(s).Rows affected: 1
[08:03:29]  Saved SQL contents to file: /Users/surekhakumari/Docum
[08:30:49]  Saved SQL contents to file: /Users/surekhakumari/Docum
[17:55:38]  Query finished in 0.001 second(s).
[17:59:10]  Error while executing SQL query on database 'db_proj_re
[18:22:16]  Error while executing SQL query on database 'db_proj_re
[18:31:20]  Query finished in 0.004 second(s).
[18:53:17]  Query finished in 0.002 second(s).
[18:54:30]  Query finished in 0.000 second(s).
[18:56:23]  Query finished in 0.002 second(s).
[18:57:03]  Query finished in 0.001 second(s).
[18:57:46]  Cannot execute empty query.
[18:59:54]  Query finished in 0.001 second(s).
[19:00:20]  Query finished in 0.001 second(s).
```

```
[07:57:18]  Query finished in 0.004 second(s).Rows affected: 1
[07:57:49]  Saved SQL contents to file: /Users/surekhakumari/D
[07:59:57]  Query finished in 0.001 second(s).
[08:01:48]  Query finished in 0.004 second(s).Rows affected: 1
[08:02:35]  Query finished in 0.005 second(s).Rows affected: 1
[08:03:22]  Query finished in 0.003 second(s).Rows affected: 1
[08:03:29]  Saved SQL contents to file: /Users/surekhakumari/D
[08:30:49]  Saved SQL contents to file: /Users/surekhakumari/D
[17:55:38]  Query finished in 0.001 second(s).
[17:59:10]  Error while executing SQL query on database 'db_pr
[18:22:16]  Error while executing SQL query on database 'db_pr
[18:31:20]  Query finished in 0.004 second(s).
[18:53:17]  Query finished in 0.002 second(s).
[18:54:30]  Query finished in 0.000 second(s).
[18:56:23]  Query finished in 0.002 second(s).
[18:57:03]  Query finished in 0.001 second(s).
[18:57:46]  Cannot execute empty query.
[18:59:54]  Query finished in 0.001 second(s).
[19:00:20]  Query finished in 0.001 second(s).
```


Databases
Filter by name
database_proj (SQLite 3)
db_proj_re (SQLite 3)
Tables (8)
airplane
employee
Columns (6)
ssn
name
address
salary
phone_number
union_membership_no
Indexes
Triggers
expert_in
faa_test
Columns (3)
faa_no
name
max_score
Indexes
Triggers
model
Columns (2)
model_no
weight
Indexes
Triggers
plane_test
technicians
traffic_controller
Columns (7)
ssn
name
address
salary
phone_number
union_membership_no
testdate
Indexes
Triggers
Views
SQL editor 1
db_proj_r
Query
Results
History
6 phone_number varchar(11),
7 union_membership_no varchar unique not null);
8
9 create table traffic_controller(
10 ssn varchar(9) primary key references employee(ssn),
11 name varchar(20),
12 address varchar(35),
13 salary decimal,
14 phone_number varchar(11),
15 union_membership_no varchar unique not null,
16 testdate date);
17
18 create table technicians(
19 ssn varchar(9) primary key references employee(ssn),
20 name varchar(20),
21 address varchar(35),
22 salary decimal,
23 phone_number varchar(11),
24 union_membership_no varchar unique not null);
25
26 create table model(
27 model_no varchar primary key,
28 weight decimal
29 capacity integer);
30
31 create table expert_in(
32 model_no varchar references model(model_no),
33 ssn varchar(9) references technicians(ssn),
34 constraint pk_expert_in primary key(model_no,ssn));
35
36 create table airplane(
37
Status
db_project_script
[07:59:57] Query finished in 0.001 second(s).
