+

Database Management CS 4342 / CS 5342

Spring 2023 Semester

Assignment 3

Clean Services

The 8th Team

Dario Vazquez,
Christopher Ibarra,
Priscila Vargas,
Taposh Sarker

March 5th, 2023

Table ofContents

1. SCOPE

The database system will be used by a company that provides aircraft cleaning services. The purpose of the database system is to keep track of the services provided to different airports by the employees of the company. The system should keep track of the personnel information. Each worker has a unique social security number. They also have a name (first name and last name), birth date, gender, phone number, which can be multiple, salary per hour, and a status (available, unavailable or in service). Each group has 7 workers including one supervisor. The group (AKA team) cannot work with less than 7 employees. Only the workers with available status will be assigned to one group by the company. The system should keep track of the groups and the cleaning services each provide. The group has a unique composite location (airport code, gate), a status (in service/not in service), and a unique code number. A group is hired to work 8 hours a day. The system should keep track of the services provided by storing a unique service number, date, and time it took to provide the service. The system should keep track of the planes that have been cleaned, the planes have a type that falls into only one category (A, B, or C) according to the size of the plane from largest (A) to smallest (C). The planes can be in different gates, one at a time. Each plane's size is expected to be cleaned in a minimum and maximum time in minutes. Each plane has a unique code and composite location (airport code, gate).

The system should allow the general manager to change the workers to a different group and update the information. To retrieve the total number of employees and the employees that are available. The system should allow the general manager to know the location of a group, the planes that have been cleaned, when and by which group, who is the supervisor, and the groups that are in service. The system should allow the general manager to know what groups took more time than expected to clean a plane and to do reports for a specified period to improve the services provided.

Note: The scope was created by Maria and refined by all members of the team.

2. REQUIREMENTS

Vazquez - Design Lead:

R1. The system should keep track of the personnel information: a unique social security number, name (first name and last name), birth date, gender, phone number, and salary per hour.

R2. The system should not allow employees to be part of more than one group at a time.

R3. The system shall generate a report to keep track of the plane's location via airport code and gate.

R4. The system shall generate a report containing the planes that were cleaned for a specific time period.

R5. The system should only allow one employee to supervise a group at a given time.

Vargas SQL Lead:

R6. The system should allow the general manager to generate a list of the services provided per plane, airport, group for a specified period.

R7. The system should keep track of the services: service number, plane code, airport code, date, time it took to fulfill the service, and group code.

R8. The system should allow the general manager to generate information to know the total number of employees and the employees that are available at a specific moment.

R9. The system should keep track of each group, its status, supervisor, location and code.

R10. The system should allow us to update the group membership of the worker.

R11 When assigned to a group the status of a worker is changed to in service and the status of an in service worker cannot be changed to unavailable.

Sarker - Interface Lead:

R12 The system will allow update of which airport and plane is assigned to each group.

R13. The system should generate a report of which group cleaned which plane at which airport at which specific time.

R14. The system shall keep records of the exact location (airport, plane, gate) of a group while in-service.

R15. The system shall keep track of which groups are in-service at a particular point of time.

R16. The system will make sure that a group is assigned to only one plane at a time.

R17. The system will allow the general manager to generate a list of service provided by airport.

Ibarra - Report Lead:

R18. The system will keep track of the planes' information: category (A, B, C), expected time for it to be cleaned (min and max time), a unique code and compose location (airport code, gate).

R19. The system will generate a report of the groups that took longer than expected to clean a plane.

R20. The system will make sure that each group has 7 members for it to be considered operational.

R21. The system will make sure that each group has a supervisor.

R22. The system will make sure that no two or more employees share the same social security number.

Note: Requirements were created by each team member.

b. Assumptions

A1. We assume that there are three types of cleaning services according to the plane size.

A2. We assume that the min and max time to clean a plane depends on the plane size (A, B, or C).

A3. We assume that by comparing the service duration with the min and max time for cleaning a plane we will be able to know what groups took longer than the max time to clean a plane.

A4. We assume that all the workers of the same group have the same location (airport code and gate).

A5. We assume that the total number of workers can change.

A6. We are assuming that the general manager will be the person in charge of retrieving information and generating reports from the database.

Note: In the subsequent sections, “group” will be referred to as “team” due to “group” being a reserved keyword in sql.

3. E/R DIAGRAM

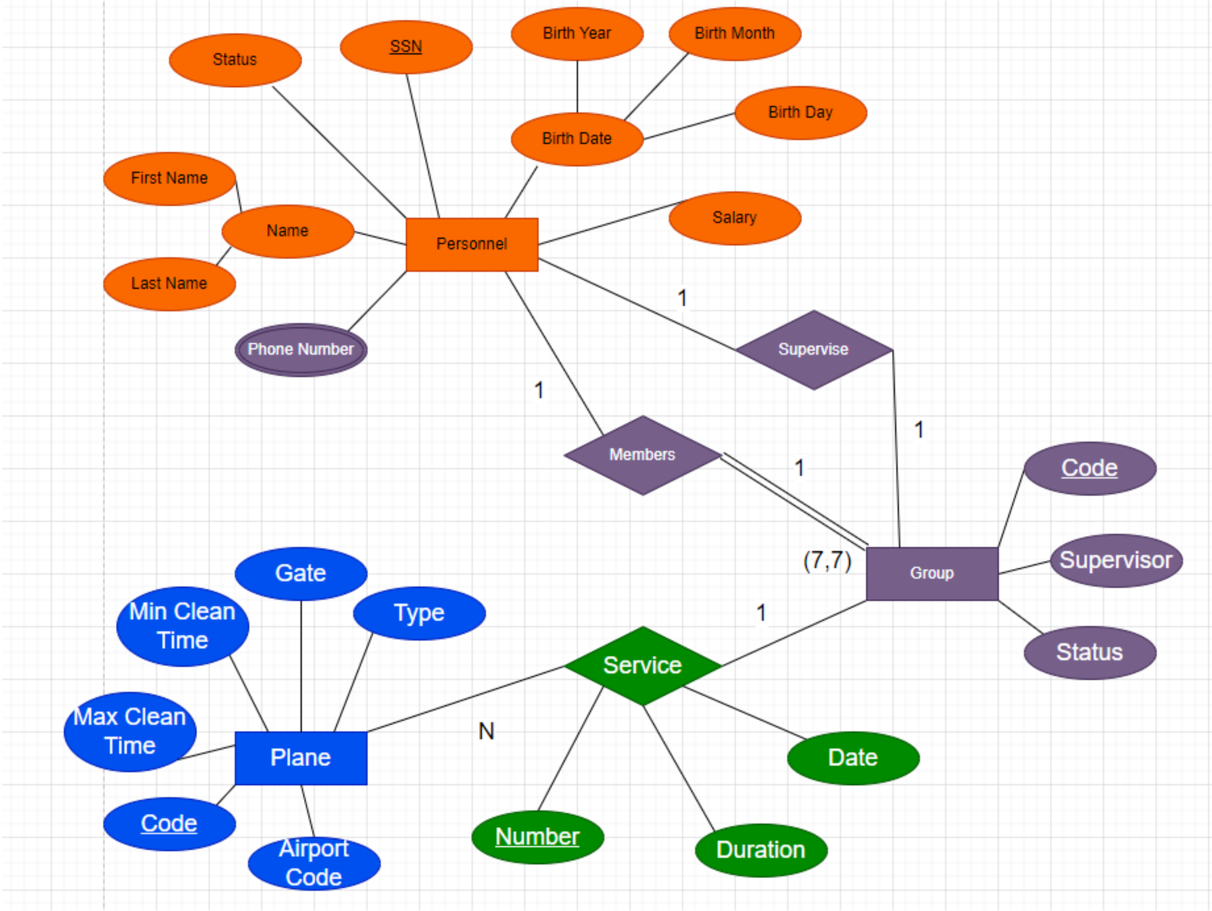


Figure 1 E-R Diagram

Note: The E/R Diagram components were designed by the following team members as follows:
Orange Components – Christopher Ibarra
Blue Components – Maria Vargas
Green Components – Dario Vazquez
Purple Components – Taposh Sarker

4. RELATIONAL MODEL

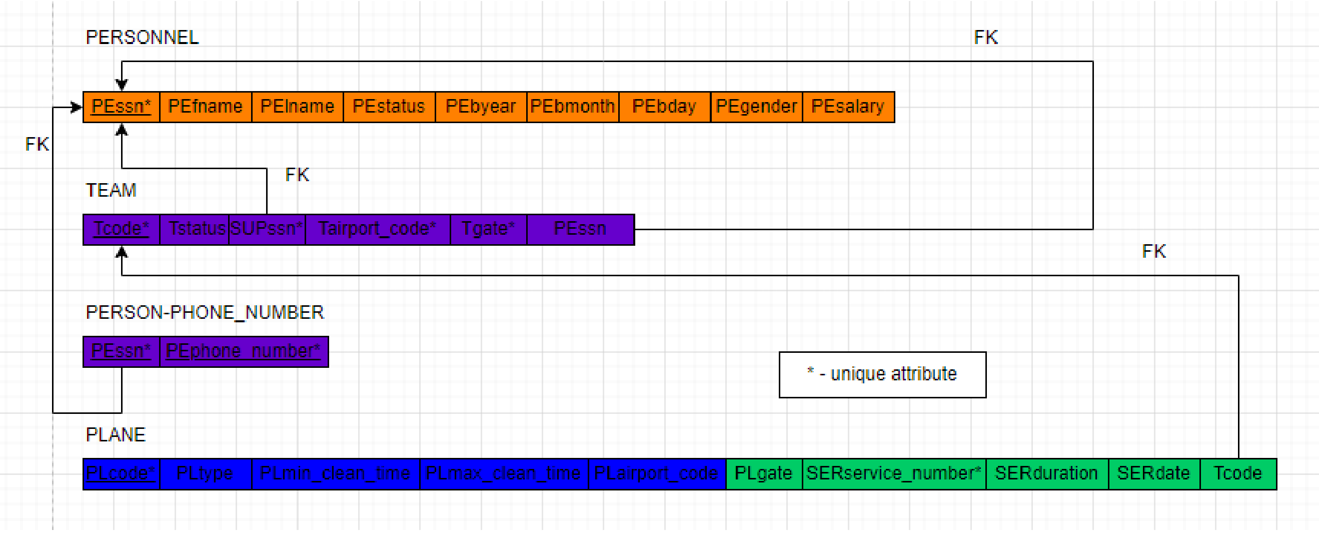

PERSONNEL
FK
PEssn* | PEfname | PElname | PEstatus | PEbyear | PEbmonth | PEbday | PEgender | PEsalary
FK
TEAM
FK
Tcode* | Tstatus | SUPssn* | Tairport_code* | Tgate* | PEssn
FK
PERSON-PHONE_NUMBER
PEssn* | PEphone_number*
* - unique attribute
PLANE
PLcode* | PLtype | PLmin_clean_time | PLmax_clean_time | PLairport_code | PLgate | SERservice_number* | SERduration | SERdate | Tcode

Figure 2 Un-normalized Relational Schema (Note: SUPssn - Superviso SSN and PARTssn - Part of SSN)

Note: The Relational Model Schema components were designed by the following team members as follows:
Orange Components – Taposh,
Purple Components – Chris,
Blue Components – Maria.
Green Components – Dario.

Diagram put together by Chris.

5. NORMALIZED SCHEMA

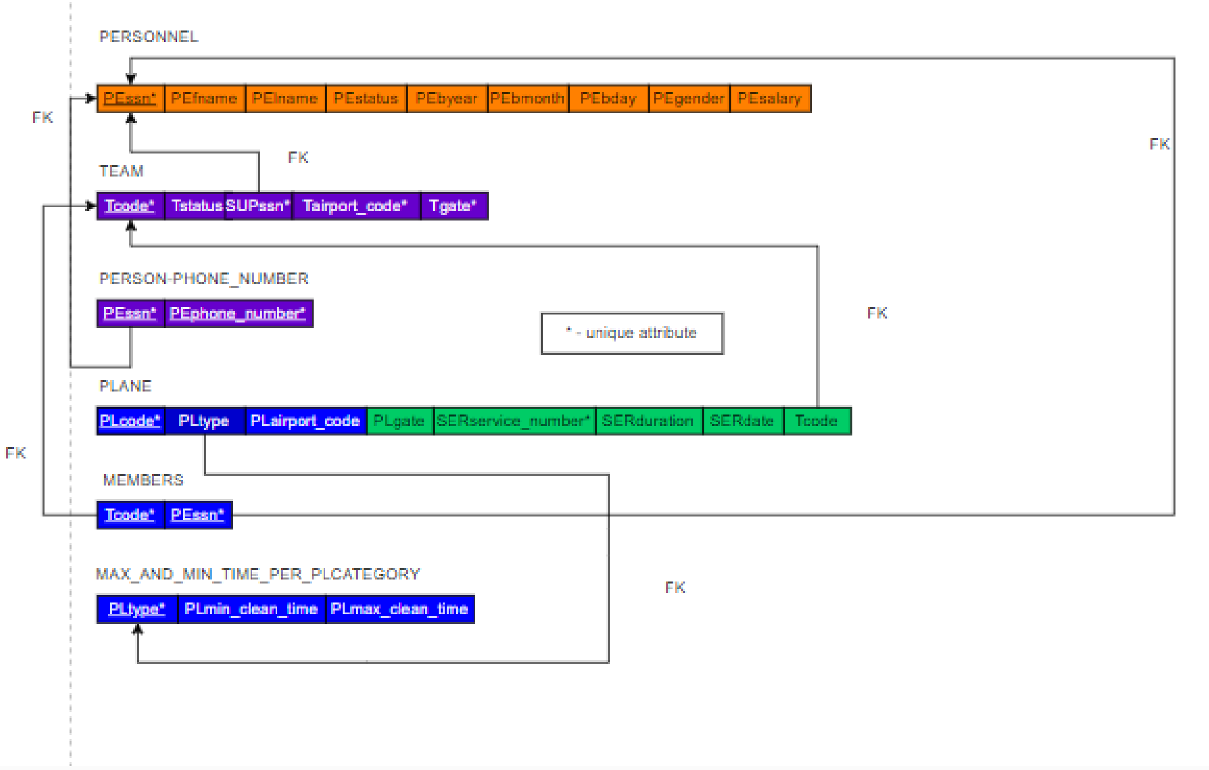


Figure 3 Normalized Relational Schema

Note: Relational diagram normalized by Taposh, Maria, Dario and Chris.

PLmax_clean_time and PLmin_clean_time have a functional dependencie 3, because they depend on a not primary key attribute, PLtype. To remove that functional dependency 3, we will create a table for these attributes depending on PLtype which will be the primary key in this new table.

The team table had a functional dependency 1 since there are multiple PEssn per Tcode which is not allowed since Tcode is the primary key. To remedy this, we created a new table members to keep track of what members are in a given team.

Functional dependencies:

PERSONNEL Relation:
FD1 (trivial) = {PEssn} -> {PEssn,PEstatus, PEfname, PElname, PEbyear, PEbmonth, PEbday, PEsalary, PEgender}

TEAM_V2 Relation:
FD1 (trivial) = {TCode} -> {TCode, Tstatus, Tairport_code, Tgate, SUPssn}

PERSON-PHONE_NUMBER Relation:
FD1 (trivial) = {PEssn,PEphone_number} -> {PEssn,PEphone_number}

PLANE Relation:
FD1 (trivial) = {PLCode} -> {PLCode, PLtype, PLgate, PLairport_code, SERservice_number, SERduration, SERdate, TCode}

MEMBERS Relation:
FD1 (trivial) = {Tcode,PEssn} -> {Tcode,PEssn}

MAX_AND_MIN_TIME_PER_PLCATEGORY Relation:
FD1 (trivial) = {PLtype} -> {PLtype, PLmin_clean_time, PLmax_clean_time}

Note: Functional Dependencies by Taposh and Maria.

Explanations for Functional Dependencies:

Normalization:
The PERSONNEL relation is in First Normal Form (1NF) because all its attributes are atomic. The PERSONNEL relation is in Second Normal Form (2NF) because all the non-prime attributes depend fully on the PK.
The PERSONNEL relation is in Third Normal Form (3NF) because none of the non-prime attributes depend transitively on the PK or none of the non-prime attributes depend on another non-prime attribute.

The TEAM_V2 relation is in First Normal Form (1NF) because all its attributes are atomic. The TEAM_V2 relation is in Second Normal Form (2NF) because all the non-prime attributes depend fully on the PK.
The TEAM_V2 relation is in Third Normal Form (3NF) because none of the non-prime attributes depend transitively on the PK or none of the non-prime attributes depend on another non-prime attribute.

The PERSON-PHONE_NUMBER relation is in First Normal Form (1NF) because all its attributes are atomic. The PERSON-PHONE relation is in Second Normal Form (2NF) because all the non-prime attributes depend fully on the PK.
The PERSON-PHONE_NUMBER relation is in Third Normal Form (3NF) because none of the non-prime attributes depend transitively on the PK or none of the non-prime attributes depend on another non-prime attribute.

The PLANE relation is in First Normal Form (1NF) because all its attributes are atomic. The PLANE relation is in Second Normal Form (2NF) because all the non-prime attributes depend fully on the PK.
The PLANE relation is in Third Normal Form (3NF) because none of the non-prime attributes depend transitively on the PK or none of the non-prime attributes depend on another non-prime attribute.

The MEMBERS is in First Normal Form (1NF) because all its attributes are atomic. The MEMBERS Prelation is in Second Normal Form (2NF) because all the non-prime attributes depend fully on the PK.
The MEMBERS relation is in Third Normal Form (3NF) because none of the non-prime attributes depend transitively on the PK or none of the non-prime attributes depend on another non-prime attribute.

The MAX_AND_MIN_TIME_PER_PLCATEGORY is in First Normal Form (1NF) because all its attributes are atomic. The MAX_AND_MIN_TIME_PER_PLCATEGORY relation is in Second Normal Form (2NF) because all the non-prime attributes depend fully on the PK.
The MAX_AND_MIN_TIME_PER_PLCATEGORY relation is in Third Normal Form (3NF) because none of the non-

prime attributes depend transitively on the PK or none of the non-prime attributes depend on another non-prime attribute.

Note: Explanations by Taposh and Maria.

6. MySQL SERVER

Note: Figure 4 shows the Create statements for the relations in section 5.

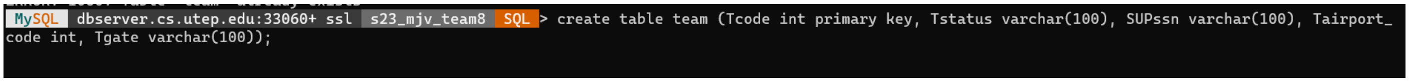
Chris did TEAM, PERSON-PHONE_NUMBER, and MAX_AND_MIN_TIME_PER_PLCATEGORY.

Maria did MEMBERS, PLANE, and PERSONNEL.

Dario and Taposh revised.

Table: TEAM

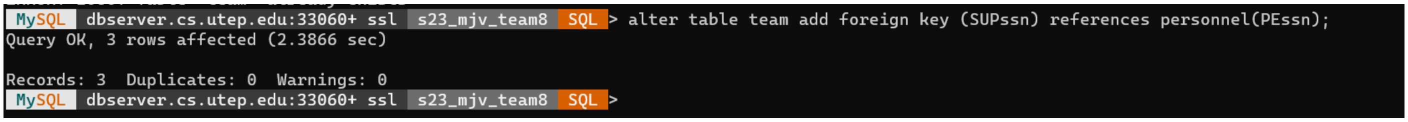
Table created:



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create table team (Tcode int primary key, Tstatus varchar(100), SUPssn varchar(100), Tairport_
code int, Tgate varchar(100));
```

Figure 4 Team Table Created

Foreign key added PEssn from PERSONNEL:



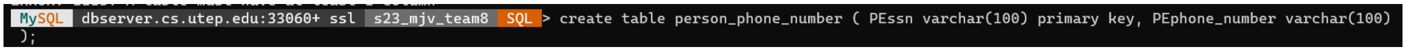
```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > alter table team add foreign key (SUPssn) references personnel(PEssn);
Query OK, 3 rows affected (2.3866 sec)

Records: 3  Duplicates: 0  Warnings: 0
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 5 Team Altered: added SUPssn as foreign key

*Note: Some screenshots may have ”team_v2” instead of ”team”. The table was renamed ”team” for simplicity at the end.

Table: PERSON-PHONE_NUMBER

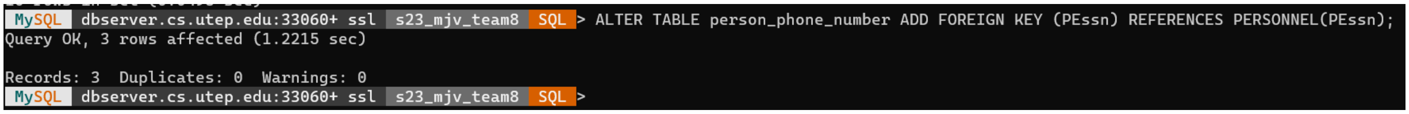
Table created:



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create table person_phone_number ( PEssn varchar(100) primary key, PEphone_number varchar(100)
 );
```

Figure 6 Person_phone_number Table Created

Foreign key added PEssn from PERSONNEL:

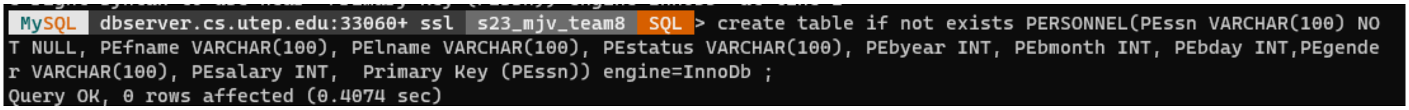


```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ALTER TABLE person_phone_number ADD FOREIGN KEY (PEssn) REFERENCES PERSONNEL(PEssn);
Query OK, 3 rows affected (1.2215 sec)

Records: 3  Duplicates: 0  Warnings: 0
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 7 Person_phone_number Altered: added PEssn as foreign key

Table: PERSONNEL

Table created:



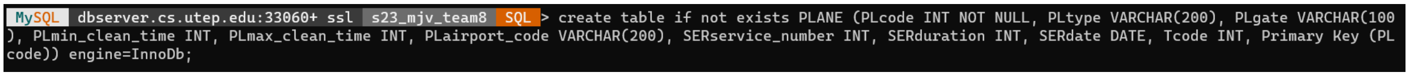
```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create table if not exists PERSONNEL(PEssn VARCHAR(100) NO
T NULL, PEfname VARCHAR(100), PElname VARCHAR(100), PEstatus VARCHAR(100), PEbyear INT, PEbmonth INT, PEbday INT,PEgende
r VARCHAR(100), PEsalary INT,  Primary Key (PEssn)) engine=InnoDb ;
Query OK, 0 rows affected (0.4074 sec)
```

Figure 8 Personnel Table Created

Foreign key added: (No foreign keys in PERSONNEL)

Table: PLANE

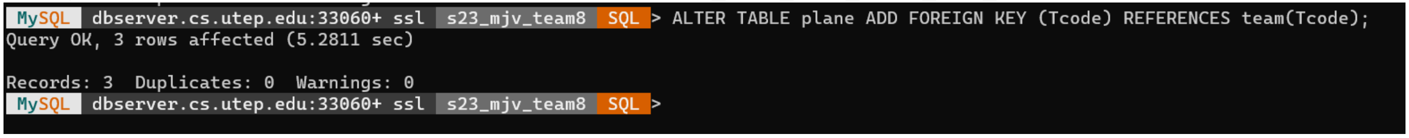
Table created:



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create table if not exists PLANE (PLcode INT NOT NULL, PLtype VARCHAR(200), PLgate VARCHAR(100
), PLmin_clean_time INT, PLmax_clean_time INT, PLairport_code VARCHAR(200), SERservice_number INT, SERduration INT, SERdate DATE, Tcode INT, Primary Key (PL
code)) engine=InnoDb;
```

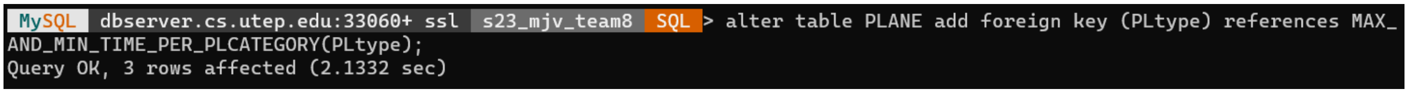
Figure 9 Plane Table Created

Foreign key added Tcode from TEAM and PLtype from MAX_AND_MIN_TIME_PER_PLCATEGORY:



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ALTER TABLE plane ADD FOREIGN KEY (Tcode) REFERENCES team(Tcode);
Query OK, 3 rows affected (5.2811 sec)

Records: 3  Duplicates: 0  Warnings: 0
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 10 Plane Table Altered: added Tcode as foreign key

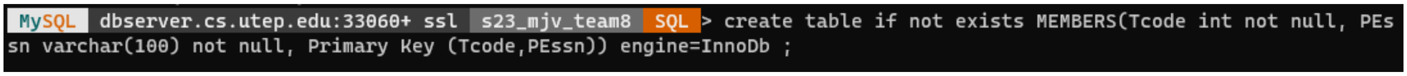


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > alter table PLANE add foreign key (PLtype) references MAX_
AND_MIN_TIME_PER_PLCATEGORY(PLtype);
Query OK, 3 rows affected (2.1332 sec)
```