[08:01:48] Query finished in 0.004 second(s).Rows affected: 1
[08:02:35] Query finished in 0.005 second(s).Rows affected: 1
[08:03:22] Query finished in 0.003 second(s).Rows affected: 1
[08:03:29] Saved SQL contents to file: /Users/surekhakumari/ Documents/Spring 22/Database Design/Project/ db_project_script
[08:30:49] Saved SQL contents to file: /Users/surekhakumari/ Documents/Spring 22/Database Design/Project/ db_project_script

# 14. Additional Queries and Views



**View to find technicians who have worked less than five hours:**

```
CREATE VIEW EMPLOYEE_WORK_LESSTHAN_5H AS
SELECT T.ssn, T.name
FROM plane_test PT, technicians T
where PT.ssn = T.ssn
group by T.ssn having sum(PT.technician_hour) < 5;

select *
from EMPLOYEE_WORK_LESSTHAN_5H;
```

**Query to find airplanes which haven't gotten tested since March:**

```
select AA.reg_no, AA.model_no
From plane_test PT, airplane AA
where PT.reg_no = AA.reg_no
and PT.date < date('2022-03-01');
```

**Trigger to update employee table when traffic controller table is updated:**

```
create trigger update_employee_from_trafficcontroller
after insert on traffic_controller
referencing new table newtrafficcontrollers
for each statement
insert
into employee(ssn, name, address, salary, phone_number, union_membership_no)
select  ssn, name, address, salary, phone_number, union_membership_no