Figure 11 Plane Table Altered: added PLtype as foreign key

Table: MEMBERS

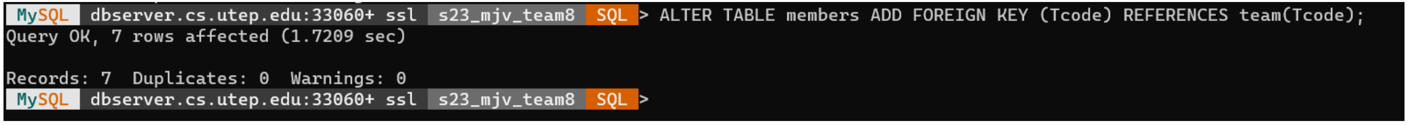
Table created:



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create table if not exists MEMBERS(Tcode int not null, PEs
sn varchar(100) not null, Primary Key (Tcode,PEssn)) engine=InnoDb ;
```

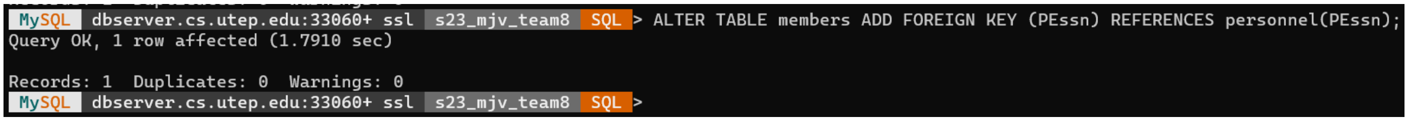
Figure 12 Members Table Created

Foreign keys added Tcode from TEAM and PEssn from PERSONNEL:



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ALTER TABLE members ADD FOREIGN KEY (Tcode) REFERENCES team(Tcode);
Query OK, 7 rows affected (1.7209 sec)

Records: 7  Duplicates: 0  Warnings: 0
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 13 Members Table Altered: added Tcode as foreign key

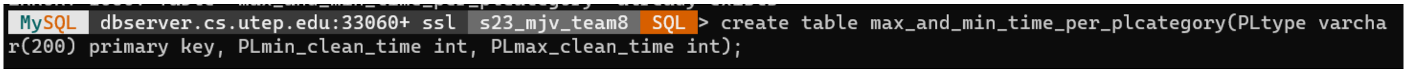


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ALTER TABLE members ADD FOREIGN KEY (PEssn) REFERENCES personnel(PEssn);
Query OK, 1 row affected (1.7910 sec)

Records: 1  Duplicates: 0  Warnings: 0
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 14 Members Table Altered: added PEssn as foreign key

Table: MAX_AND_MIN_TIME_PER_PLCATEGORY

Table created:



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create table max_and_min_time_per_plcategory(PLtype varcha
r(200) primary key, PLmin_clean_time int, PLmax_clean_time int);
```

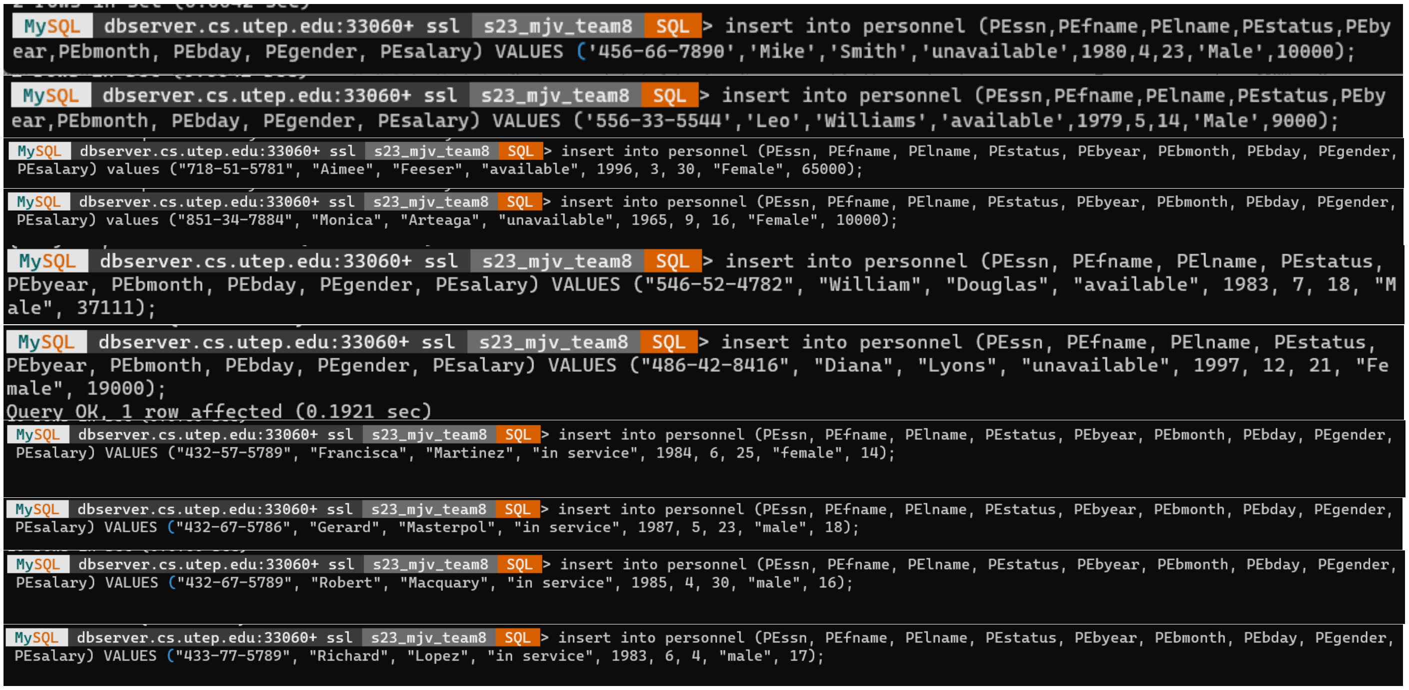
Figure 15 Max_and_min_time_per_plcategory Table Created

Figure 4. Screenshot of MySQL Statements to create tables.

7. DATABASE RECORDS (Section Done)

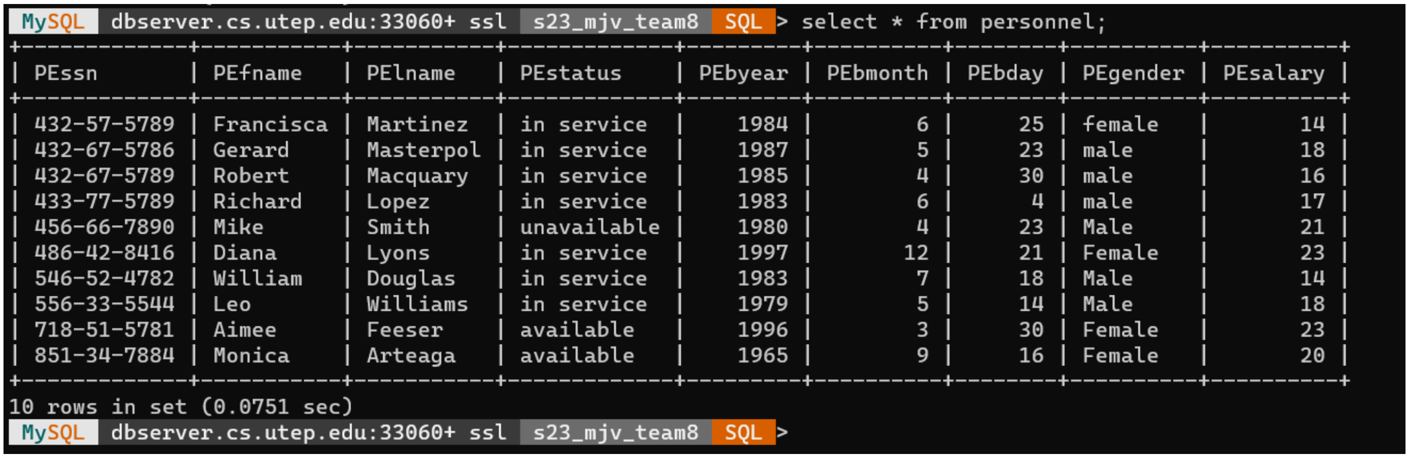
Table: PERSONNEL

Figure 5 shows the database records and insert statements for the Personnel table. These records were created by Maria, Chris and Taposh.



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn,PEfname,PElname,PEstatus,PEby
ear,PEbmonth, PEbday, PEgender, PEsalary) VALUES ('456-66-7890','Mike','Smith','unavailable',1980,4,23,'Male',10000);

MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn,PEfname,PElname,PEstatus,PEby
ear,PEbmonth, PEbday, PEgender, PEsalary) VALUES ('556-33-5544','Leo','Williams','available',1979,5,14,'Male',9000);

MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn, PEfname, PElname, PEstatus, PEbyear, PEbmonth, PEbday, PEgender,
 PEsalary) values ("718-51-5781", "Aimee", "Feeser", "available", 1996, 3, 30, "Female", 65000);

MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn, PEfname, PElname, PEstatus, PEbyear, PEbmonth, PEbday, PEgender,
 PEsalary) values ("851-34-7884", "Monica", "Arteaga", "unavailable", 1965, 9, 16, "Female", 10000);

MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn, PEfname, PElname, PEstatus,
PEbyear, PEbmonth, PEbday, PEgender, PEsalary) VALUES ("546-52-4782", "William", "Douglas", "available", 1983, 7, 18, "M
ale", 37111);

MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn, PEfname, PElname, PEstatus,
PEbyear, PEbmonth, PEbday, PEgender, PEsalary) VALUES ("486-42-8416", "Diana", "Lyons", "unavailable", 1997, 12, 21, "Fe
male", 19000);
Query OK, 1 row affected (0.1921 sec)

MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn, PEfname, PElname, PEstatus, PEbyear, PEbmonth, PEbday, PEgender,
 PEsalary) VALUES ("432-57-5789", "Francisca", "Martinez", "in service", 1984, 6, 25, "female", 14);

MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn, PEfname, PElname, PEstatus, PEbyear, PEbmonth, PEbday, PEgender,
 PEsalary) VALUES ("432-67-5786", "Gerard", "Masterpol", "in service", 1987, 5, 23, "male", 18);

MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn, PEfname, PElname, PEstatus, PEbyear, PEbmonth, PEbday, PEgender,
 PEsalary) VALUES ("432-67-5789", "Robert", "Macquary", "in service", 1985, 4, 30, "male", 16);

MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into personnel (PEssn, PEfname, PElname, PEstatus, PEbyear, PEbmonth, PEbday, PEgender,
 PEsalary) VALUES ("433-77-5789", "Richard", "Lopez", "in service", 1983, 6, 4, "male", 17);
```

Figure 16 Records Inserted into Personnel Table

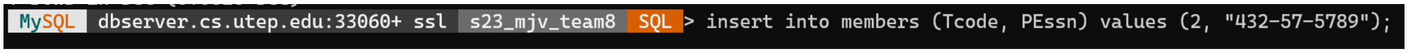


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from personnel;
+-------------+-----------+-----------+-------------+---------+----------+--------+----------+----------+
| PEssn       | PEfname   | PElname   | PEstatus    | PEbyear | PEbmonth | PEbday | PEgender | PEsalary |
+-------------+-----------+-----------+-------------+---------+----------+--------+----------+----------+
| 432-57-5789 | Francisca | Martinez  | in service  |    1984 |        6 |     25 | female   |       14 |
| 432-67-5786 | Gerard    | Masterpol | in service  |    1987 |        5 |     23 | male     |       18 |
| 432-67-5789 | Robert    | Macquary  | in service  |    1985 |        4 |     30 | male     |       16 |
| 433-77-5789 | Richard   | Lopez     | in service  |    1983 |        6 |      4 | male     |       17 |
| 456-66-7890 | Mike      | Smith     | unavailable |    1980 |        4 |     23 | Male     |       21 |
| 486-42-8416 | Diana     | Lyons     | in service  |    1997 |       12 |     21 | Female   |       23 |
| 546-52-4782 | William   | Douglas   | in service  |    1983 |        7 |     18 | Male     |       14 |
| 556-33-5544 | Leo       | Williams  | in service  |    1979 |        5 |     14 | Male     |       18 |
| 718-51-5781 | Aimee     | Feeser    | available   |    1996 |        3 |     30 | Female   |       23 |
| 851-34-7884 | Monica    | Arteaga   | available   |    1965 |        9 |     16 | Female   |       20 |
+-------------+-----------+-----------+-------------+---------+----------+--------+----------+----------+
10 rows in set (0.0751 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

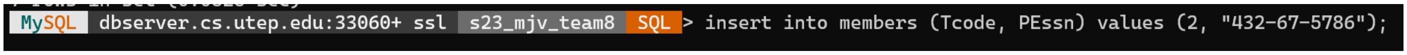
Figure 17 Show Records of Personnel Table

Table: MEMBERS

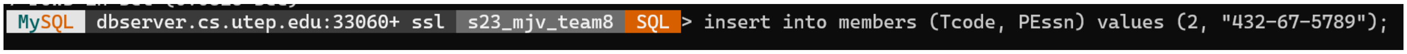
Records in Figure 6 created by Chris and Maria.



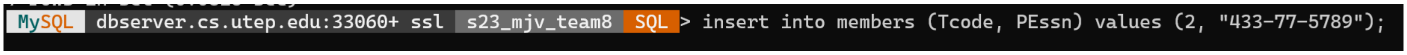
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into members (Tcode, PEssn) values (2, "432-57-5789");
```



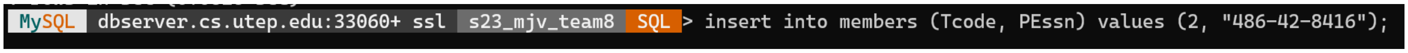
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into members (Tcode, PEssn) values (2, "432-67-5786");
```



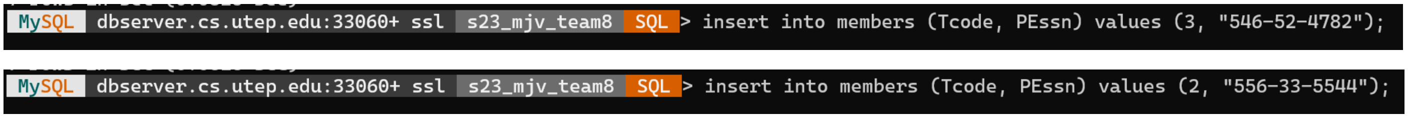
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into members (Tcode, PEssn) values (2, "432-67-5789");
```



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into members (Tcode, PEssn) values (2, "433-77-5789");
```



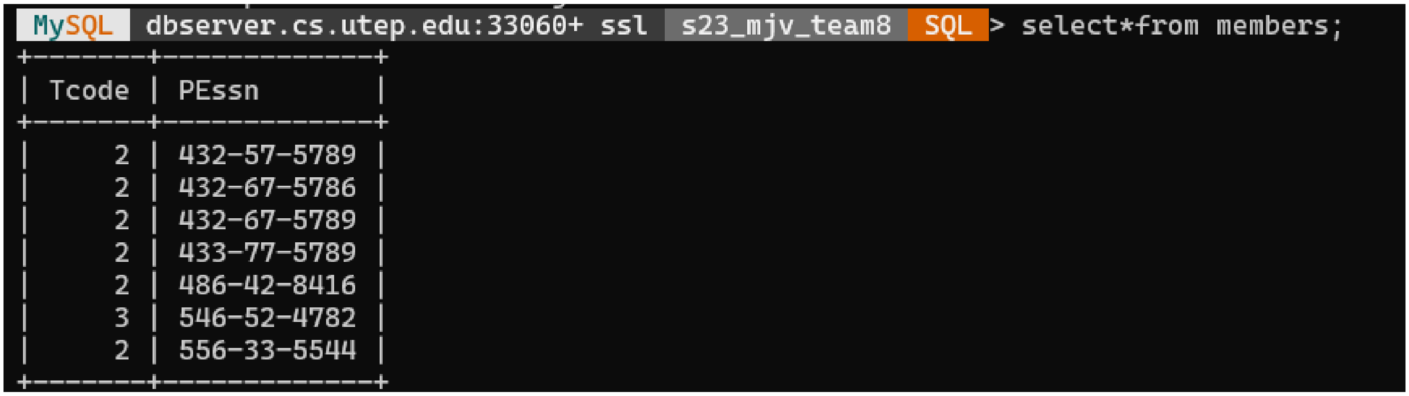
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into members (Tcode, PEssn) values (2, "486-42-8416");
```



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into members (Tcode, PEssn) values (3, "546-52-4782");
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into members (Tcode, PEssn) values (2, "556-33-5544");
```

Figure 18 Records Inserted into Members Table

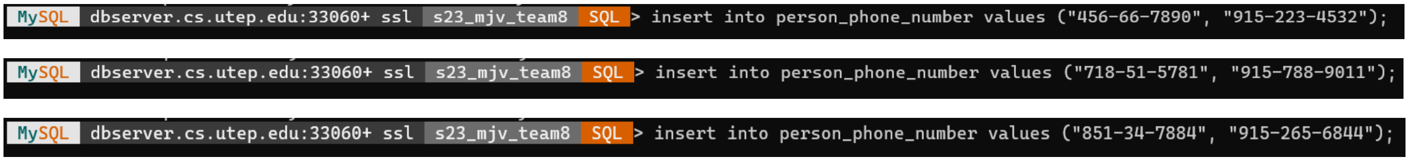


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from members;
+-------+-------------+
| Tcode | PEssn       |
+-------+-------------+
|     2 | 432-57-5789 |
|     2 | 432-67-5786 |
|     2 | 432-67-5789 |
|     2 | 433-77-5789 |
|     2 | 486-42-8416 |
|     3 | 546-52-4782 |
|     2 | 556-33-5544 |
+-------+-------------+
```

Figure 19 Show Records of Members Table

Table: PERSON-PHONE_NUMBER

Records in Figure 7 created by Dario and screenshots by Chris.

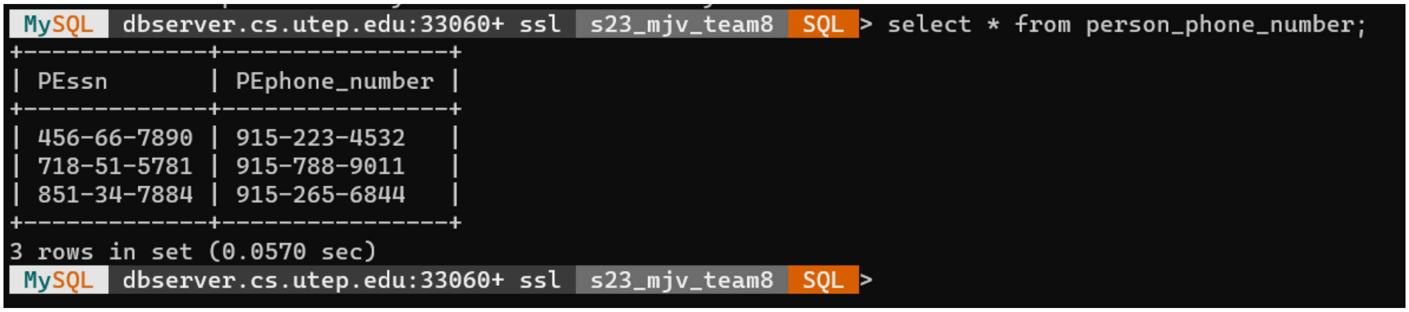


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into person_phone_number values ("456-66-7890", "915-223-4532");
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into person_phone_number values ("718-51-5781", "915-788-9011");
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into person_phone_number values ("851-34-7884", "915-265-6844");
```

Figure 20 Records Inserted into Person_phone_number Table

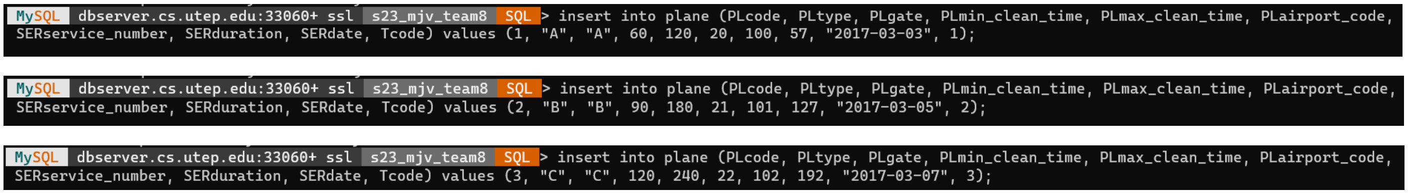


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from person_phone_number;
+-------------+---------------+
| PEssn       | PEphone_number |
+-------------+---------------+
| 456-66-7890 | 915-223-4532   |
| 718-51-5781 | 915-788-9011   |
| 851-34-7884 | 915-265-6844   |
+-------------+---------------+
3 rows in set (0.0570 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 21 Show Records of Person_phone_number Table

Table: PLANE

Records in Figure 8 created by Dario and screenshots by Chris.

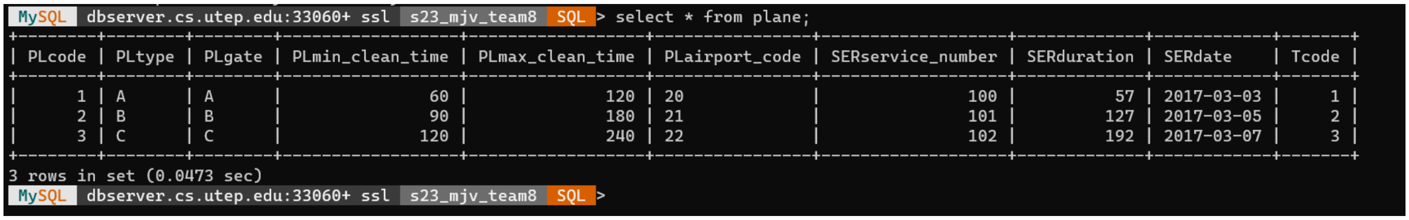


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into plane (PLcode, PLtype, PLgate, PLmin_clean_time, PLmax_clean_time, PLairport_code,
SERservice_number, SERduration, SERdate, Tcode) values (1, "A", "A", 60, 120, 20, 100, 57, "2017-03-03", 1);
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into plane (PLcode, PLtype, PLgate, PLmin_clean_time, PLmax_clean_time, PLairport_code,
SERservice_number, SERduration, SERdate, Tcode) values (2, "B", "B", 90, 180, 21, 101, 127, "2017-03-05", 2);
```

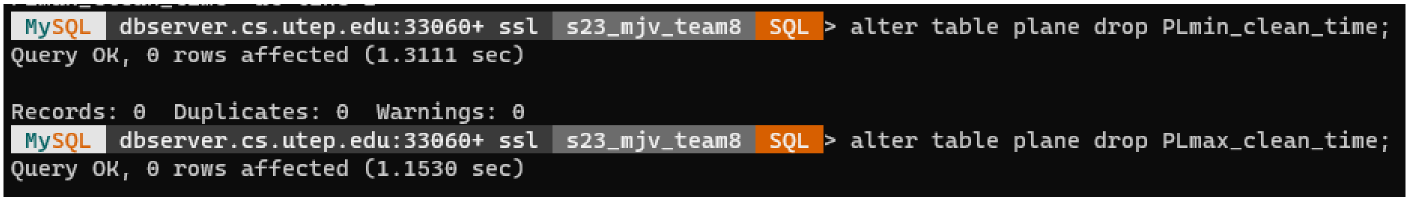
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into plane (PLcode, PLtype, PLgate, PLmin_clean_time, PLmax_clean_time, PLairport_code,
SERservice_number, SERduration, SERdate, Tcode) values (3, "C", "C", 120, 240, 22, 102, 192, "2017-03-07", 3);
```

Figure 22 Records Inserted into Plane Table



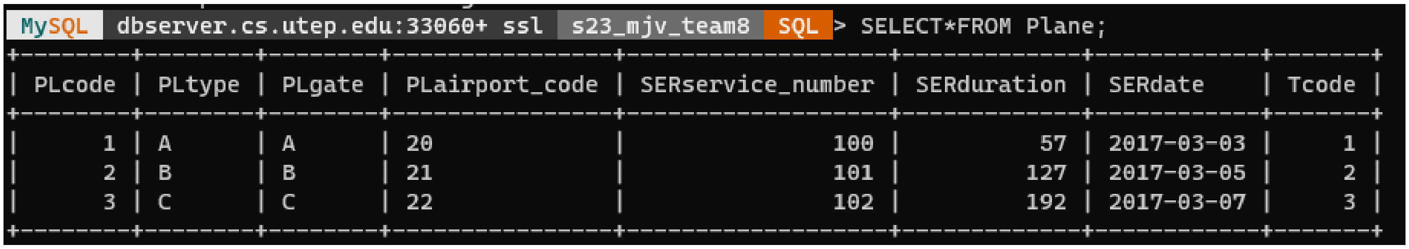
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from plane;
+--------+--------+--------+-----------------+-----------------+----------------+-------------------+-------------+------------+-------+
| PLcode | PLtype | PLgate | PLmin_clean_time | PLmax_clean_time | PLairport_code | SERservice_number | SERduration | SERdate    | Tcode |
+--------+--------+--------+-----------------+-----------------+----------------+-------------------+-------------+------------+-------+
|      1 | A      | A      |              60 |             120 | 20             |               100 |          57 | 2017-03-03 |     1 |
|      2 | B      | B      |              90 |             180 | 21             |               101 |         127 | 2017-03-05 |     2 |
|      3 | C      | C      |             120 |             240 | 22             |               102 |         192 | 2017-03-07 |     3 |
+--------+--------+--------+-----------------+-----------------+----------------+-------------------+-------------+------------+-------+
3 rows in set (0.0473 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 23 Show Recrods of Plane Table



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > alter table plane drop PLmin_clean_time;
Query OK, 0 rows affected (1.3111 sec)

Records: 0  Duplicates: 0  Warnings: 0
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > alter table plane drop PLmax_clean_time;
Query OK, 0 rows affected (1.1530 sec)
```

Figure 24 Normalized Plane Table by Droping Columns PLmni_clean_time and PLmax_clean_time

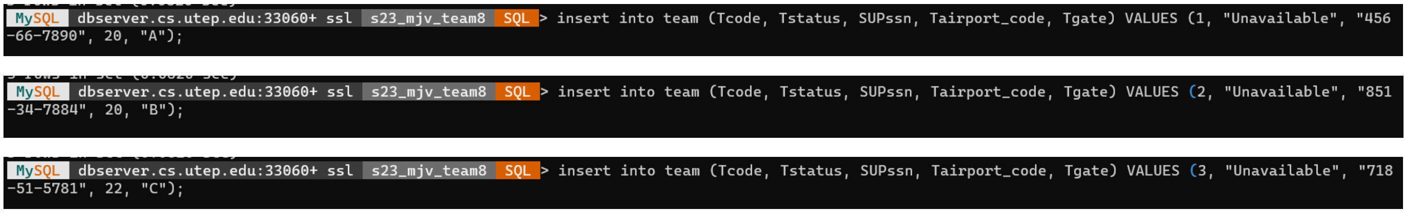


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT*FROM Plane;
+--------+--------+--------+----------------+-------------------+-------------+------------+-------+
| PLcode | PLtype | PLgate | PLairport_code | SERservice_number | SERduration | SERdate    | Tcode |
+--------+--------+--------+----------------+-------------------+-------------+------------+-------+
|      1 | A      | A      | 20             |               100 |          57 | 2017-03-03 |     1 |
|      2 | B      | B      | 21             |               101 |         127 | 2017-03-05 |     2 |
|      3 | C      | C      | 22             |               102 |         192 | 2017-03-07 |     3 |
+--------+--------+--------+----------------+-------------------+-------------+------------+-------+
```

Figure 25 Show Records of Plane Table

Table: TEAM

Records in Figure 26 reated by Dario and screenshots by Chris.

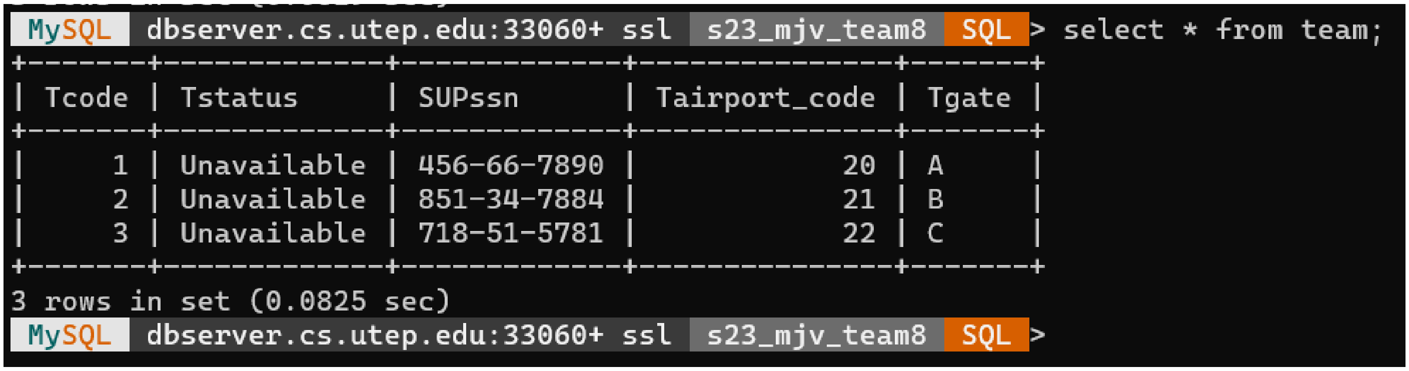


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into team (Tcode, Tstatus, SUPssn, Tairport_code, Tgate) VALUES (1, "Unavailable", "456
-66-7890", 20, "A");
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into team (Tcode, Tstatus, SUPssn, Tairport_code, Tgate) VALUES (2, "Unavailable", "851
-34-7884", 20, "B");
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into team (Tcode, Tstatus, SUPssn, Tairport_code, Tgate) VALUES (3, "Unavailable", "718
-51-5781", 22, "C");
```

Figure 26 Records Inserted into Team Table

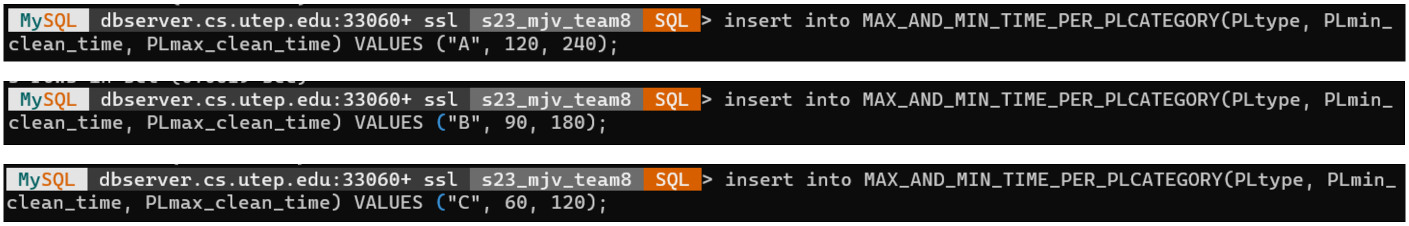


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from team;
+-------+-------------+-------------+---------------+-------+
| Tcode | Tstatus     | SUPssn      | Tairport_code | Tgate |
+-------+-------------+-------------+---------------+-------+
|     1 | Unavailable | 456-66-7890 |            20 | A     |
|     2 | Unavailable | 851-34-7884 |            21 | B     |
|     3 | Unavailable | 718-51-5781 |            22 | C     |
+-------+-------------+-------------+---------------+-------+
3 rows in set (0.0825 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 27 Show Records of Team Table

Table: MAX_AND_MIN_TIME_PER_PLCATEGORY

Records in Figure 10 created by Chris.

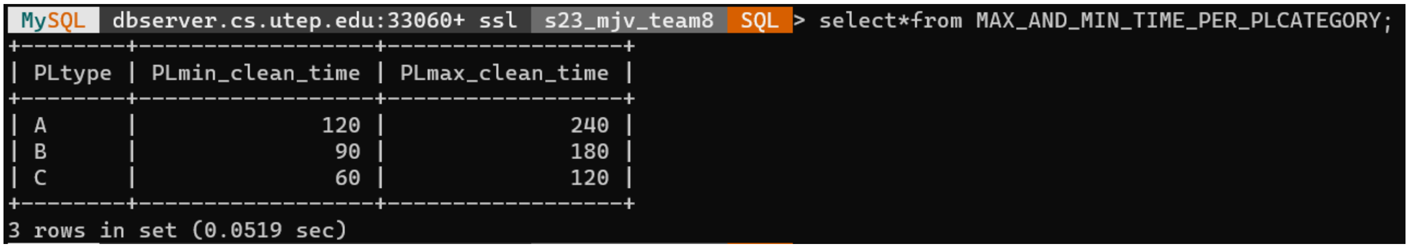


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into MAX_AND_MIN_TIME_PER_PLCATEGORY(PLtype, PLmin_
clean_time, PLmax_clean_time) VALUES ("A", 120, 240);
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into MAX_AND_MIN_TIME_PER_PLCATEGORY(PLtype, PLmin_
clean_time, PLmax_clean_time) VALUES ("B", 90, 180);
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > insert into MAX_AND_MIN_TIME_PER_PLCATEGORY(PLtype, PLmin_
clean_time, PLmax_clean_time) VALUES ("C", 60, 120);
```

Figure 28 Records Inserted into Max_and_min_time_per_plcategory Table



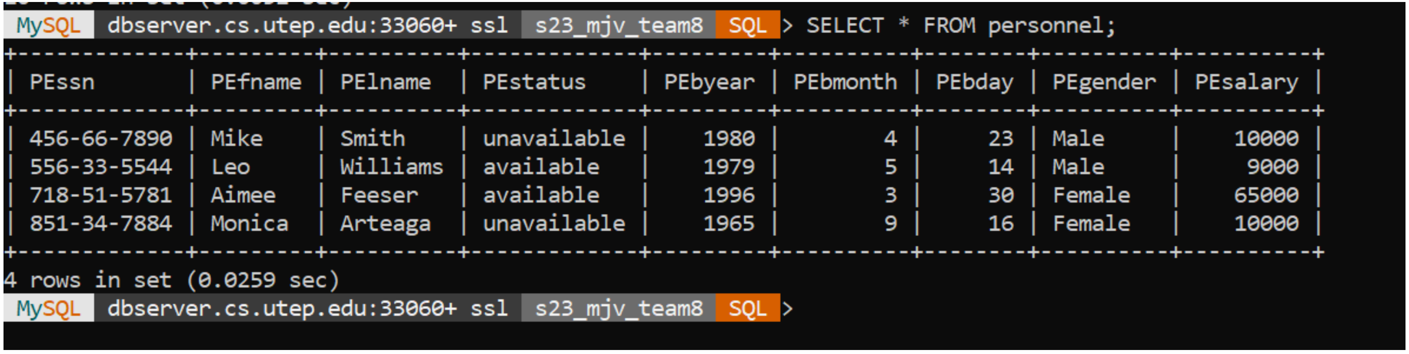
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from MAX_AND_MIN_TIME_PER_PLCATEGORY;
+--------+------------------+------------------+
| PLtype | PLmin_clean_time | PLmax_clean_time |
+--------+------------------+------------------+
| A      |              120 |              240 |
| B      |               90 |              180 |
| C      |               60 |              120 |
+--------+------------------+------------------+
3 rows in set (0.0519 sec)
```

Figure 29 Show Records of Max_and_min_time_per_plcategory Table

8. SQL QUERIES

Vazquez - Design Lead:

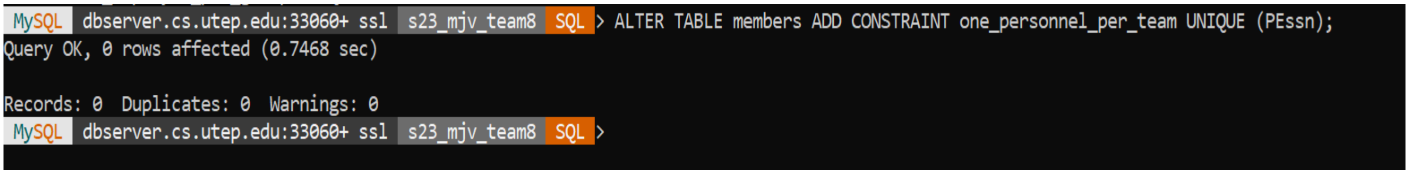
R1. The system should keep track of the personnel information: a unique social security number, name (first name and last name), birth date, gender, phone number, and salary per hour.
Figure 30 shows the Personnel table which keeps track of all the information mentioned in R1.



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT * FROM personnel;
+-------------+--------+----------+-------------+--------+---------+--------+----------+----------+
| PEssn       | PEfname | PElname | PEstatus    | PEbyear | PEbmonth | PEbday | PEgender | PEsalary |
+-------------+--------+----------+-------------+--------+---------+--------+----------+----------+
| 456-66-7890 | Mike   | Smith    | unavailable |   1980 |       4 |     23 | Male     |    10000 |
| 556-33-5544 | Leo    | Williams | available   |   1979 |       5 |     14 | Male     |     9000 |
| 718-51-5781 | Aimee  | Feeser   | available   |   1996 |       3 |     30 | Female   |    65000 |
| 851-34-7884 | Monica | Arteaga  | unavailable |   1965 |       9 |     16 | Female   |    10000 |
+-------------+--------+----------+-------------+--------+---------+--------+----------+----------+
4 rows in set (0.0259 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 30 R1 SQL Query

R2. The system should not allow employees to be part of more than one group at a time.
Figure 31. shows the SQL query that assures that the PEssn will be unique in the members table. Thus, personnel will not be able to be associated with more than one group at a time and R2 is fulfilled.

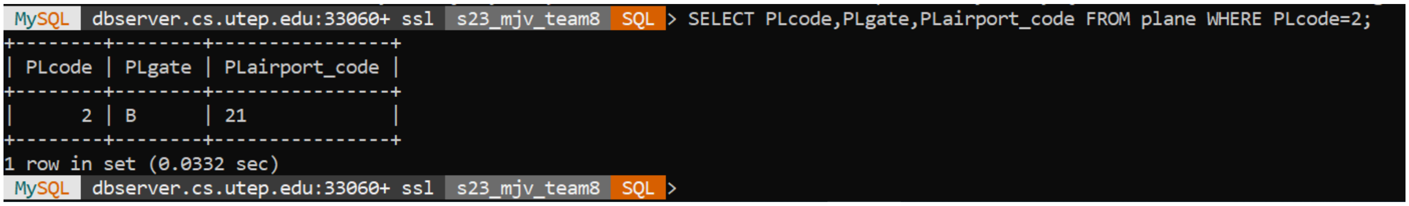


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ALTER TABLE members ADD CONSTRAINT one_personnel_per_team UNIQUE (PEssn);
Query OK, 0 rows affected (0.7468 sec)

Records: 0  Duplicates: 0  Warnings: 0
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 31 R2 SQL Query

R3. The system shall generate a report to keep track of the plane's location via airport code and gate.
Figure 32 shows the gate and airport code for the plane with the ID of '2'. This

ID can be selected by the user to have the location of a specific plane. Thus, R3 is fulfilled.

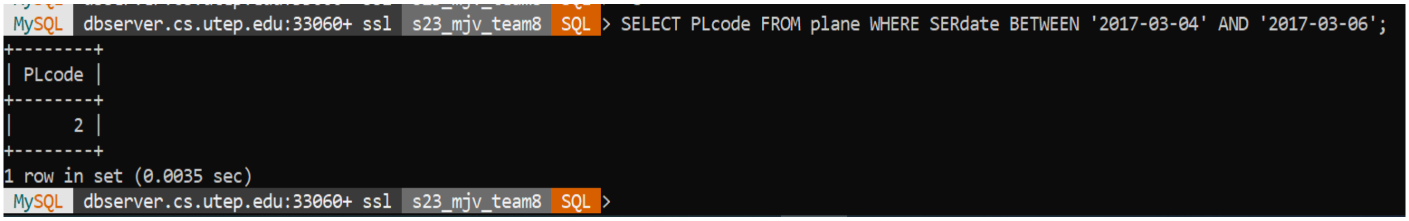


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT PLcode,PLgate,PLairport_code FROM plane WHERE PLcode=2;
+--------+--------+----------------+
| PLcode | PLgate | PLairport_code |
+--------+--------+----------------+
|      2 | B      | 21             |
+--------+--------+----------------+
1 row in set (0.0332 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 32 R3 SQL Query

R4. The system shall generate a report containing the planes that were cleaned for a specific period.

Figure 33 shows the subquery which requests the plane code of planes that were cleaned during a certain date range which fulfills R4.

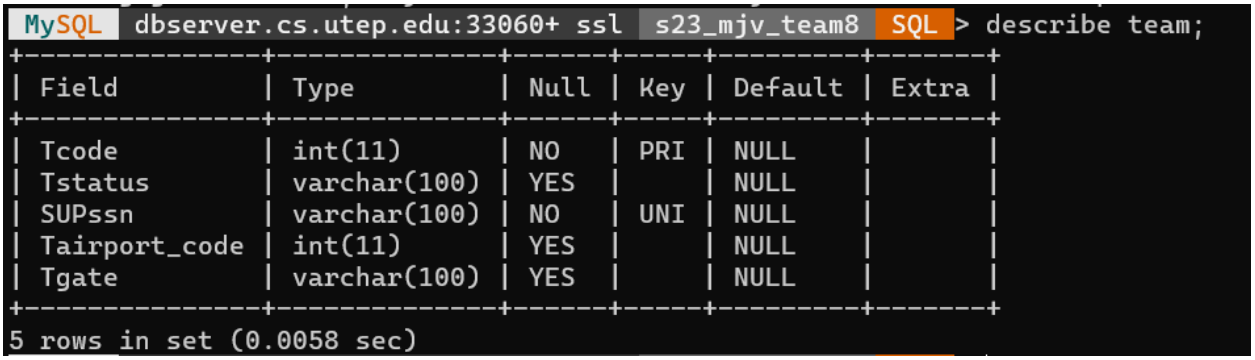


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT PLcode FROM plane WHERE SERdate BETWEEN '2017-03-04' AND '2017-03-06';
+--------+
| PLcode |
+--------+
|      2 |
+--------+
1 row in set (0.0035 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 33 R4 SQL Query

R5. The system should only allow one employee to supervise a group at a given time.

Figure 34 shows the SQL query that assures that the SUPssn will be unique in the team table. Thus, a team will not have more than one supervisor and R5 is fulfilled.

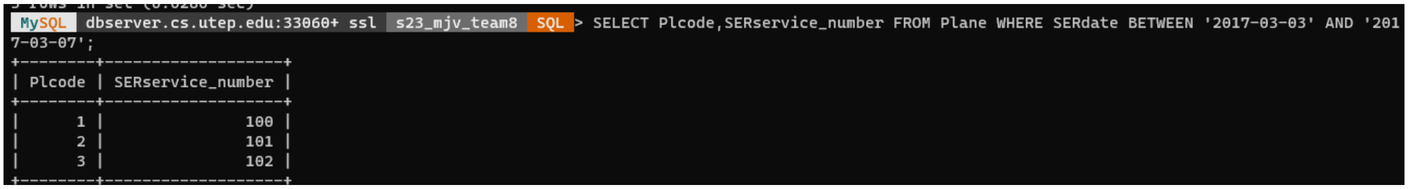


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > describe team;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| Tcode         | int(11)      | NO   | PRI | NULL    |       |
| Tstatus       | varchar(100) | YES  |     | NULL    |       |
| SUPssn        | varchar(100) | NO   | UNI | NULL    |       |
| Tairport_code | int(11)      | YES  |     | NULL    |       |
| Tgate         | varchar(100) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
5 rows in set (0.0058 sec)
```

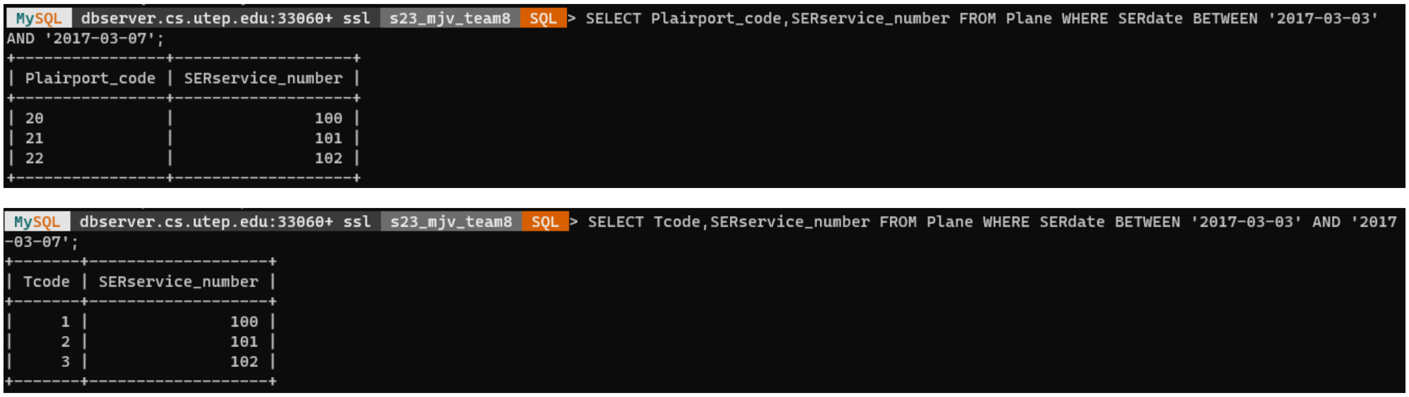
Figure 34 R5 SQL Query

Vargas – SQL Lead:

Figure 35 shows the SQL Query that satisfies R6 (i.e. The system should allow the general manager to generate a list of the services provided per plane, airport, group for a specified period.)



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT Plcode,SERservice_number FROM Plane WHERE SERdate BETWEEN '2017-03-03' AND '201
7-03-07';
+--------+-------------------+
| Plcode | SERservice_number |
+--------+-------------------+
|      1 |               100 |
|      2 |               101 |
|      3 |               102 |
+--------+-------------------+
```

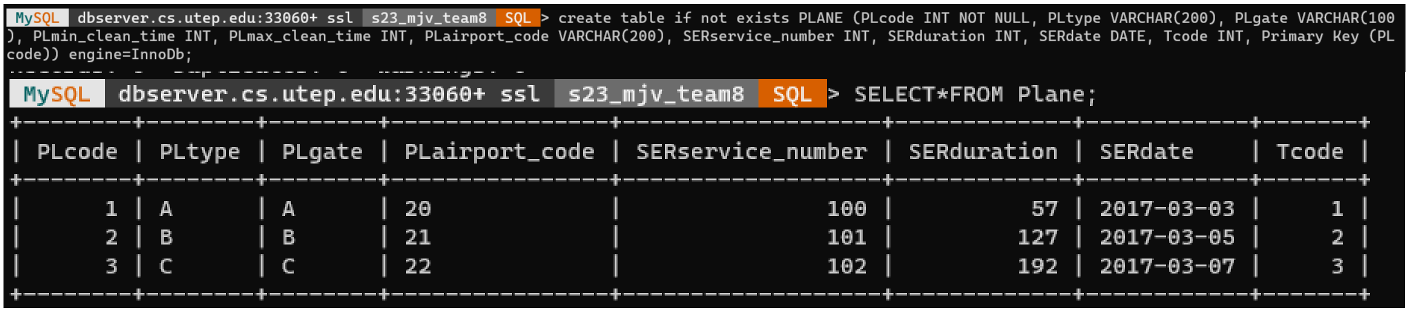


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT Plairport_code,SERservice_number FROM Plane WHERE SERdate BETWEEN '2017-03-03' AND '2017-03-07';
+----------------+-------------------+
| Plairport_code | SERservice_number |
+----------------+-------------------+
| 20             |               100 |
| 21             |               101 |
| 22             |               102 |
+----------------+-------------------+
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT Tcode,SERservice_number FROM Plane WHERE SERdate BETWEEN '2017-03-03' AND '2017-03-07';
+-------+-------------------+
| Tcode | SERservice_number |
+-------+-------------------+
|     1 |               100 |
|     2 |               101 |
|     3 |               102 |
+-------+-------------------+
```

Figure 35 R6 SQL Query. SQL Query that retrieves services provided per plane, airport, group in a specified period.

Figure 36 shows the SQL Query that satisfies R7(i.e. The system should keep track of the services: service number, plane code, airport code, date, time it took to fulfill the service, and group code.)

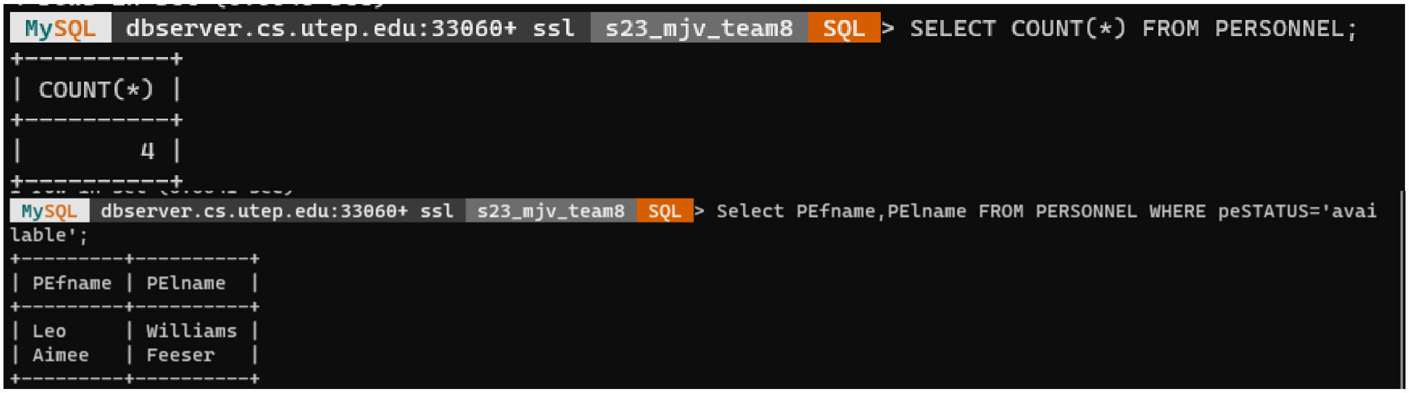


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create table if not exists PLANE (PLcode INT NOT NULL, PLtype VARCHAR(200), PLgate VARCHAR(100), PLmin_clean_time INT, PLmax_clean_time INT, PLairport_code VARCHAR(200), SERservice_number INT, SERduration INT, SERdate DATE, Tcode INT, Primary Key (PLcode)) engine=InnoDb;
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT*FROM Plane;
+--------+--------+--------+----------------+-------------------+-------------+------------+-------+
| PLcode | PLtype | PLgate | PLairport_code | SERservice_number | SERduration | SERdate    | Tcode |
+--------+--------+--------+----------------+-------------------+-------------+------------+-------+
|      1 | A      | A      | 20             |               100 |          57 | 2017-03-03 |     1 |
|      2 | B      | B      | 21             |               101 |         127 | 2017-03-05 |     2 |
|      3 | C      | C      | 22             |               102 |         192 | 2017-03-07 |     3 |
+--------+--------+--------+----------------+-------------------+-------------+------------+-------+
```

Figure 36 R7 SQL Query. SQL Query that keeps track of the services provided: service number, plane code, airport code, date, time it took to fulfill the service, and group code.