from newtrafficcontrollers
```

**Trigger to update employee table when technicians table is updated:**

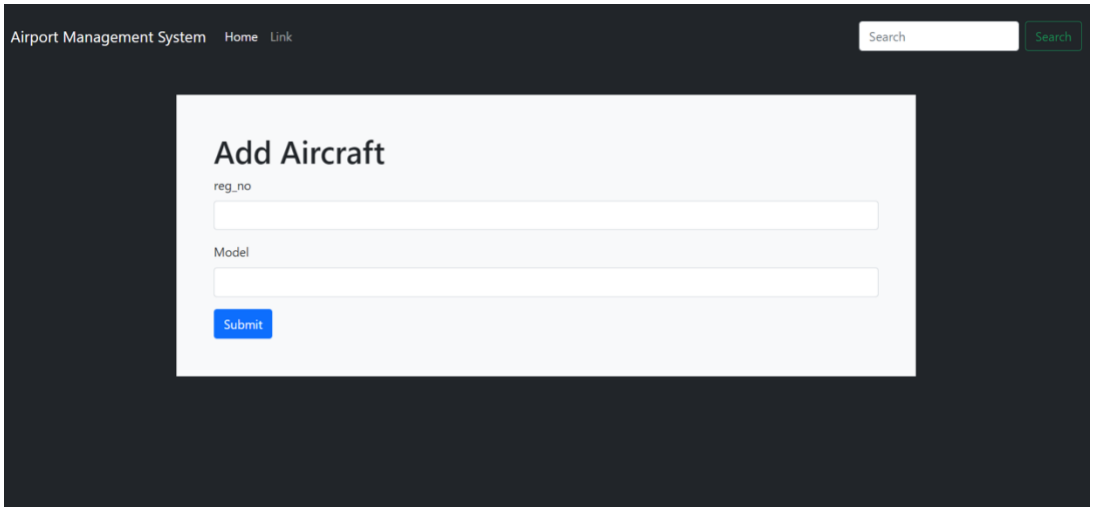
```
create trigger update_employee_from_technicians
after insert on technicians
referencing new table newtechnicians
for each statement
insert
into employee(ssn, name, address, salary, phone_number, union_membership_no)
select  ssn, name, address, salary, phone_number, union_membership_no
from newtechnicians
```

## 15. User application interface

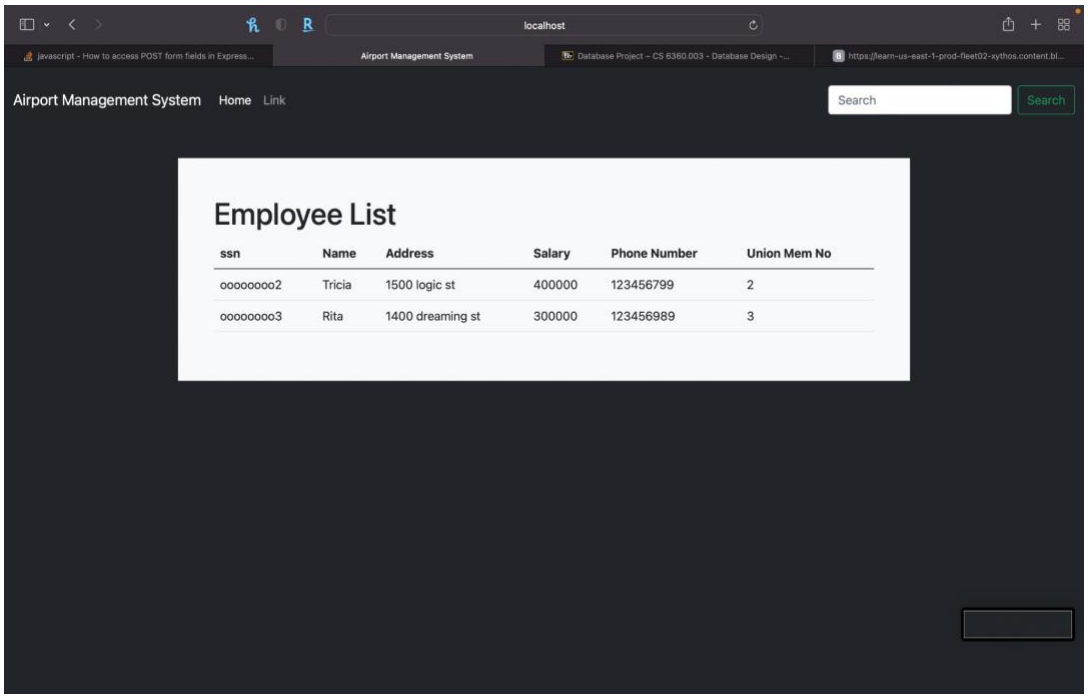
The user can interact with the database via web application. This web application is developed using the Express js and node js. The front end is developed using HTML, CSS, and bootstrap. The backend is developed using the Express js and ejs view engine. Various node modules are used to provide the required functionalities.

The options (functions) provided to users are (CRD):

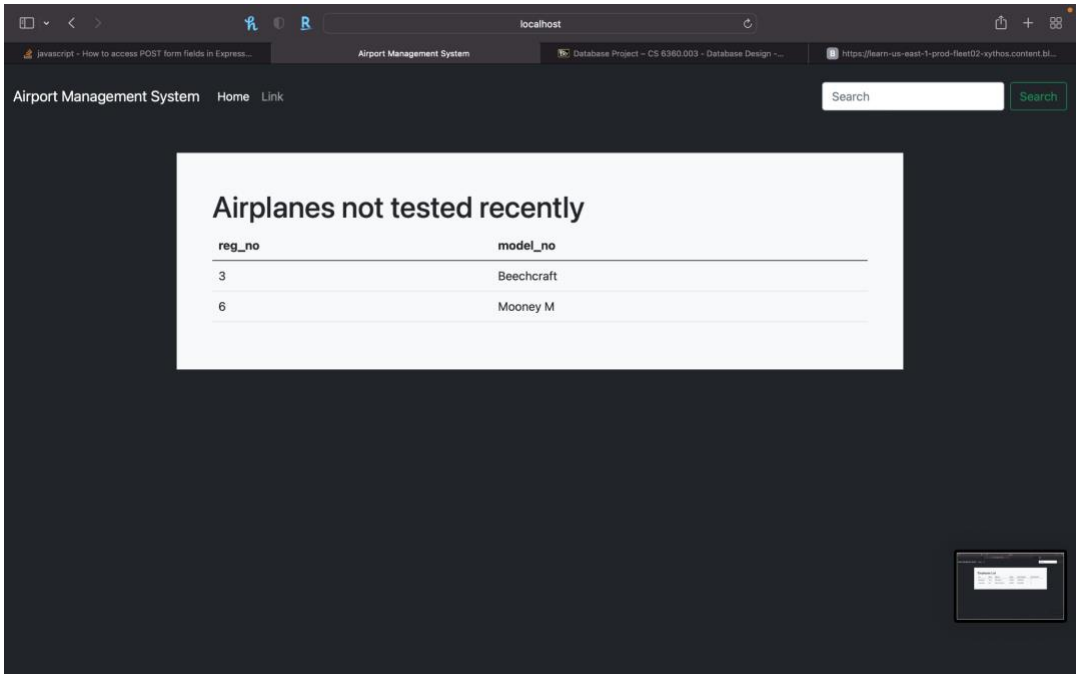
Add new Aircraft Model



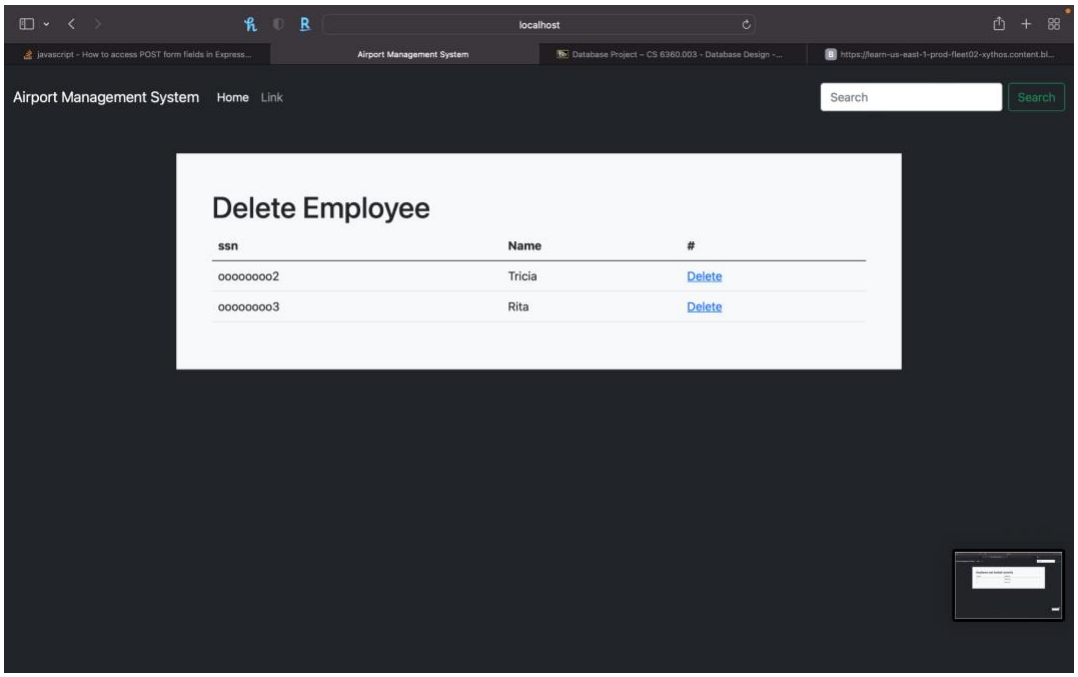

View all employee



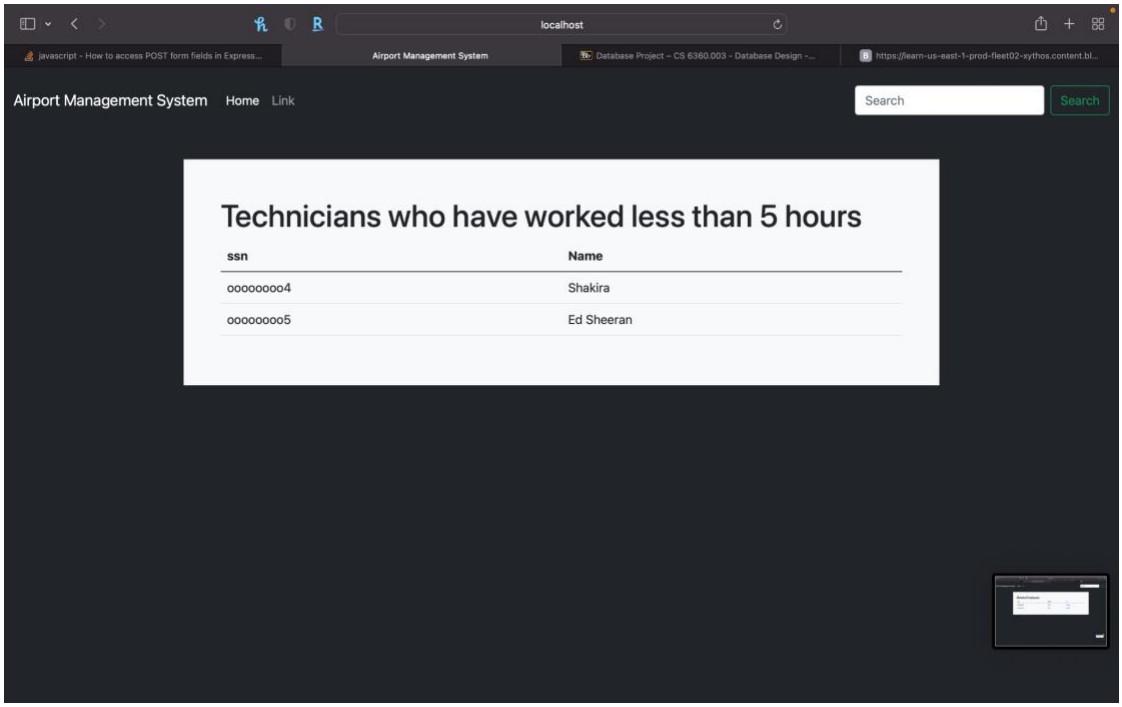
View not tested aircrafts




Delete employee



View technicians who have worked less than five hours




Once the node server and SQL server are running, the user can access the web application by visiting the hostel URL.