Figure 37 shows the SQL Query that satisfies R8 (i.e. The system should allow the general manager to generate information to know the total number of employees and the employees that are available at a specific moment.)

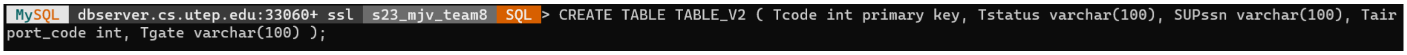


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT COUNT(*) FROM PERSONNEL;
+----------+
| COUNT(*) |
+----------+
|        4 |
+----------+
```

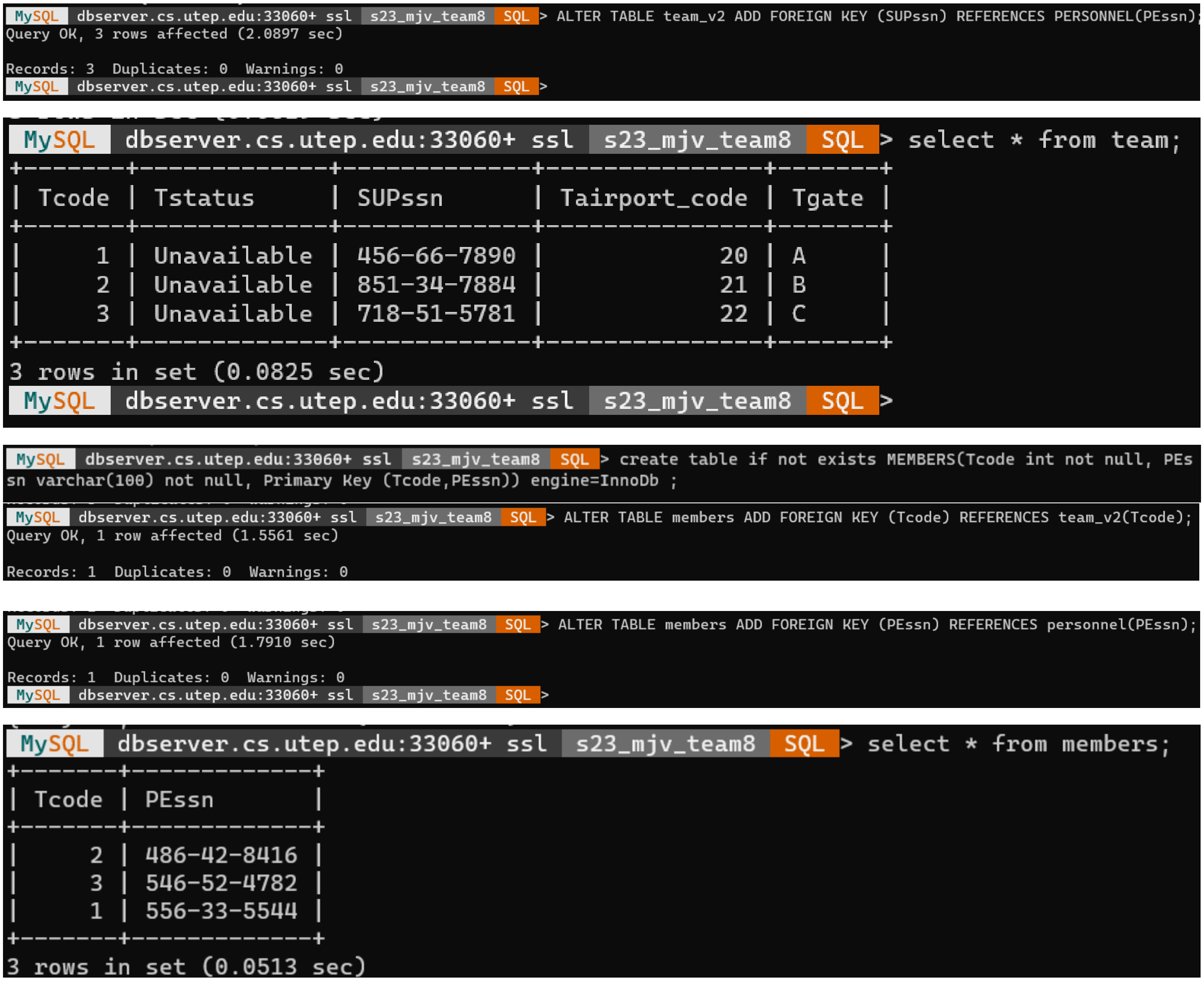
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > Select PEfname,PElname FROM PERSONNEL WHERE peSTATUS='available';
+---------+----------+
| PEfname | PElname  |
+---------+----------+
| Leo     | Williams |
| Aimee   | Feeser   |
+---------+----------+
```

Figure 37 R8 SQL Query. SQL Query that retrieves total number of employees and employees with an available status.

Figure 38 shows the SQL Query that satisfies R9. (The system should keep track of each group, its status, supervisor, location and code.)



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > CREATE TABLE TABLE_V2 ( Tcode int primary key, Tstatus varchar(100), SUPssn varchar(100), Tairport_code int, Tgate varchar(100) );
```



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ALTER TABLE team_v2 ADD FOREIGN KEY (SUPssn) REFERENCES PERSONNEL(PEssn);
Query OK, 3 rows affected (2.0897 sec)

Records: 3  Duplicates: 0  Warnings: 0
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from team;
+-------+-------------+-------------+---------------+-------+
| Tcode | Tstatus     | SUPssn      | Tairport_code | Tgate |
+-------+-------------+-------------+---------------+-------+
|     1 | Unavailable | 456-66-7890 |            20 | A     |
|     2 | Unavailable | 851-34-7884 |            21 | B     |
|     3 | Unavailable | 718-51-5781 |            22 | C     |
+-------+-------------+-------------+---------------+-------+
3 rows in set (0.0825 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create table if not exists MEMBERS(Tcode int not null, PEs
sn varchar(100) not null, Primary Key (Tcode,PEssn)) engine=InnoDb ;
```

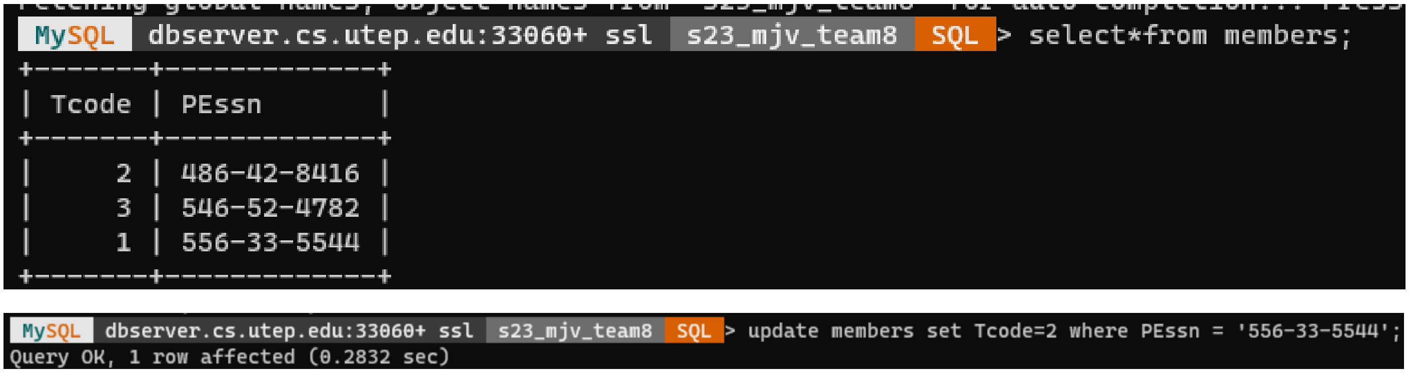
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ALTER TABLE members ADD FOREIGN KEY (Tcode) REFERENCES team_v2(Tcode);
Query OK, 1 row affected (1.5561 sec)

Records: 1  Duplicates: 0  Warnings: 0
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ALTER TABLE members ADD FOREIGN KEY (PEssn) REFERENCES personnel(PEssn);
Query OK, 1 row affected (1.7910 sec)

Records: 1  Duplicates: 0  Warnings: 0
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from members;
+-------+-------------+
| Tcode | PEssn       |
+-------+-------------+
|     2 | 486-42-8416 |
|     3 | 546-52-4782 |
|     1 | 556-33-5544 |
+-------+-------------+
3 rows in set (0.0513 sec)
```

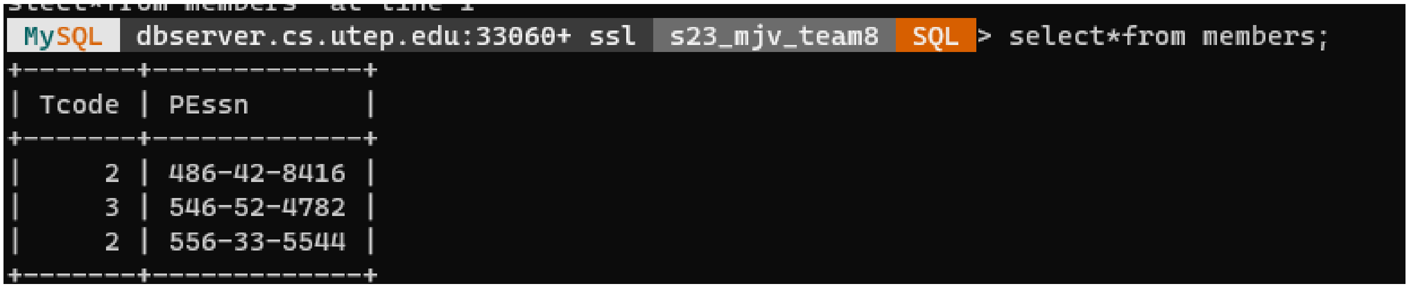
Figure 38 R9 SQL Query. SQL Queries that generate the tables that keep track of each group, its status, supervisor, location and code.)

Figure 39 shows the SQL Query that satisfies R10. (The system should allow us to update the group membership of the worker.



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from members;
+-------+-------------+
| Tcode | PEssn       |
+-------+-------------+
|     2 | 486-42-8416 |
|     3 | 546-52-4782 |
|     1 | 556-33-5544 |
+-------+-------------+
```

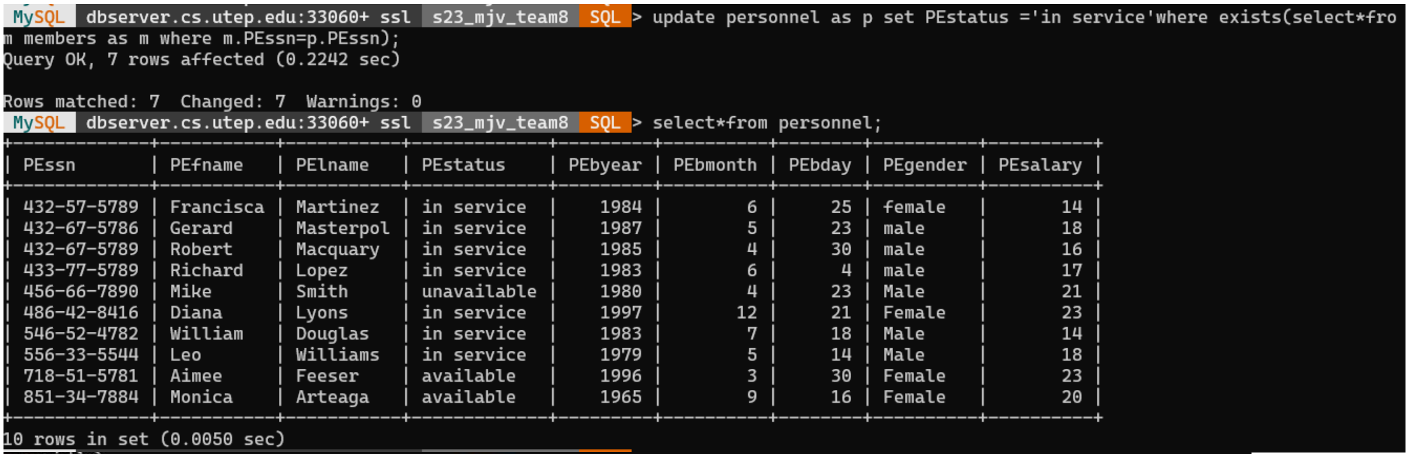
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > update members set Tcode=2 where PEssn = '556-33-5544';
Query OK, 1 row affected (0.2832 sec)
```



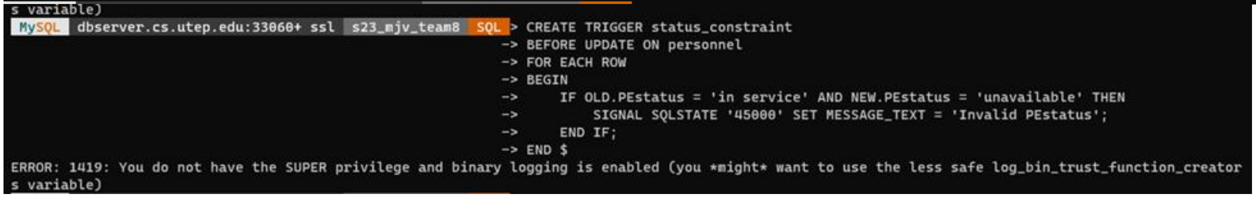
```
select*from members' at line 1
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from members;
+-------+--------------+
| Tcode | PEssn        |
+-------+--------------+
|     2 | 486-42-8416 |
|     3 | 546-52-4782 |
|     2 | 556-33-5544 |
+-------+--------------+
```

Figure 39 R10 SQL Query.

Figure 40 shows the SQL Query that satisfies R11. (When assigned to a group the status of a worker is changed to in service and the status of an in service worker cannot be change to unavailable.)



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > update personnel as p set PEstatus ='in service'where exists(select*fro
m members as m where m.PEssn=p.PEssn);
Query OK, 7 rows affected (0.2242 sec)

Rows matched: 7  Changed: 7  Warnings: 0
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from personnel;
+-------------+----------+----------+-------------+---------+----------+--------+----------+----------+
| PEssn       | PEfname  | PElname  | PEstatus    | PEbyear | PEbmonth | PEbday | PEgender | PEsalary |
+-------------+----------+----------+-------------+---------+----------+--------+----------+----------+
| 432-57-5789 | Francisca | Martinez | in service |    1984 |        6 |     25 | female   |       14 |
| 432-67-5786 | Gerard   | Masterpol | in service |    1987 |        5 |     23 | male     |       18 |
| 432-67-5789 | Robert   | Macquary | in service  |    1985 |        4 |     30 | male     |       16 |
| 433-77-5789 | Richard  | Lopez    | in service  |    1983 |        6 |      4 | male     |       17 |
| 456-66-7890 | Mike     | Smith    | unavailable |    1980 |        4 |     23 | Male     |       21 |
| 486-42-8416 | Diana    | Lyons    | in service  |    1997 |       12 |     21 | Female   |       23 |
| 546-52-4782 | William  | Douglas  | in service  |    1983 |        7 |     18 | Male     |       14 |
| 556-33-5544 | Leo      | Williams | in service  |    1979 |        5 |     14 | Male     |       18 |
| 718-51-5781 | Aimee    | Feeser   | available   |    1996 |        3 |     30 | Female   |       23 |
| 851-34-7884 | Monica   | Arteaga  | available   |    1965 |        9 |     16 | Female   |       20 |
+-------------+----------+----------+-------------+---------+----------+--------+----------+----------+
10 rows in set (0.0050 sec)
```

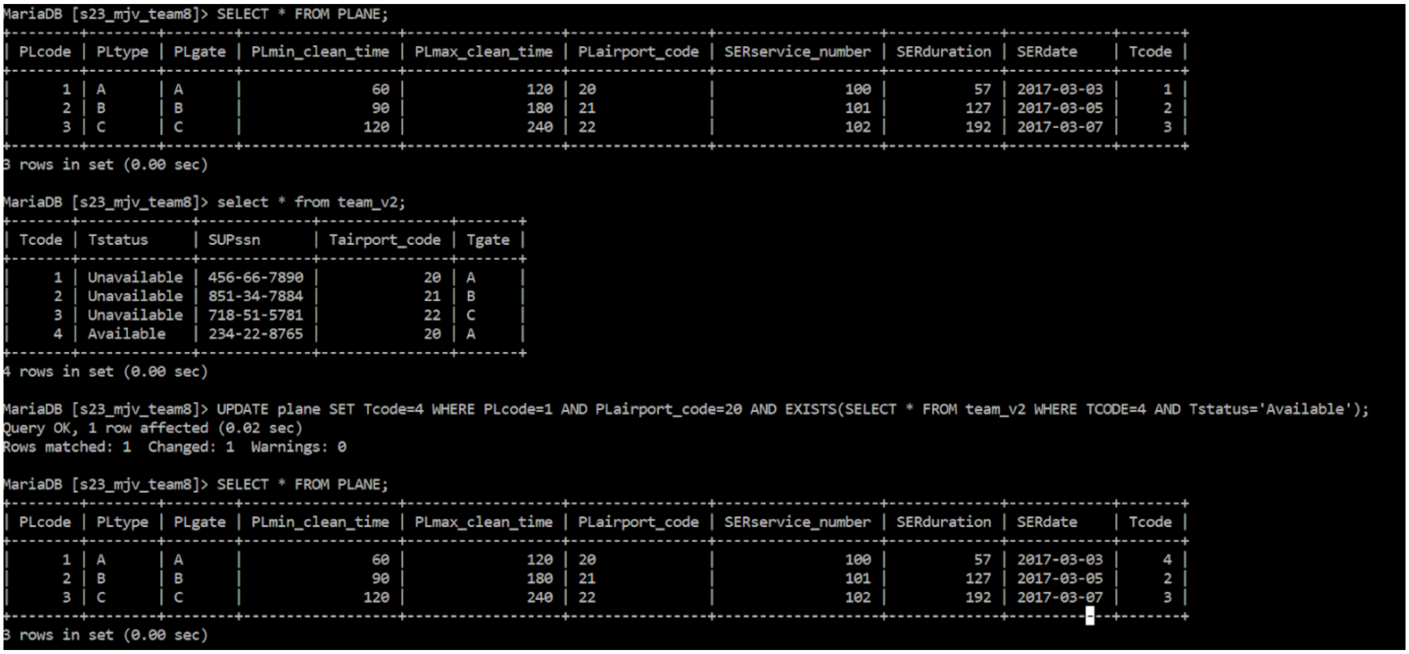


```
s variable)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > CREATE TRIGGER status_constraint
                                                          -> BEFORE UPDATE ON personnel
                                                          -> FOR EACH ROW
                                                          -> BEGIN
                                                          ->     IF OLD.PEstatus = 'in service' AND NEW.PEstatus = 'unavailable' THEN
                                                          ->         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid PEstatus';
                                                          ->     END IF;
                                                          -> END $
ERROR: 1419: You do not have the SUPER privilege and binary logging is enabled (you *might* want to use the less safe log_bin_trust_function_creator
s variable)
```

Figure 40 R11 SQL Query. SQL Query that changes the status of a worker to 'in service' when the worker is assigned to a group and the status of an in service worker cannot be change to unavailable.

Taposh Sarker - Interface Lead:

R12. The system will allow update of which airport and plane is assigned to each group.
That means a particular plane and airport can be assigned to a different group. We can use UPDATE query to do that.
Example Query:
UPDATE plane SET Tcode=4 WHERE PLcode=1 AND PLairport_code=20 AND EXISTS(SELECT * FROM team_v2 WHERE TCODE=4 AND Tstatus='Available');

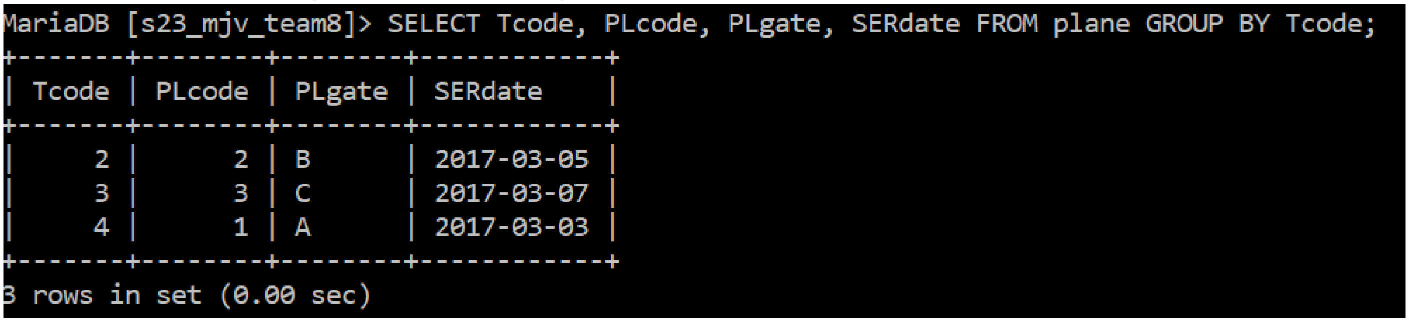


```
MariaDB [s23_mjv_team8]> SELECT * FROM PLANE;
+--------+--------+--------+------------------+------------------+----------------+-------------------+-------------+------------+-------+
| PLcode | PLtype | PLgate | PLmin_clean_time | PLmax_clean_time | PLairport_code | SERservice_number | SERduration | SERdate    | Tcode |
+--------+--------+--------+------------------+------------------+----------------+-------------------+-------------+------------+-------+
|      1 | A      | A      |               60 |              120 | 20             |               100 |          57 | 2017-03-03 |     1 |
|      2 | B      | B      |               90 |              180 | 21             |               101 |         127 | 2017-03-05 |     2 |
|      3 | C      | C      |              120 |              240 | 22             |               102 |         192 | 2017-03-07 |     3 |
+--------+--------+--------+------------------+------------------+----------------+-------------------+-------------+------------+-------+
3 rows in set (0.00 sec)

MariaDB [s23_mjv_team8]> select * from team_v2;
+-------+-------------+-------------+---------------+-------+
| Tcode | Tstatus     | SUPssn      | Tairport_code | Tgate |
+-------+-------------+-------------+---------------+-------+
|     1 | Unavailable | 456-66-7890 |            20 | A     |
|     2 | Unavailable | 851-34-7884 |            21 | B     |
|     3 | Unavailable | 718-51-5781 |            22 | C     |
|     4 | Available   | 234-22-8765 |            20 | A     |
+-------+-------------+-------------+---------------+-------+
4 rows in set (0.00 sec)

MariaDB [s23_mjv_team8]> UPDATE plane SET Tcode=4 WHERE PLcode=1 AND PLairport_code=20 AND EXISTS(SELECT * FROM team_v2 WHERE TCODE=4 AND Tstatus='Available');
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [s23_mjv_team8]> SELECT * FROM PLANE;
+--------+--------+--------+------------------+------------------+----------------+-------------------+-------------+------------+-------+
| PLcode | PLtype | PLgate | PLmin_clean_time | PLmax_clean_time | PLairport_code | SERservice_number | SERduration | SERdate    | Tcode |
+--------+--------+--------+------------------+------------------+----------------+-------------------+-------------+------------+-------+
|      1 | A      | A      |               60 |              120 | 20             |               100 |          57 | 2017-03-03 |     4 |
|      2 | B      | B      |               90 |              180 | 21             |               101 |         127 | 2017-03-05 |     2 |
|      3 | C      | C      |              120 |              240 | 22             |               102 |         192 | 2017-03-07 |     3 |
+--------+--------+--------+------------------+------------------+----------------+-------------------+-------------+------------+-------+
3 rows in set (0.00 sec)
```

Figure 41 R12 SQL Query.