For now, the application can be hosted on local server is accessible ay reaching localhost URL: “http://localhost:3000/home”.

The Express JS backend is connected to the SQL server via node module “mysql2”. The connection credentials are presented in the JS files and SQL queries are executed using the “mysql2” module.

## 16. Conclusions and Future Work

Efficient mechanism to store and organize data than spreadsheets. Allows for a centralized facility that can easily be modified and quickly shared among multiple users. Having a web based front end removes the requirement of users having to understand a database and allows users to connect from anywhere with an internet connection and a basic web browser. Reduces human intervention in storing the records of items and provides a secure way of data storage by reducing the errors.

For future work, the system can be further implemented to enable more functions, the database can be extended to support commercial aspects of the airport, login, airlines etc. On the user side it can be extended for login and other controls, more functionalities can be added to enable user role specific functions, access to the functionality, security etc. The System can be improvised to include more elements other than airplanes and employees. For instance, functions like handling visiting passenger details, pilots, and elements like airlines.

Summing up, the Online Airport System is a carefully designed system to support current requirements and enable future functionalities. Despite the difficulty of building an organized system, in the long run it will reduce manual effort of keeping and updating real time data. The system will make it faster to process, search and manipulate or update records and will reduce human error.

## 17. References


1) *Fundamentals of Database Systems, 7th Edition*, R. Elmasri and S. B. Navathe, Addison Wesley. ISBN-13: 978-0133970777
2) *Node.js Express FrameWork Tutorial - Learn in 10 Minutes*, James Hartman, https://www.guru99.com/node-js-express.html
3) *Express - Fast, unopinionated, minimalist web framework for Node.js*, https://expressjs.com/
4) *MySQL Tutorial*, https://www.mysqltutorial.org/
5) *MySQL Tutorial*, w3schools, https://www.w3schools.com/mySQl/default.asp
6) *Node.js Tutorial*, w3schools, https://www.w3schools.com/nodejs/
7) *HTML Tutorial*, w3schools, https://www.w3schools.com/html/
8) *Bootstrap 3 Tutorial*, w3schools, https://www.w3schools.com/bootstrap/
9) *Node.js MySQL tutorial: a step-by-step getting started guide with Express js REST API*, Geshan Manandhar, https://geshan.com.np/blog/2020/11/nodejs-mysql-tutorial/