R13. The system should generate a report of which group cleaned which plane at which airport at which specific time.

SELECT Tcode, PLcode, PLgate, SERdate FROM plane GROUP BY Tcode;

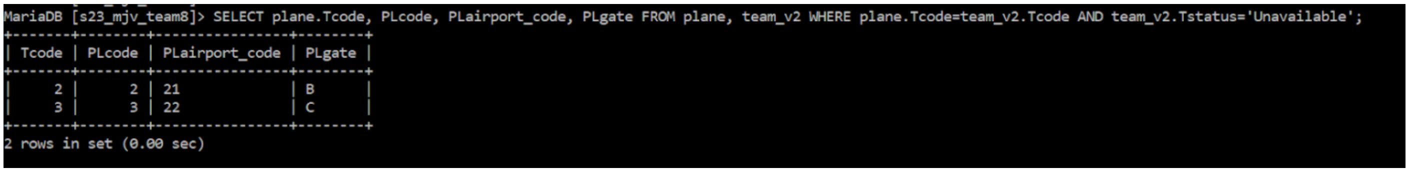


```
MariaDB [s23_mjv_team8]> SELECT Tcode, PLcode, PLgate, SERdate FROM plane GROUP BY Tcode;
+-------+--------+--------+------------+
| Tcode | PLcode | PLgate | SERdate    |
+-------+--------+--------+------------+
|     2 |      2 | B      | 2017-03-05 |
|     3 |      3 | C      | 2017-03-07 |
|     4 |      1 | A      | 2017-03-03 |
+-------+--------+--------+------------+
3 rows in set (0.00 sec)
```

Figure 42 R13 SQL Query.

R14. The system shall keep records of the exact location (airport, plane, gate) of a group while in-service.

```
SELECT P.PLcode, P.PLgate, P.PLairport_code, P.SERdate
            FROM plane AS P
            INNER JOIN team AS T
            ON P.PLcode = T.Tcode
            WHERE T.Tstatus = 'in service' AND T.Tcode = 2;
```



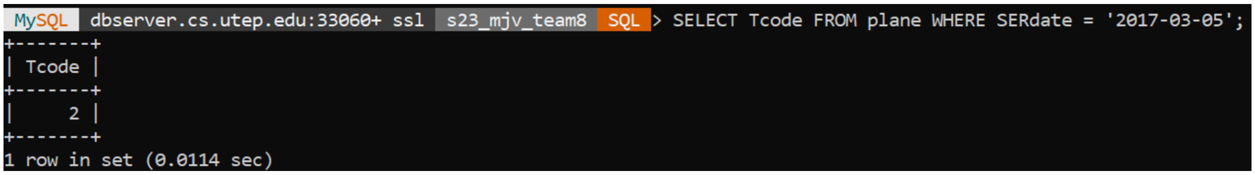
```
MariaDB [s23_mjv_team8]> SELECT plane.Tcode, PLcode, PLairport_code, PLgate FROM plane, team_v2 WHERE plane.Tcode=team_v2.Tcode AND team_v2.Tstatus='Unavailable';
+-------+--------+----------------+--------+
| Tcode | PLcode | PLairport_code | PLgate |
+-------+--------+----------------+--------+
|     2 |      2 | 21             | B      |
|     3 |      3 | 22             | C      |
+-------+--------+----------------+--------+
2 rows in set (0.00 sec)
```

Figure 43 R14 SQL Query.

R15. The system shall keep track of which groups are in-service at a particular point of time.

Let’s say we want to query for date: 2017-03-05

SELECT Tcode FROM plane WHERE SERdate= ‘2017-03-05';



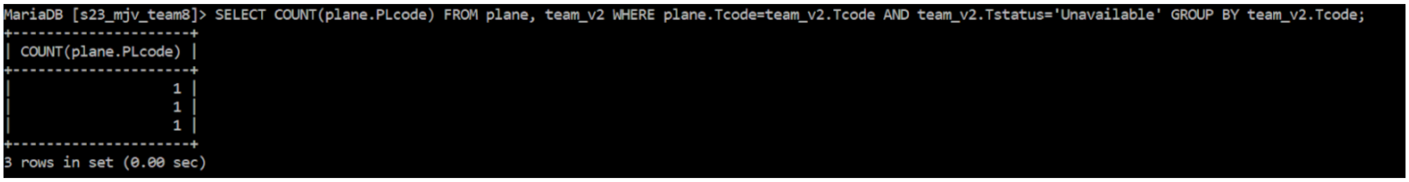
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT Tcode FROM plane WHERE SERdate = '2017-03-05';
+-------+
| Tcode |
+-------+
|     2 |
+-------+
1 row in set (0.0114 sec)
```

Figure 44 R15 SQL Query.

R16. The system will make sure that a group is assigned to only one plane at a time.

We can see how many planes are being serviced currently by each team:

SELECT COUNT(plane.PLcode) FROM plane, team_v2 WHERE plane.Tcode=team_v2.Tcode AND team_v2.Tstatus='Unavailable' GROUP BY team_v2.Tcode;



```
MariaDB [s23_mjv_team8]> SELECT COUNT(plane.PLcode) FROM plane, team_v2 WHERE plane.Tcode=team_v2.Tcode AND team_v2.Tstatus='Unavailable' GROUP BY team_v2.Tcode;
+---------------------+
| COUNT(plane.PLcode) |
+---------------------+
|                   1 |
|                   1 |
|                   1 |
+---------------------+
3 rows in set (0.00 sec)
```

Figure 45 R16 SQL Query. But to ensure the value does not go above 1, we have to put constraints for the attribute in the schema.

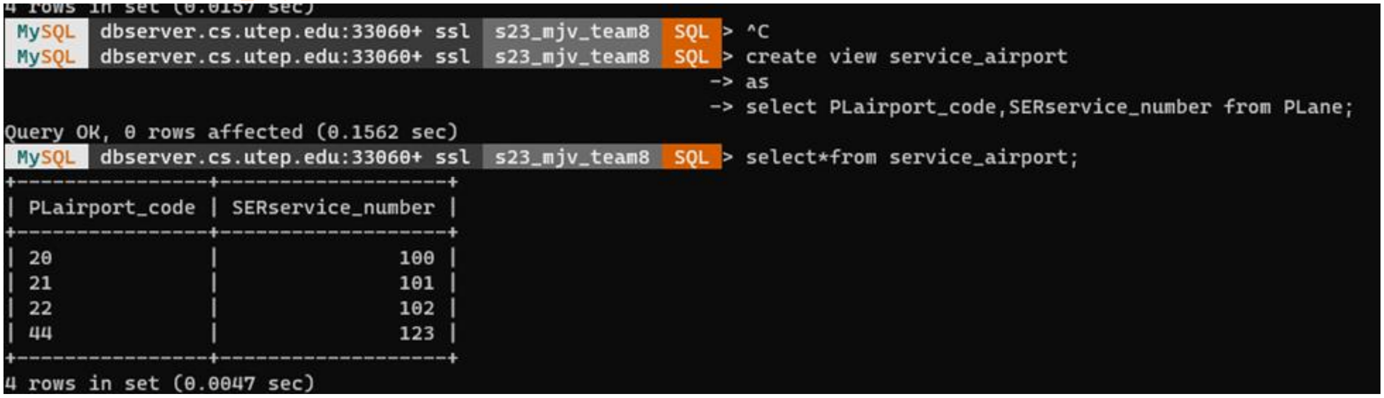
R17. Generate a report that displays all the services per airport.

First we create a view:

CREATE VIEW service_airport as SELECT PLairport_code, SERservice_number from Plane;

Then we query the view.

SELECT * from service_airport;

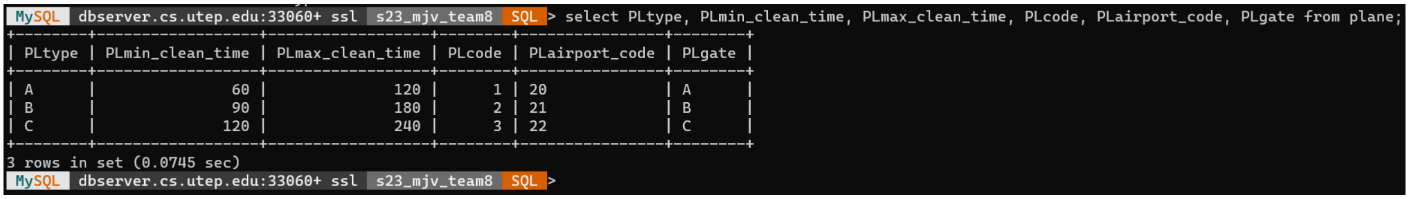


```
4 rows in set (0.0157 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ^C
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create view service_airport
                                                          -> as
                                                          -> select PLairport_code,SERservice_number from PLane;
Query OK, 0 rows affected (0.1562 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from service_airport;
+----------------+-------------------+
| PLairport_code | SERservice_number |
+----------------+-------------------+
| 20             |               100 |
| 21             |               101 |
| 22             |               102 |
| 44             |               123 |
+----------------+-------------------+
4 rows in set (0.0047 sec)
```

Figure 46 R17 SQL Query.

Ibarra - Report Lead:

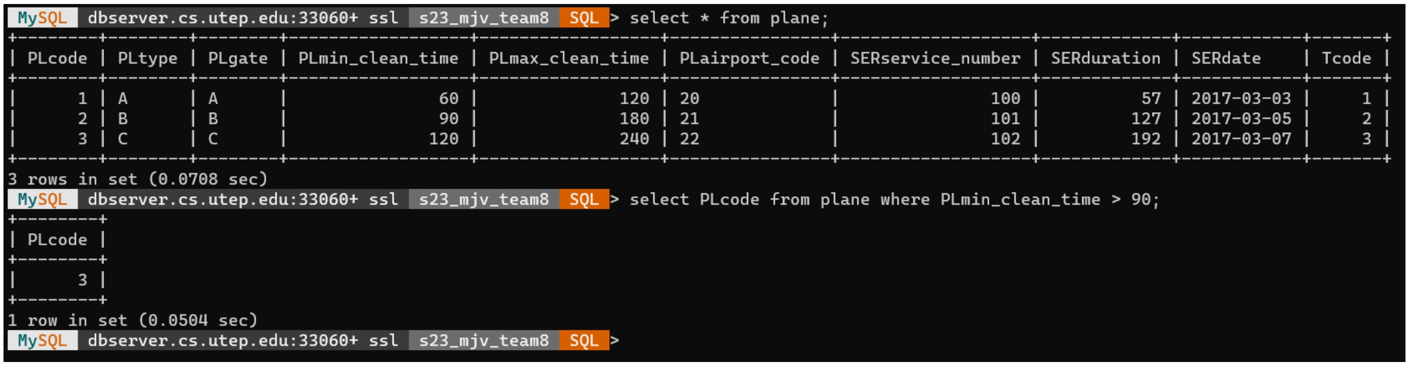
R18. The system will keep track of the planes' information: category (A, B, C), expected time for it to be cleaned (min and max time), a unique code and compose location (airport code, gate).



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select PLtype, PLmin_clean_time, PLmax_clean_time, PLcode, PLairport_code, PLgate from plane;
+--------+------------------+------------------+--------+----------------+--------+
| PLtype | PLmin_clean_time | PLmax_clean_time | PLcode | PLairport_code | PLgate |
+--------+------------------+------------------+--------+----------------+--------+
| A      |               60 |              120 |      1 | 20             | A      |
| B      |               90 |              180 |      2 | 21             | B      |
| C      |              120 |              240 |      3 | 22             | C      |
+--------+------------------+------------------+--------+----------------+--------+
3 rows in set (0.0745 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 47 R18 SQL Query.

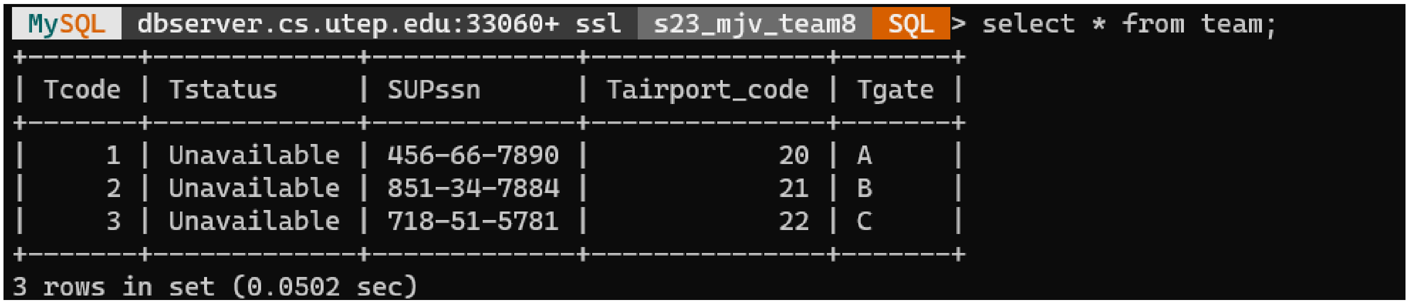
R19. The system will generate a report of the groups that took longer than expected to clean a plane. (assuming avg cleaning time = 90 min.)



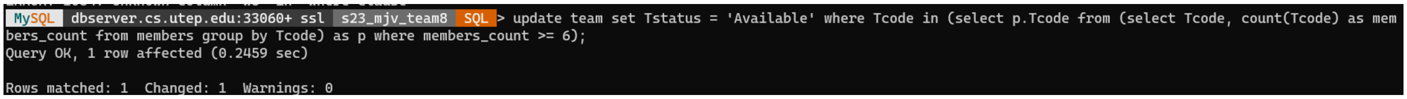
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from plane;
+--------+--------+--------+------------------+------------------+----------------+--------------------+-------------+------------+-------+
| PLcode | PLtype | PLgate | PLmin_clean_time | PLmax_clean_time | PLairport_code | SERservice_number | SERduration | SERdate    | Tcode |
+--------+--------+--------+------------------+------------------+----------------+--------------------+-------------+------------+-------+
|      1 | A      | A      |               60 |              120 | 20             |                100 |          57 | 2017-03-03 |     1 |
|      2 | B      | B      |               90 |              180 | 21             |                101 |         127 | 2017-03-05 |     2 |
|      3 | C      | C      |              120 |              240 | 22             |                102 |         192 | 2017-03-07 |     3 |
+--------+--------+--------+------------------+------------------+----------------+--------------------+-------------+------------+-------+
3 rows in set (0.0708 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select PLcode from plane where PLmin_clean_time > 90;
+--------+
| PLcode |
+--------+
|      3 |
+--------+
1 row in set (0.0504 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 48 R19 SQL Query.

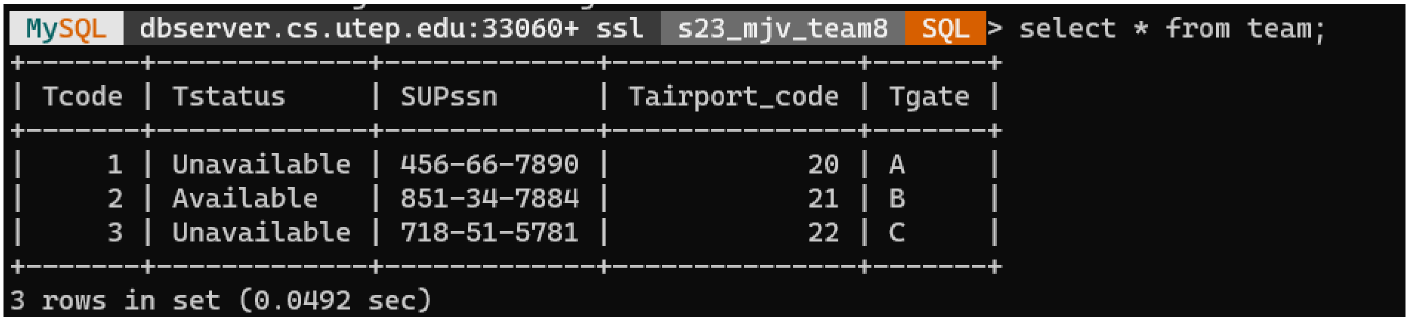
R20. "The system will make sure that each group has 7 members for it to be considered operational." For this, we count the number of members with the same team code (Tcode) in the members table and determine the count for each team code. Afterwards, the team code counts with 6 members or more will be considered operational (since a team has to have a supervisor and thus that increases the total Tcode count by 1).



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from team;
+-------+-------------+-------------+----------------+-------+
| Tcode | Tstatus     | SUPssn      | Tairport_code | Tgate |
+-------+-------------+-------------+----------------+-------+
|     1 | Unavailable | 456-66-7890 |             20 | A     |
|     2 | Unavailable | 851-34-7884 |             21 | B     |
|     3 | Unavailable | 718-51-5781 |             22 | C     |
+-------+-------------+-------------+----------------+-------+
3 rows in set (0.0502 sec)
```



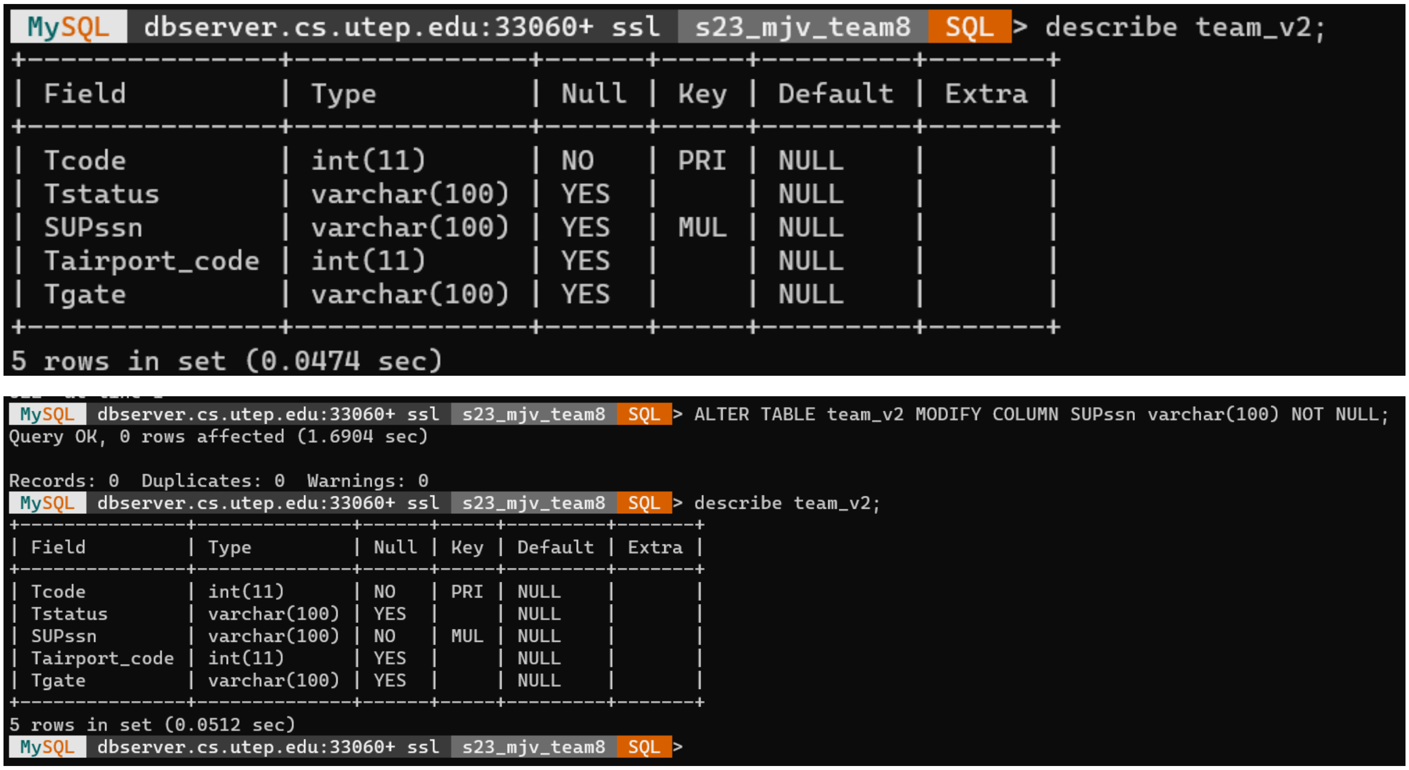
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > update team set Tstatus = 'Available' where Tcode in (select p.Tcode from (select Tcode, count(Tcode) as mem
bers_count from members group by Tcode) as p where members_count >= 6);
Query OK, 1 row affected (0.2459 sec)

Rows matched: 1  Changed: 1  Warnings: 0
```



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from team;
+-------+-------------+-------------+---------------+-------+
| Tcode | Tstatus     | SUPssn      | Tairport_code | Tgate |
+-------+-------------+-------------+---------------+-------+
|     1 | Unavailable | 456-66-7890 |            20 | A     |
|     2 | Available   | 851-34-7884 |            21 | B     |
|     3 | Unavailable | 718-51-5781 |            22 | C     |
+-------+-------------+-------------+---------------+-------+
3 rows in set (0.0492 sec)
```

Figure 49 R20 SQL Query.

R21. “The system will make sure that each group has a supervisor.” For this we look at table “team” or “team_v2” and make sure that the SUPssn cannot be a null value.

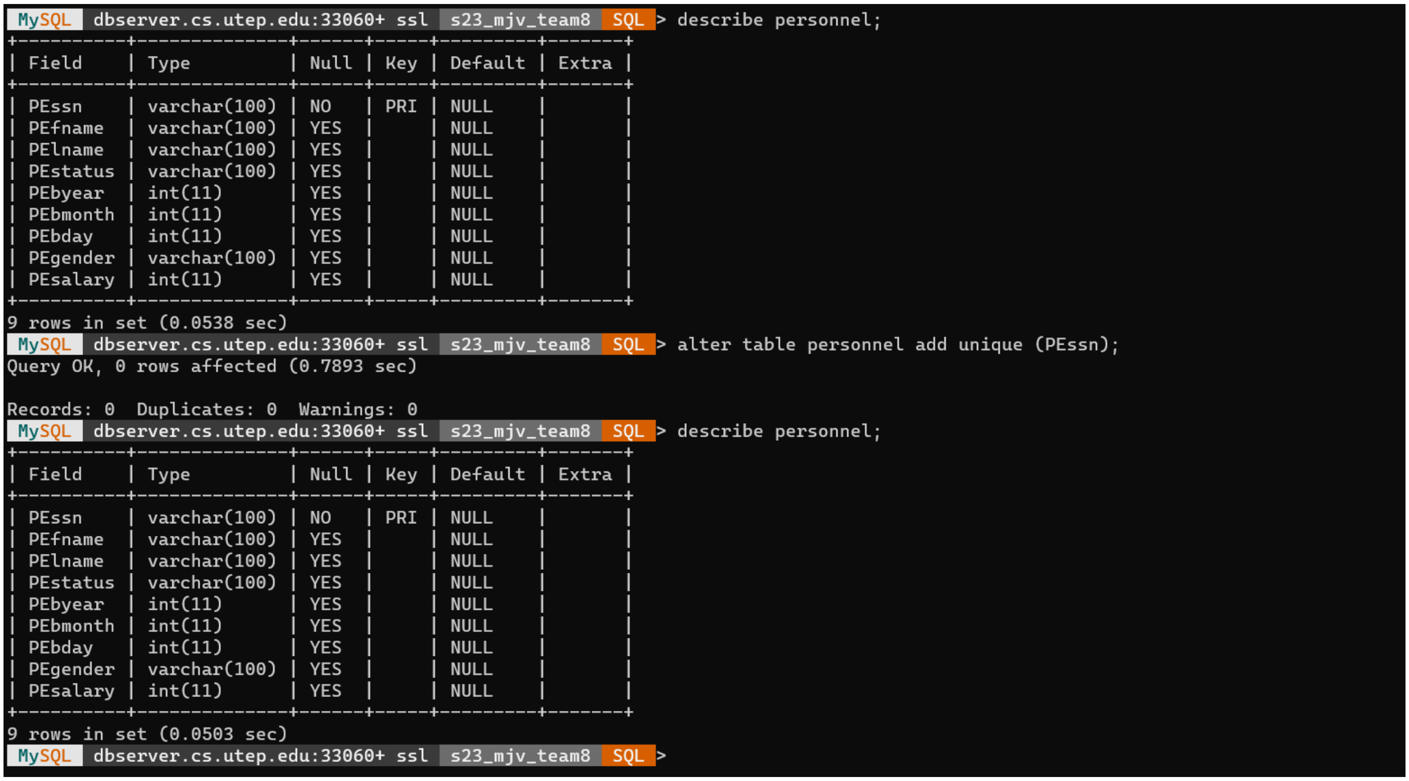


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > describe team_v2;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| Tcode         | int(11)      | NO   | PRI | NULL    |       |
| Tstatus       | varchar(100) | YES  |     | NULL    |       |
| SUPssn        | varchar(100) | YES  | MUL | NULL    |       |
| Tairport_code | int(11)      | YES  |     | NULL    |       |
| Tgate         | varchar(100) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
5 rows in set (0.0474 sec)
```

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > ALTER TABLE team_v2 MODIFY COLUMN SUPssn varchar(100) NOT NULL;
Query OK, 0 rows affected (1.6904 sec)

Records: 0  Duplicates: 0  Warnings: 0
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > describe team_v2;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| Tcode         | int(11)      | NO   | PRI | NULL    |       |
| Tstatus       | varchar(100) | YES  |     | NULL    |       |
| SUPssn        | varchar(100) | NO   | MUL | NULL    |       |
| Tairport_code | int(11)      | YES  |     | NULL    |       |
| Tgate         | varchar(100) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
5 rows in set (0.0512 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 50 R21 SQL Query. (SUPssn can no longer be Null)

R22. “The system will make sure that no two or more employees share the same social security number.” For this, we make sure that the PEssn in the personnel table has to be unique. By default, this attribute is unique because it is the primary key of the table.

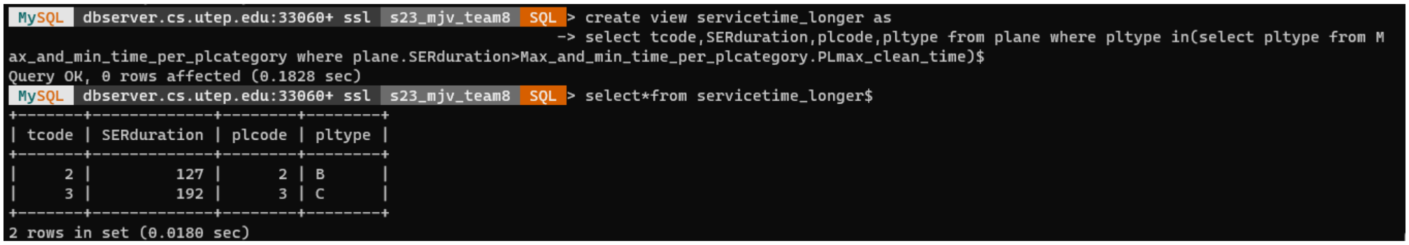


```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > describe personnel;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| PEssn    | varchar(100) | NO   | PRI | NULL    |       |
| PEfname  | varchar(100) | YES  |     | NULL    |       |
| PElname  | varchar(100) | YES  |     | NULL    |       |
| PEstatus | varchar(100) | YES  |     | NULL    |       |
| PEbyear  | int(11)      | YES  |     | NULL    |       |
| PEbmonth | int(11)      | YES  |     | NULL    |       |
| PEbday   | int(11)      | YES  |     | NULL    |       |
| PEgender | varchar(100) | YES  |     | NULL    |       |
| PEsalary | int(11)      | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
9 rows in set (0.0538 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > alter table personnel add unique (PEssn);
Query OK, 0 rows affected (0.7893 sec)

Records: 0  Duplicates: 0  Warnings: 0
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > describe personnel;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| PEssn    | varchar(100) | NO   | PRI | NULL    |       |
| PEfname  | varchar(100) | YES  |     | NULL    |       |
| PElname  | varchar(100) | YES  |     | NULL    |       |
| PEstatus | varchar(100) | YES  |     | NULL    |       |
| PEbyear  | int(11)      | YES  |     | NULL    |       |
| PEbmonth | int(11)      | YES  |     | NULL    |       |
| PEbday   | int(11)      | YES  |     | NULL    |       |
| PEgender | varchar(100) | YES  |     | NULL    |       |
| PEsalary | int(11)      | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
9 rows in set (0.0503 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL >
```

Figure 51 R22 SQL Query.

9. VIEWS

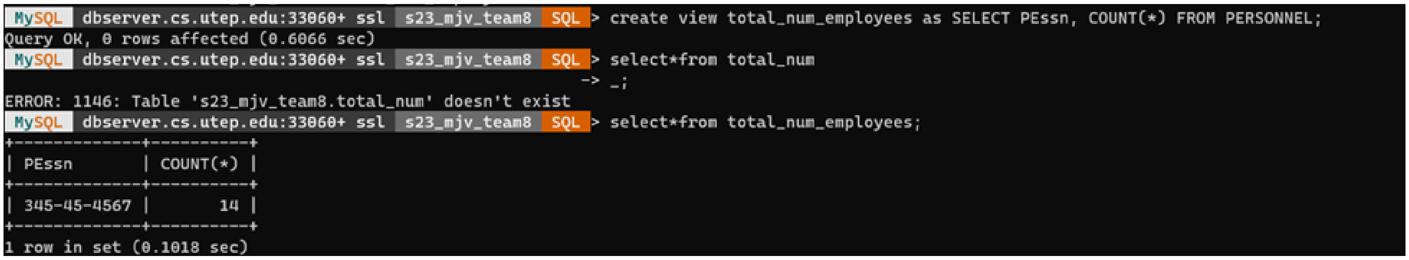
The figure 52 shows the view servicetime_longer which displays the team code, duration of service, code and type of the plane.



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create view servicetime_longer as
                                                          -> select tcode,SERduration,plcode,pltype from plane where pltype in(select pltype from M
ax_and_min_time_per_plcategory where plane.SERduration>Max_and_min_time_per_plcategory.PLmax_clean_time)$
Query OK, 0 rows affected (0.1828 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from servicetime_longer$
+-------+-------------+--------+--------+
| tcode | SERduration | plcode | pltype |
+-------+-------------+--------+--------+
|     2 |         127 |      2 | B      |
|     3 |         192 |      3 | C      |
+-------+-------------+--------+--------+
2 rows in set (0.0180 sec)
```

Figure 52 View: servicetime_longer

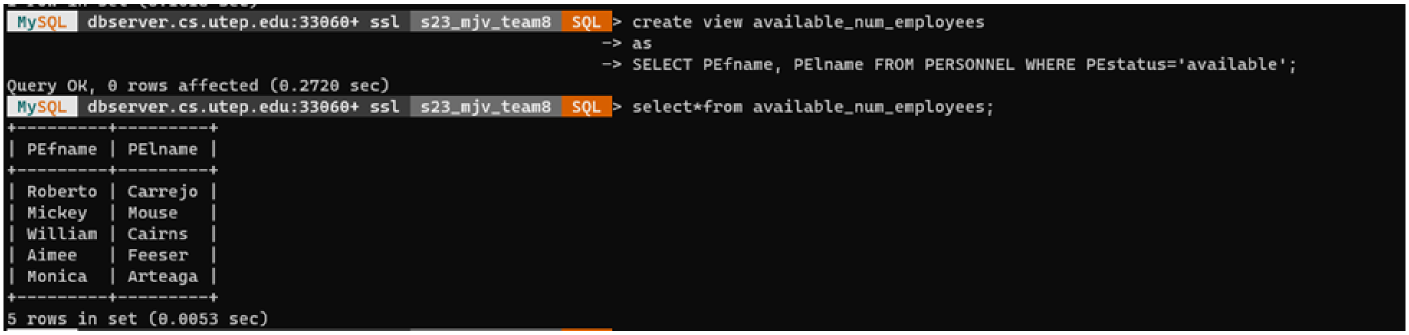
The figure 53 shows the view total_num_employees, it displays the total number of employees and its Social Security Number.



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create view total_num_employees as SELECT PEssn, COUNT(*) FROM PERSONNEL;
Query OK, 0 rows affected (0.6066 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from total_num
                                                          -> _;
ERROR: 1146: Table 's23_mjv_team8.total_num' doesn't exist
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from total_num_employees;
+-------------+----------+
| PEssn       | COUNT(*) |
+-------------+----------+
| 345-45-4567 |       14 |
+-------------+----------+
1 row in set (0.1018 sec)
```

Figure 53 View: total_num_employees

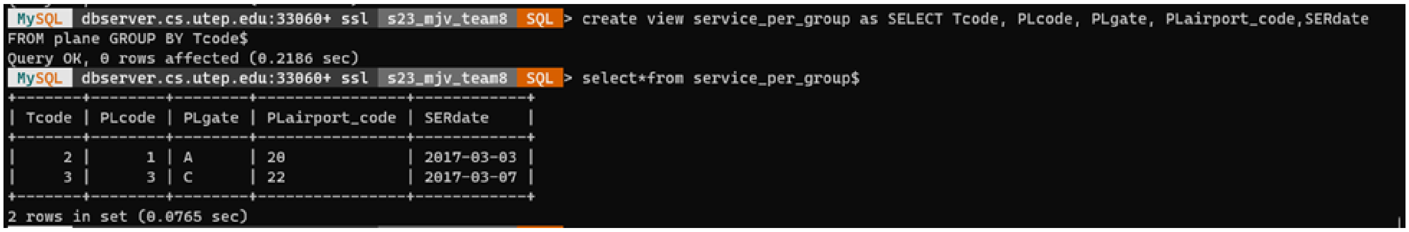
The figure 54 shows the view available_num_employees, it displays the total number of employees and its Social Security Number.



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create view available_num_employees
                                                          -> as
                                                          -> SELECT PEfname, PElname FROM PERSONNEL WHERE PEstatus='available';
Query OK, 0 rows affected (0.2720 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from available_num_employees;
+---------+---------+
| PEfname | PElname |
+---------+---------+
| Roberto | Carrejo |
| Mickey  | Mouse   |
| William | Cairns  |
| Aimee   | Feeser  |
| Monica  | Arteaga |
+---------+---------+
5 rows in set (0.0053 sec)
```

Figure 54 View: available_num_employees

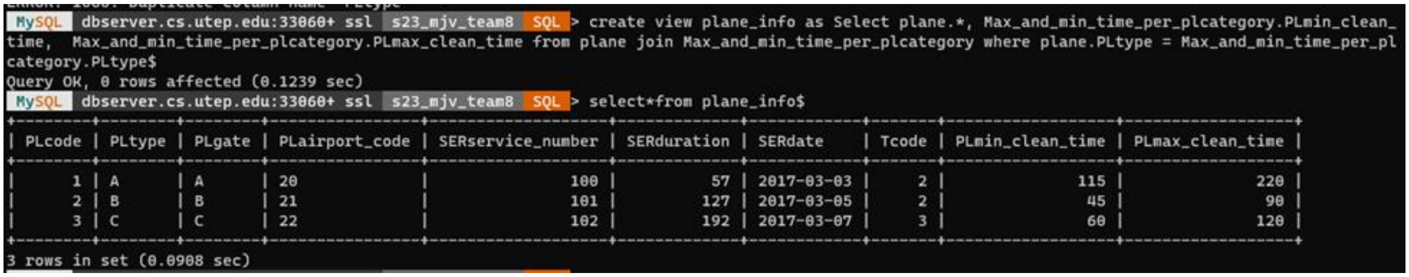
The figure 55 shows the view service_per_group it shows the information of the team that cleaned the plane id by its unique code the location of the service provided and the date of the service.



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create view service_per_group as SELECT Tcode, PLcode, PLgate, PLairport_code,SERdate
FROM plane GROUP BY Tcode$
Query OK, 0 rows affected (0.2186 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from service_per_group$
+-------+--------+--------+----------------+------------+
| Tcode | PLcode | PLgate | PLairport_code | SERdate    |
+-------+--------+--------+----------------+------------+
|     2 |      1 | A      | 20             | 2017-03-03 |
|     3 |      3 | C      | 22             | 2017-03-07 |
+-------+--------+--------+----------------+------------+
2 rows in set (0.0765 sec)
```

Figure 55 View: service_per_group

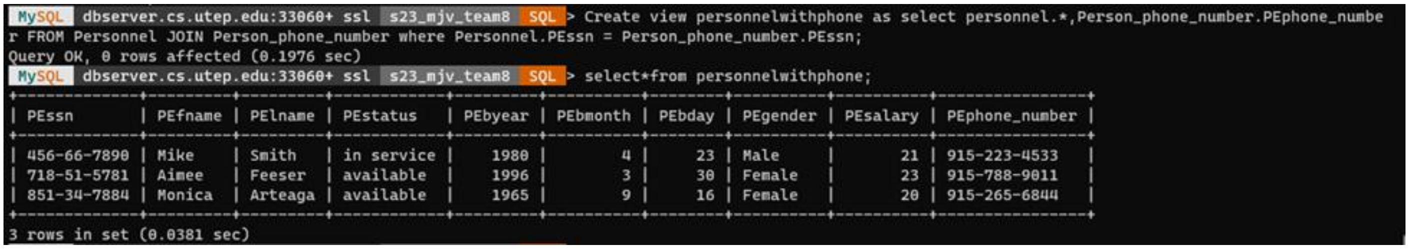
The figure 56 shows the view plane_info which joins the table plane and the maximum and minimum time expected per service.



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create view plane_info as Select plane.*, Max_and_min_time_per_plcategory.PLmin_clean_
time,  Max_and_min_time_per_plcategory.PLmax_clean_time from plane join Max_and_min_time_per_plcategory where plane.PLtype = Max_and_min_time_per_pl
category.PLtype$
Query OK, 0 rows affected (0.1239 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from plane_info$
+--------+--------+--------+----------------+-------------------+-------------+------------+-------+------------------+------------------+
| PLcode | PLtype | PLgate | PLairport_code | SERservice_number | SERduration | SERdate    | Tcode | PLmin_clean_time | PLmax_clean_time |
+--------+--------+--------+----------------+-------------------+-------------+------------+-------+------------------+------------------+
|      1 | A      | A      | 20             |               100 |          57 | 2017-03-03 |     2 |              115 |              220 |
|      2 | B      | B      | 21             |               101 |         127 | 2017-03-05 |     2 |               45 |               90 |
|      3 | C      | C      | 22             |               102 |         192 | 2017-03-07 |     3 |               60 |              120 |
+--------+--------+--------+----------------+-------------------+-------------+------------+-------+------------------+------------------+
3 rows in set (0.0908 sec)
```

Figure 56 View: plane_info

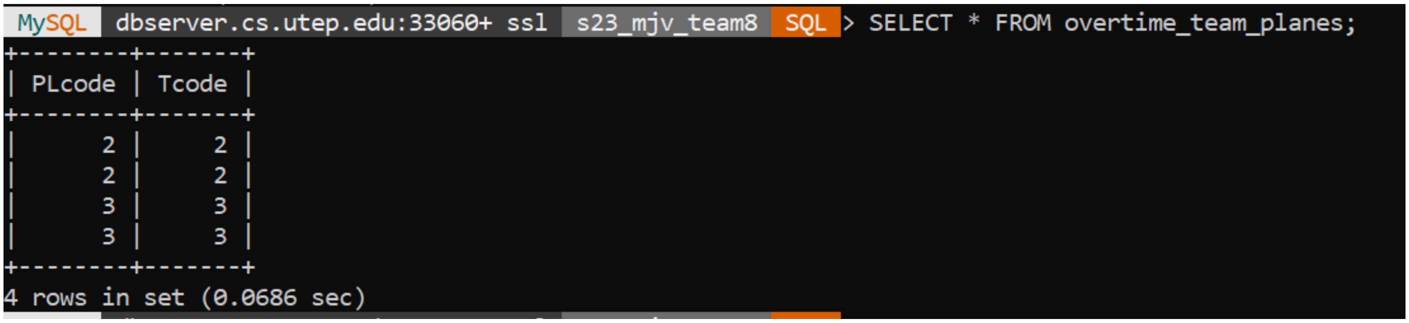
The figure 57 shows personnelwithphone view that displays the information of the employees including their phone number.



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > Create view personnelwithphone as select personnel.*,Person_phone_number.PEphone_numbe
r FROM Personnel JOIN Person_phone_number where Personnel.PEssn = Person_phone_number.PEssn;
Query OK, 0 rows affected (0.1976 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from personnelwithphone;
+-------------+---------+---------+------------+---------+----------+--------+----------+----------+-----------------+
| PEssn       | PEfname | PElname | PEstatus   | PEbyear | PEbmonth | PEbday | PEgender | PEsalary | PEphone_number  |
+-------------+---------+---------+------------+---------+----------+--------+----------+----------+-----------------+
| 456-66-7890 | Mike    | Smith   | in service |    1980 |        4 |     23 | Male     |       21 | 915-223-4533    |
| 718-51-5781 | Aimee   | Feeser  | available  |    1996 |        3 |     30 | Female   |       23 | 915-788-9011    |
| 851-34-7884 | Monica  | Arteaga | available  |    1965 |        9 |     16 | Female   |       20 | 915-265-6844    |
+-------------+---------+---------+------------+---------+----------+--------+----------+----------+-----------------+
3 rows in set (0.0381 sec)
```

Figure 57 View: personnelwithphone

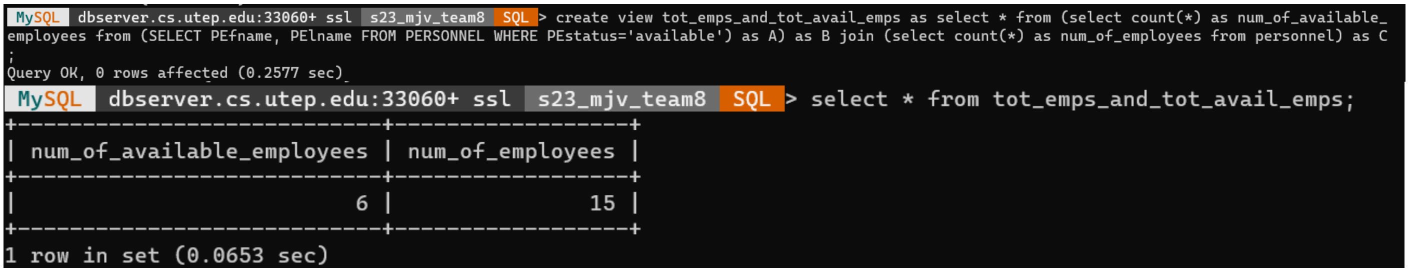
The view below shows teams that spent longer than the maximum time allotted to clean a plane.



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT * FROM overtime_team_planes;
+--------+-------+
| PLcode | Tcode |
+--------+-------+
|      2 |     2 |
|      2 |     2 |
|      3 |     3 |
|      3 |     3 |
+--------+-------+
4 rows in set (0.0686 sec)
```

Figure 58 View: overtime_team_planes

The following view was created by Chris. It displays two columns with the total number of available employees and the total number of employees.

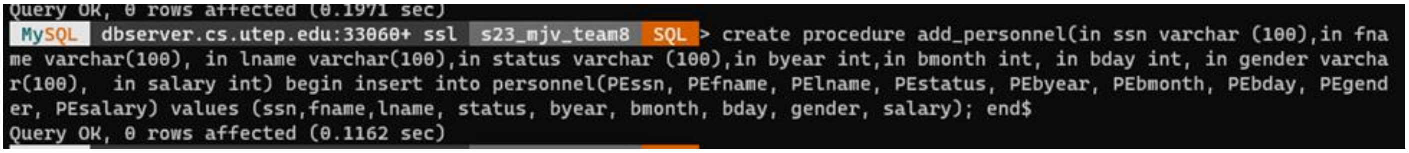


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create view tot_emps_and_tot_avail_emps as select * from (select count(*) as num_of_available_
employees from (SELECT PEfname, PElname FROM PERSONNEL WHERE PEstatus='available') as A) as B join (select count(*) as num_of_employees from personnel) as C
;
Query OK, 0 rows affected (0.2577 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from tot_emps_and_tot_avail_emps;
+---------------------------+------------------+
| num_of_available_employees | num_of_employees |
+---------------------------+------------------+
|                         6 |               15 |
+---------------------------+------------------+
1 row in set (0.0653 sec)
```

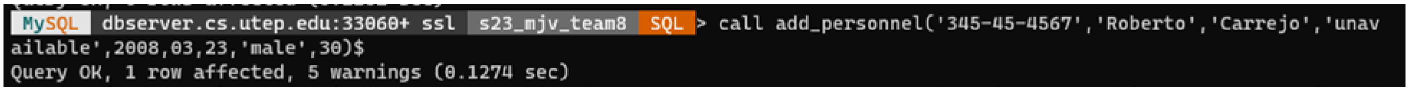
Figure 59 View: tot_emps_and_tot_avail_emps

10. PROCEDURES

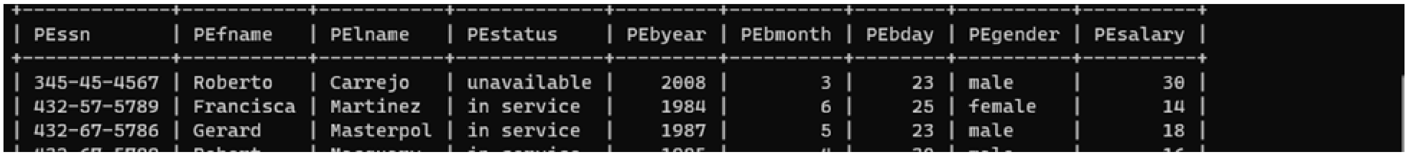
Figure 60 show the procedure add_personnel to input the information required in the personnel table.



```
Query OK, 0 rows affected (0.1971 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create procedure add_personnel(in ssn varchar (100),in fna
me varchar(100), in lname varchar(100),in status varchar (100),in byear int,in bmonth int, in bday int, in gender varcha
r(100),  in salary int) begin insert into personnel(PEssn, PEfname, PElname, PEstatus, PEbyear, PEbmonth, PEbday, PEgend
er, PEsalary) values (ssn,fname,lname, status, byear, bmonth, bday, gender, salary); end$
Query OK, 0 rows affected (0.1162 sec)
```



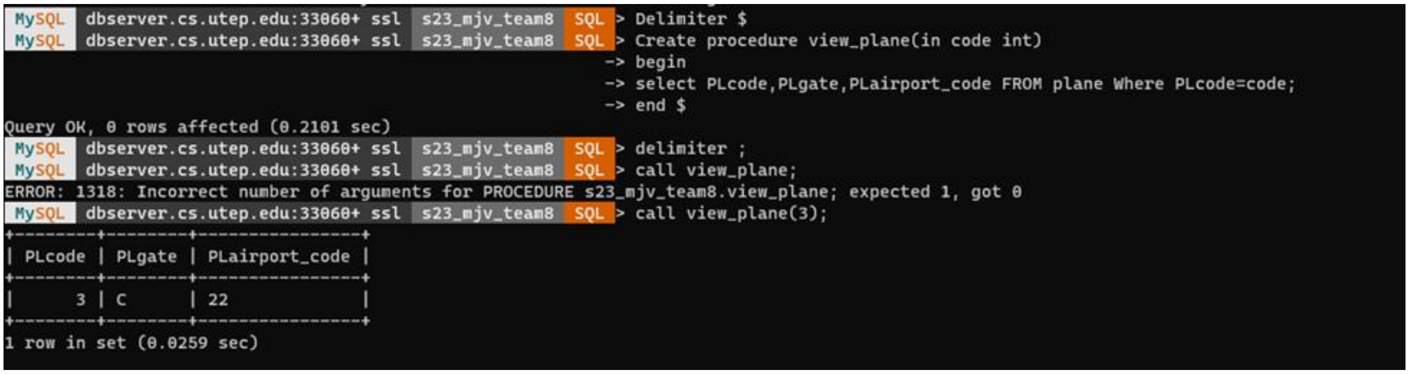
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call add_personnel('345-45-4567','Roberto','Carrejo','unav
ailable',2008,03,23,'male',30)$
Query OK, 1 row affected, 5 warnings (0.1274 sec)
```



```
+--------------+-----------+-----------+-------------+---------+----------+--------+----------+----------+
| PEssn        | PEfname   | PElname   | PEstatus    | PEbyear | PEbmonth | PEbday | PEgender | PEsalary |
+--------------+-----------+-----------+-------------+---------+----------+--------+----------+----------+
| 345-45-4567  | Roberto   | Carrejo   | unavailable |    2008 |        3 |     23 | male     |       30 |
| 432-57-5789  | Francisca | Martinez  | in service  |    1984 |        6 |     25 | female   |       14 |
| 432-67-5786  | Gerard    | Masterpol | in service  |    1987 |        5 |     23 | male     |       18 |
```

Figure 60 Procedure: add_personnel

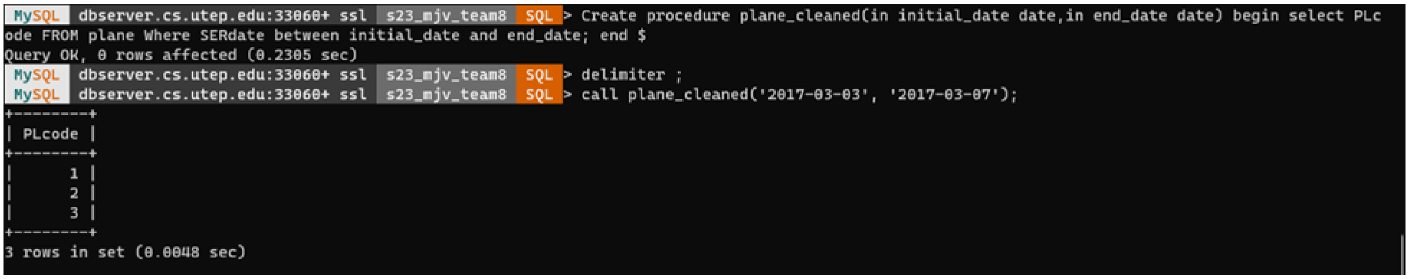
Figure 61 shows the procedure view_plane that generates a report to keep track of the plane's location via airport code and gate.



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > Delimiter $
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > Create procedure view_plane(in code int)
                                                         -> begin
                                                         -> select PLcode,PLgate,PLairport_code FROM plane Where PLcode=code;
                                                         -> end $
Query OK, 0 rows affected (0.2101 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > delimiter ;
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call view_plane;
ERROR: 1318: Incorrect number of arguments for PROCEDURE s23_mjv_team8.view_plane; expected 1, got 0
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call view_plane(3);
+--------+--------+----------------+
| PLcode | PLgate | PLairport_code |
+--------+--------+----------------+
|      3 | C      | 22             |
+--------+--------+----------------+
1 row in set (0.0259 sec)
```

Figure 61 Procedure: view_plane

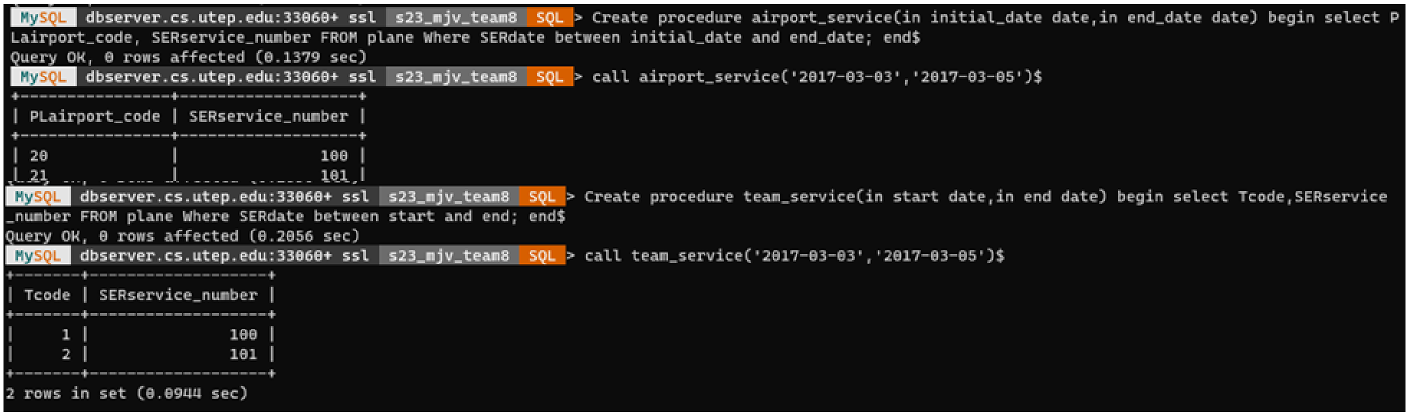
Figure 62 shows the procedure plane_cleaned. The system shall generate a report containing the planes that were cleaned for a specific period).



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > Create procedure plane_cleaned(in initial_date date,in end_date date) begin select PLc
ode FROM plane Where SERdate between initial_date and end_date; end $
Query OK, 0 rows affected (0.2305 sec)
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > delimiter ;
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call plane_cleaned('2017-03-03', '2017-03-07');
+--------+
| PLcode |
+--------+
|      1 |
|      2 |
|      3 |
+--------+
3 rows in set (0.0048 sec)
```

Figure 62 Procedure: plane_cleaned

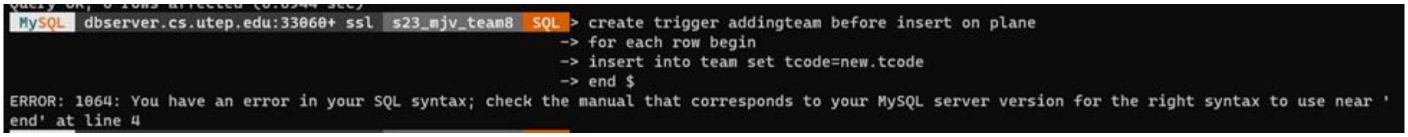
Figure 63 shows the procedure airport_service, team_service that satisfies R6 (The system should allow the general manager to generate a list of the services provided per plane, airport, group for a specified period.)



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > Create procedure airport_service(in initial_date date,in end_date date) begin select P
Lairport_code, SERservice_number FROM plane Where SERdate between initial_date and end_date; end$
Query OK, 0 rows affected (0.1379 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call airport_service('2017-03-03','2017-03-05')$
+----------------+-------------------+
| PLairport_code | SERservice_number |
+----------------+-------------------+
| 20             |               100 |
| 21             |               101 |
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > Create procedure team_service(in start date,in end date) begin select Tcode,SERservice
_number FROM plane Where SERdate between start and end; end$
Query OK, 0 rows affected (0.2056 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call team_service('2017-03-03','2017-03-05')$
+-------+-------------------+
| Tcode | SERservice_number |
+-------+-------------------+
|     1 |               100 |
|     2 |               101 |
+-------+-------------------+
2 rows in set (0.0944 sec)
```

Figure 63 Procedure: airport_service and team_service

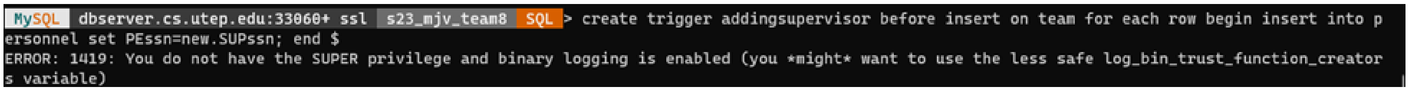
Figure 64 shows the trigger addingteam that adds the team code to the team table before inserting a new plane.



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create trigger addingteam before insert on plane
                                                            -> for each row begin
                                                            -> insert into team set tcode=new.tcode
                                                            -> end $
ERROR: 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '
end' at line 4
```

Figure 64 Trigger: addingteam

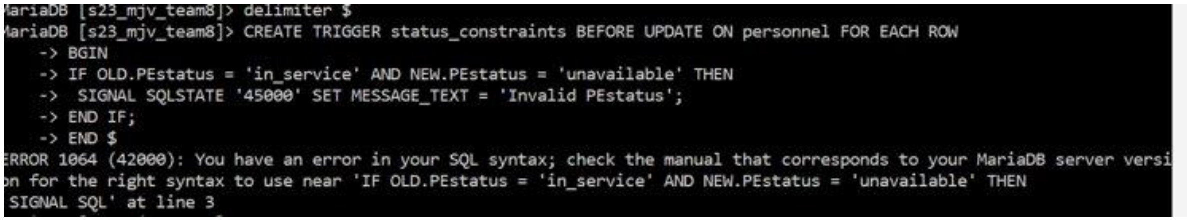
Figure 65 shows the trigger addingsupervisor that adds the team code to the team table before inserting a new plane.



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create trigger addingsupervisor before insert on team for each row begin insert into p
ersonnel set PEssn=new.SUPssn; end $
ERROR: 1419: You do not have the SUPER privilege and binary logging is enabled (you *might* want to use the less safe log_bin_trust_function_creator
s variable)
```

Figure 65 Trigger: addingsupervisor

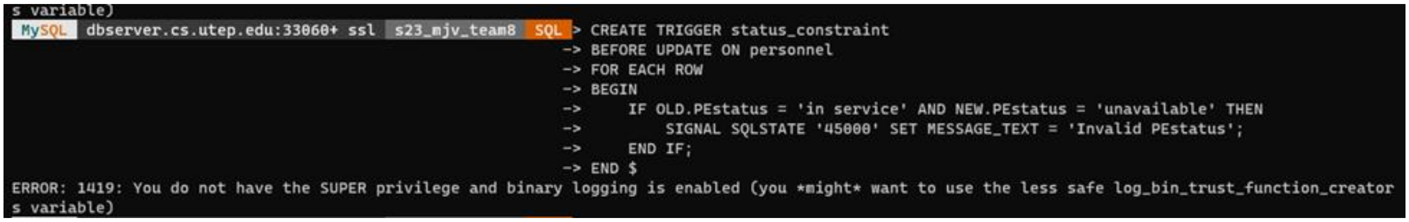
The figure 66 shows the trigger changingstatus that updates the status of a member in the personnel table after being assigned to a team.



```
lariaDB [s23_mjv_team8]> delimiter $
lariaDB [s23_mjv_team8]> CREATE TRIGGER status_constraints BEFORE UPDATE ON personnel FOR EACH ROW
    -> BGIN
    -> IF OLD.PEstatus = 'in_service' AND NEW.PEstatus = 'unavailable' THEN
    ->  SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid PEstatus';
    -> END IF;
    -> END $
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server versi
on for the right syntax to use near 'IF OLD.PEstatus = 'in_service' AND NEW.PEstatus = 'unavailable' THEN
 SIGNAL SQL' at line 3
```

Figure 66 Trigger: changingstatus

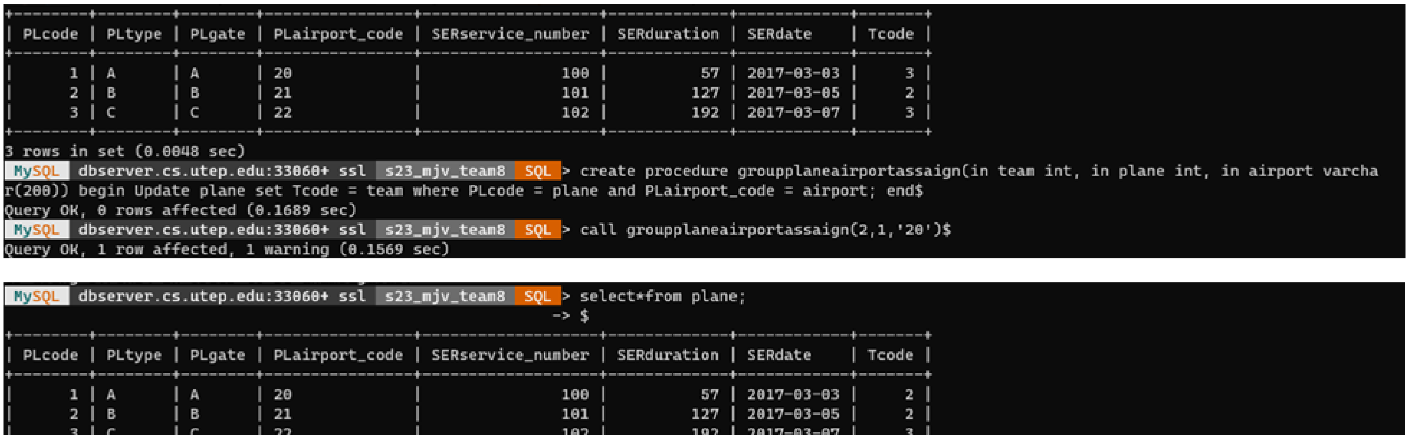
Figure 67 shows the trigger status_constraint the trigger won't allow to change the status of an employee that is in service to unavailable.



```
s variable)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > CREATE TRIGGER status_constraint
                                                            -> BEFORE UPDATE ON personnel
                                                            -> FOR EACH ROW
                                                            -> BEGIN
                                                            ->     IF OLD.PEstatus = 'in service' AND NEW.PEstatus = 'unavailable' THEN
                                                            ->         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid PEstatus';
                                                            ->     END IF;
                                                            -> END $
ERROR: 1419: You do not have the SUPER privilege and binary logging is enabled (you *might* want to use the less safe log_bin_trust_function_creator
s variable)
```

Figure 67 Trigger: status_constraint

Figure 68 shows the procedure groupplaneairportassaigns the procedures help change the assigns a group to a plane in a specific airport.

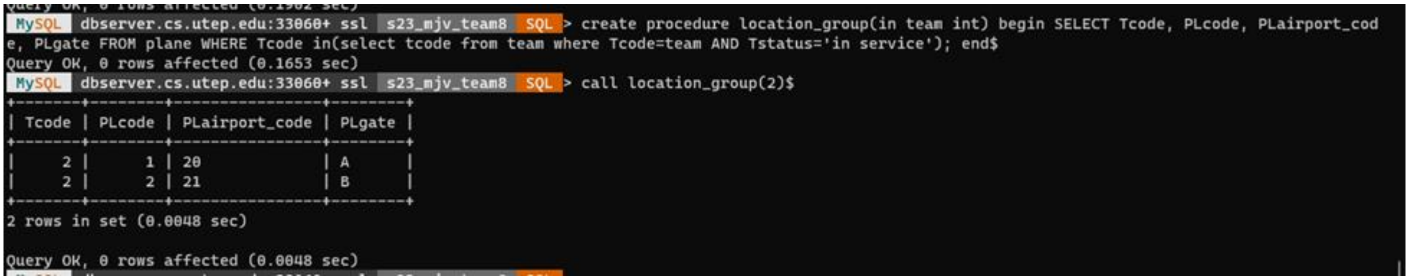


```
+--------+--------+--------+----------------+--------------------+-------------+------------+-------+
| PLcode | PLtype | PLgate | PLairport_code | SERservice_number | SERduration | SERdate    | Tcode |
+--------+--------+--------+----------------+--------------------+-------------+------------+-------+
|      1 | A      | A      | 20             |                100 |          57 | 2017-03-03 |     3 |
|      2 | B      | B      | 21             |                101 |         127 | 2017-03-05 |     2 |
|      3 | C      | C      | 22             |                102 |         192 | 2017-03-07 |     3 |
+--------+--------+--------+----------------+--------------------+-------------+------------+-------+
3 rows in set (0.0048 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create procedure groupplaneairportassaign(in team int, in plane int, in airport varcha
r(200)) begin Update plane set Tcode = team where PLcode = plane and PLairport_code = airport; end$
Query OK, 0 rows affected (0.1689 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call groupplaneairportassaign(2,1,'20')$
Query OK, 1 row affected, 1 warning (0.1569 sec)
```

```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select*from plane;
                                                          -> $
+--------+--------+--------+----------------+--------------------+-------------+------------+-------+
| PLcode | PLtype | PLgate | PLairport_code | SERservice_number | SERduration | SERdate    | Tcode |
+--------+--------+--------+----------------+--------------------+-------------+------------+-------+
|      1 | A      | A      | 20             |                100 |          57 | 2017-03-03 |     2 |
|      2 | B      | B      | 21             |                101 |         127 | 2017-03-05 |     2 |
|      3 | C      | C      | 22             |                102 |         192 | 2017-03-07 |     3 |
```

Figure 68 Procedure: groupplaneairportassaign

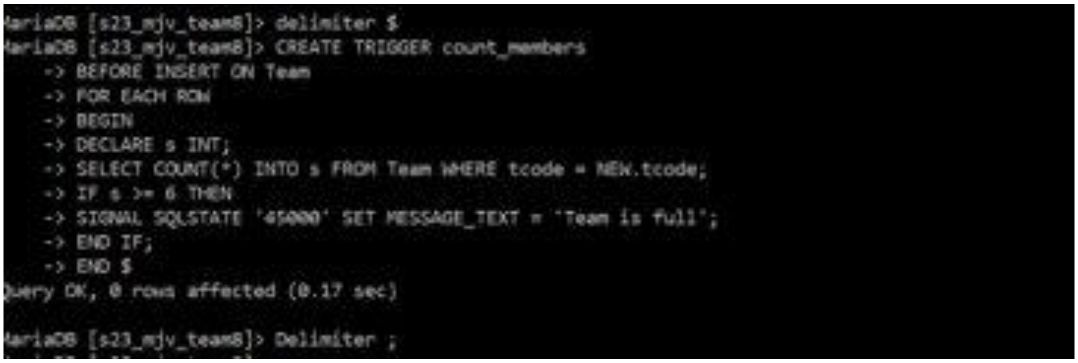
Figure 69 shows the procedure location_group that displays the information to know where a specific group is located (plane code, airport and gate).



```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create procedure location_group(in team int) begin SELECT Tcode, PLcode, PLairport_cod
e, PLgate FROM plane WHERE Tcode in(select tcode from team where Tcode=team AND Tstatus='in service'); end$
Query OK, 0 rows affected (0.1653 sec)
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call location_group(2)$
+-------+--------+----------------+--------+
| Tcode | PLcode | PLairport_code | PLgate |
+-------+--------+----------------+--------+
|     2 |      1 | 20             | A      |
|     2 |      2 | 21             | B      |
+-------+--------+----------------+--------+
2 rows in set (0.0048 sec)

Query OK, 0 rows affected (0.0048 sec)
```

Figure 69 Procedure: location_group

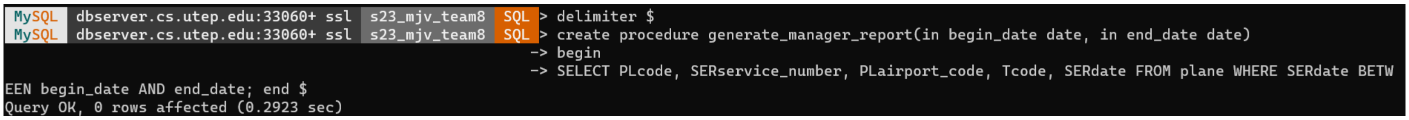
The figure 70 shows the trigger that won't allow another employee to the members table if the number of employees of a team is equal or greater than 6.



```
MariaDB [s23_mjv_team8]> delimiter $
MariaDB [s23_mjv_team8]> CREATE TRIGGER count_members
    -> BEFORE INSERT ON Team
    -> FOR EACH ROW
    -> BEGIN
    -> DECLARE s INT;
    -> SELECT COUNT(*) INTO s FROM Team WHERE tcode = NEW.tcode;
    -> IF s >= 6 THEN
    -> SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Team is full';
    -> END IF;
    -> END $
Query OK, 0 rows affected (0.17 sec)

MariaDB [s23_mjv_team8]> Delimiter ;
```

Figure 70 Procedure: count_members

The following procedure was created by Chris. It generates a report for the manager indicating the plane code, its service number, it airport code location, team code that serviced it, and service date for all planes that were serviced between two given dates.



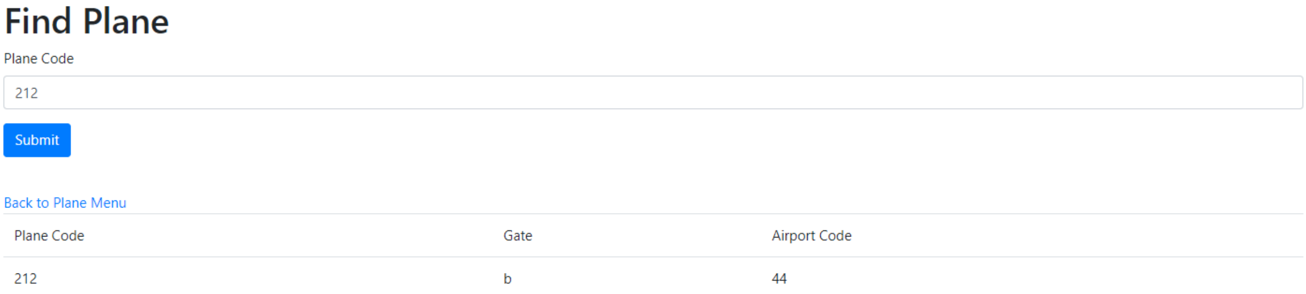
```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > delimiter $
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > create procedure generate_manager_report(in begin_date date, in end_date date)
                                                          -> begin
                                                          -> SELECT PLcode, SERservice_number, PLairport_code, Tcode, SERdate FROM plane WHERE SERdate BETW
EEN begin_date AND end_date; end $
Query OK, 0 rows affected (0.2923 sec)
```

Figure 71 Procedure: generate_manager_report

11. REPORTS (Need to add sql commands)

1. From R3, below you can see a report to find a plane's location via airport code and gate.

-Plane>>-Find Plane Location



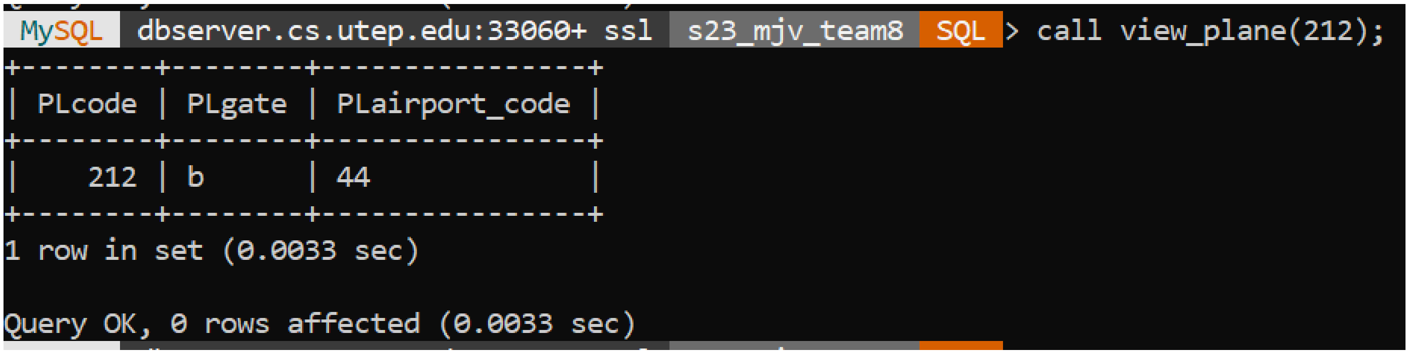
Find Plane

Plane Code

212

Plane Code	Gate	Airport Code
212	b	44

Figure 72 Report 1 GUI

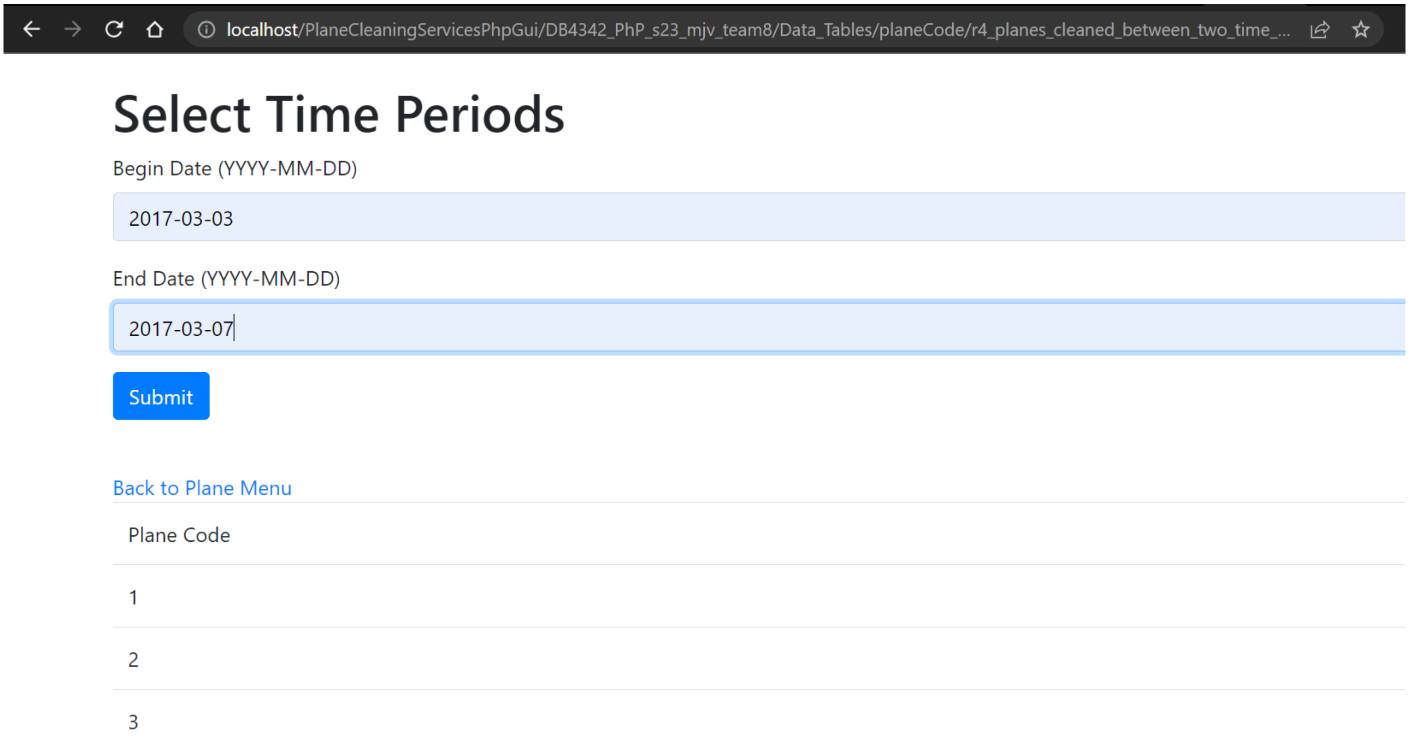


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call view_plane(212);
+--------+--------+-----------------+
| PLcode | PLgate | PLairport_code |
+--------+--------+-----------------+
|    212 | b      | 44              |
+--------+--------+-----------------+
1 row in set (0.0033 sec)

Query OK, 0 rows affected (0.0033 sec)
```

Figure 73 Report 1 SQL Query

2. From R4, below you can see a report containing the planes that were cleaned for a specific period.

-Plane>> -Choose Date Period Between Which Planes Were Cleaned


localhost/PlaneCleaningServicesPhpGui/DB4342_PhP_s23_mjv_team8/Data_Tables/planeCode/r4_planes_cleaned_between_two_time_...

Select Time Periods

Begin Date (YYYY-MM-DD)

2017-03-03

End Date (YYYY-MM-DD)

2017-03-07

Back to Plane Menu

Plane Code
1
2
3

Figure 74 Report 2 GUI

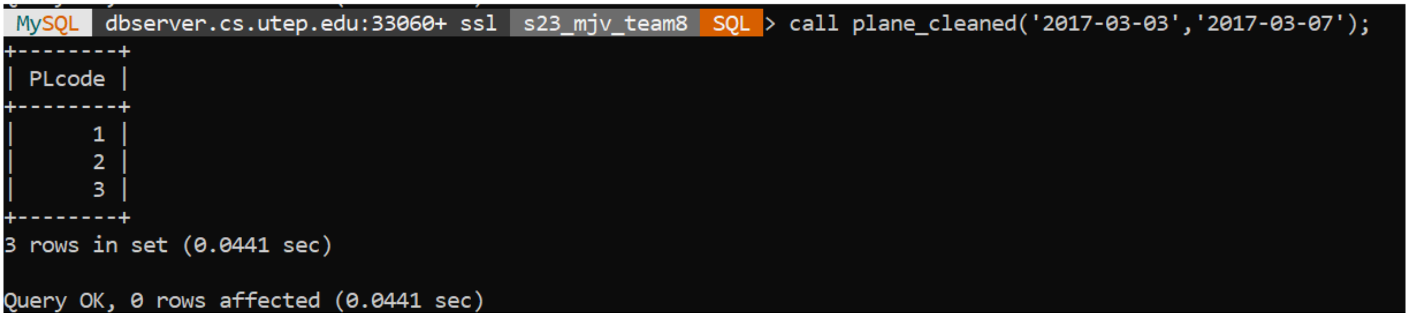
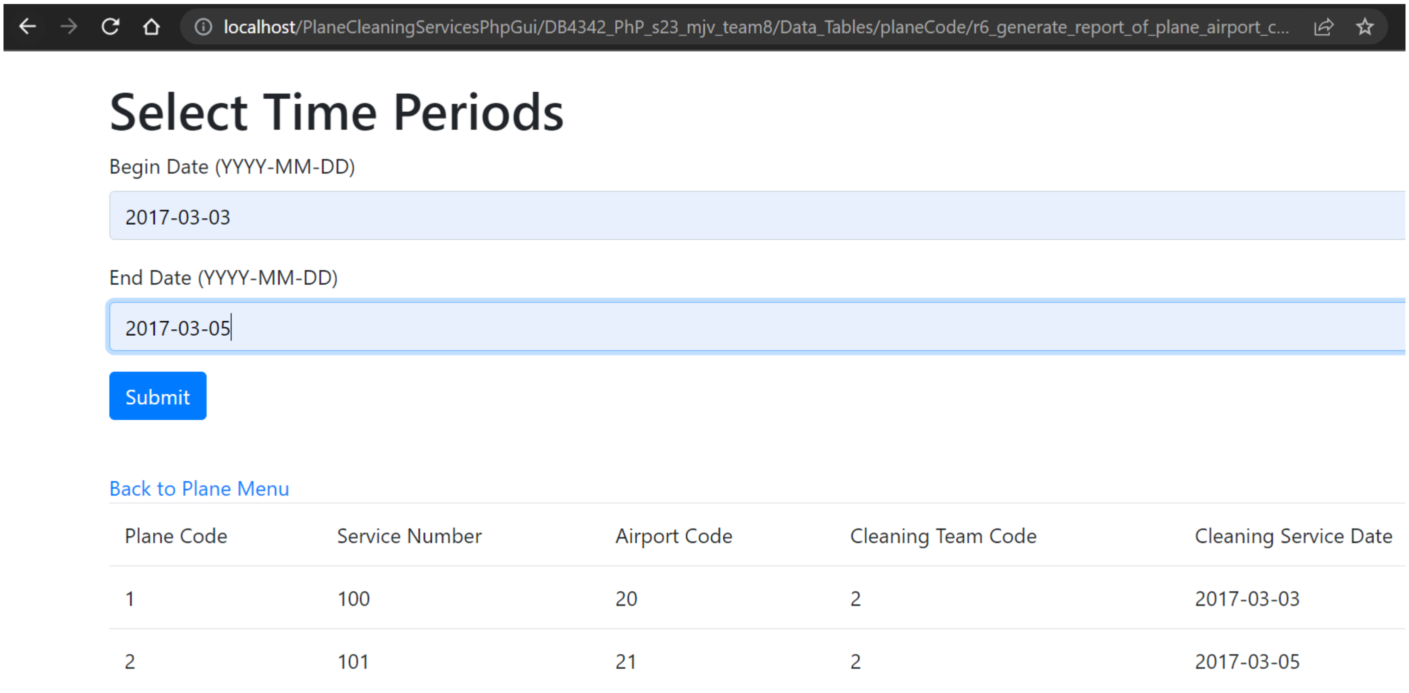

```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > call plane_cleaned('2017-03-03','2017-03-07');
+--------+
| PLcode |
+--------+
|      1 |
|      2 |
|      3 |
+--------+
3 rows in set (0.0441 sec)

Query OK, 0 rows affected (0.0441 sec)
```

Figure 75 Report 2 SQL Query

3. From R6, below you can see a report containing a list of provided services for a specific period. These include the plane, airport, and team.

-Plane>> -Manager Report For Plane Info. Between Time Periods



localhost/PlaneCleaningServicesPhpGui/DB4342_PhP_s23_mjv_team8/Data_Tables/planeCode/r6_generate_report_of_plane_airport_c...

Select Time Periods

Begin Date (YYYY-MM-DD)

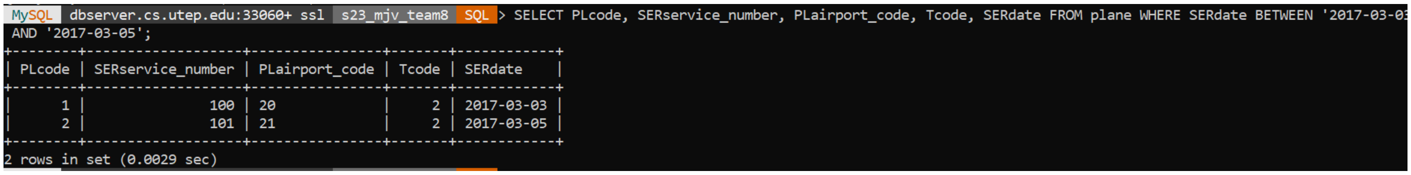
2017-03-03

End Date (YYYY-MM-DD)

2017-03-05

Plane Code	Service Number	Airport Code	Cleaning Team Code	Cleaning Service Date
1	100	20	2	2017-03-03
2	101	21	2	2017-03-05

Figure 76 Report 3 GUI

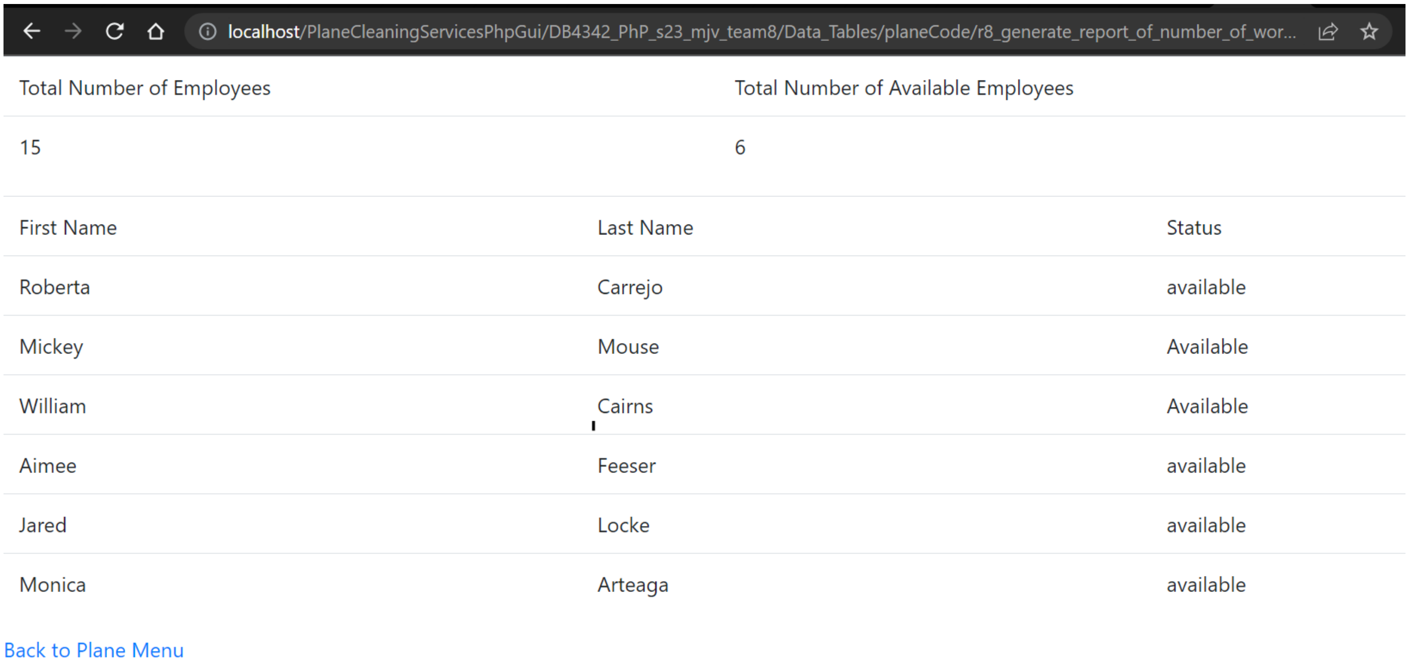


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT PLcode, SERservice_number, PLairport_code, Tcode, SERdate FROM plane WHERE SERdate BETWEEN '2017-03-03
 AND '2017-03-05';
+--------+-------------------+----------------+-------+------------+
| PLcode | SERservice_number | PLairport_code | Tcode | SERdate    |
+--------+-------------------+----------------+-------+------------+
|      1 |               100 | 20             |     2 | 2017-03-03 |
|      2 |               101 | 21             |     2 | 2017-03-05 |
+--------+-------------------+----------------+-------+------------+
2 rows in set (0.0029 sec)
```

Figure 77 Report 3 SQL Query

4. From R8, below you can see a report showing the total number of employees, and the current number of available employees.

-Personnel>> -Number of Worker, Number of Available Workers and List of Available Workers



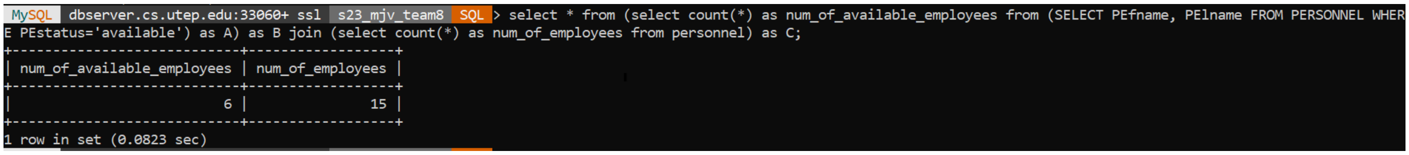
localhost/PlaneCleaningServicesPhpGui/DB4342_PhP_s23_mjv_team8/Data_Tables/planeCode/r8_generate_report_of_number_of_wor...

Total Number of Employees	Total Number of Available Employees
15	6

First Name	Last Name	Status
Roberta	Carrejo	available
Mickey	Mouse	Available
William	Cairns	Available
Aimee	Feeser	available
Jared	Locke	available
Monica	Arteaga	available

Back to Plane Menu

Figure 78 Report 4 GUI

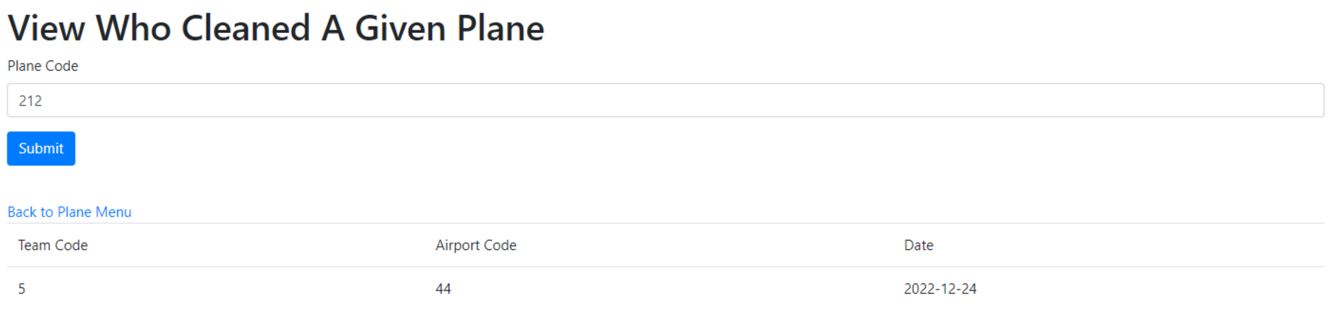


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > select * from (select count(*) as num_of_available_employees from (SELECT PEfname, PElname FROM PERSONNEL WHER
E PEstatus='available') as A) as B join (select count(*) as num_of_employees from personnel) as C;
+----------------------------+------------------+
| num_of_available_employees | num_of_employees |
+----------------------------+------------------+
|                          6 |               15 |
+----------------------------+------------------+
1 row in set (0.0823 sec)
```

Figure 79 Report 4 SQL Query

5. From R13, below you can see which group cleaned a specific plane at what time.

-Plane>> -View Who Cleaned a Given Plane



View Who Cleaned A Given Plane

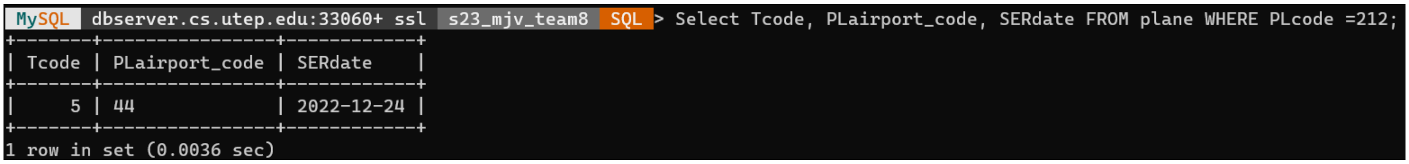
Plane Code

212

Back to Plane Menu

Team Code	Airport Code	Date
5	44	2022-12-24

Figure 80 Report 5 GUI

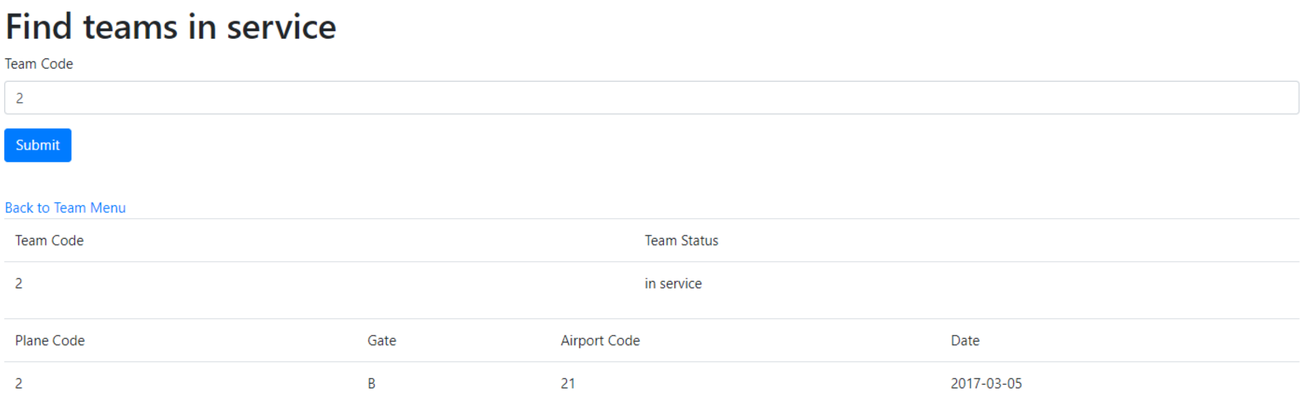


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > Select Tcode, PLairport_code, SERdate FROM plane WHERE PLcode =212;
+-------+----------------+------------+
| Tcode | PLairport_code | SERdate    |
+-------+----------------+------------+
|     5 | 44             | 2022-12-24 |
+-------+----------------+------------+
1 row in set (0.0036 sec)
```

Figure 81 Report 5 SQL Query

6. From R14, this report gives us the location of a team in service after inputting the team’s code, also showing us a table of the in-service teams from which we can choose from.

-Team>> -Find Team Location



Find teams in service

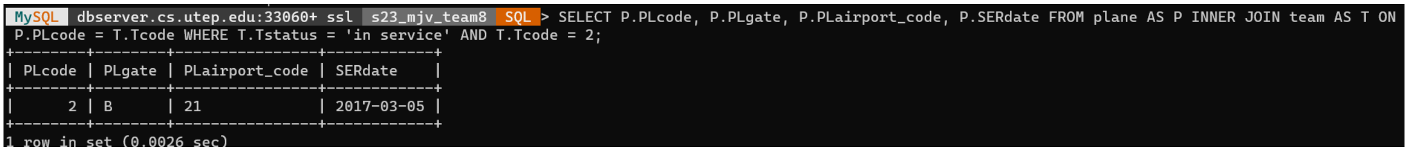
Team Code

2

Team Code	Team Status
2	in service

Plane Code	Gate	Airport Code	Date
2	B	21	2017-03-05

Figure 82 Report 6 GUI

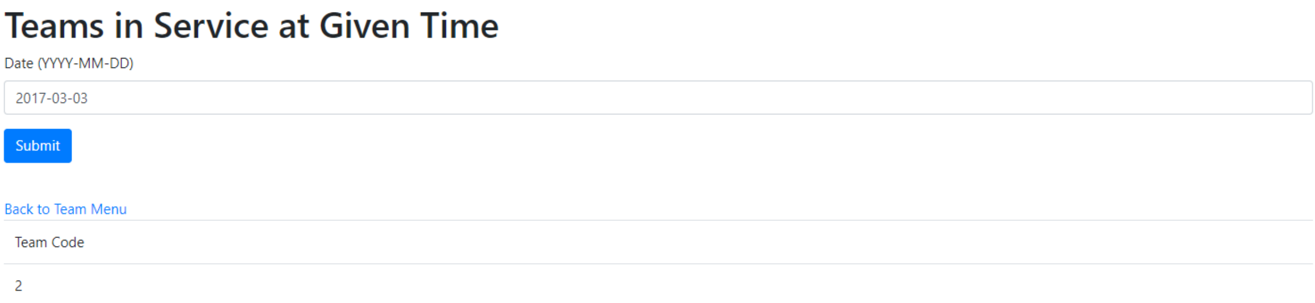


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT P.PLcode, P.PLgate, P.PLairport_code, P.SERdate FROM plane AS P INNER JOIN team AS T ON
 P.PLcode = T.Tcode WHERE T.Tstatus = 'in service' AND T.Tcode = 2;
+--------+--------+-----------------+------------+
| PLcode | PLgate | PLairport_code  | SERdate    |
+--------+--------+-----------------+------------+
|      2 | B      | 21              | 2017-03-05 |
+--------+--------+-----------------+------------+
1 row in set (0.0026 sec)
```

Figure 83 Report 6 SQL Query

7. From R15, this report tells us what teams were in service at a given date.

-Team>> -Teams in Service at Given Time



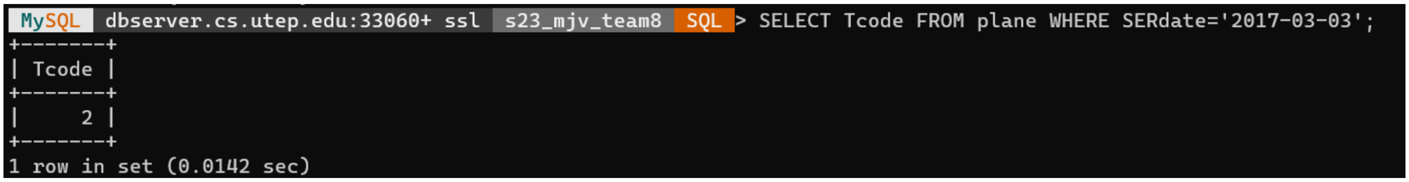
Teams in Service at Given Time

Date (YYYY-MM-DD)

2017-03-03

Team Code
2

Figure 84 Report 7 GUI

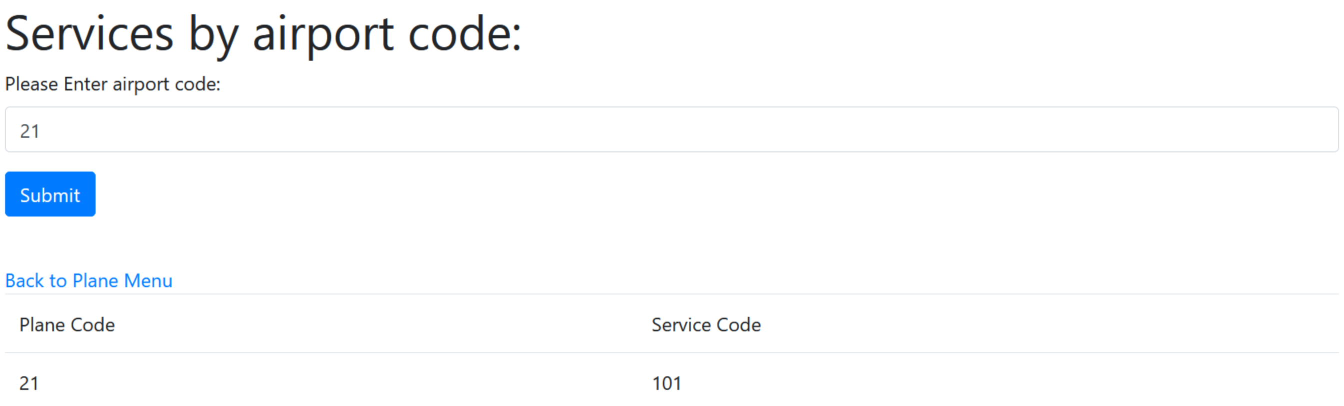


```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT Tcode FROM plane WHERE SERdate='2017-03-03';
+-------+
| Tcode |
+-------+
|     2 |
+-------+
1 row in set (0.0142 sec)
```

Figure 85 Report 7 SQL Query

8. From R17, below you see the list of services per airport where the airport code is taken as input.

-Plane>> -Airport_service



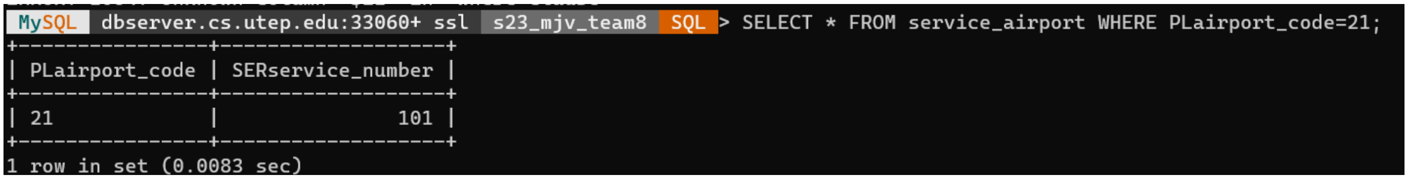
Services by airport code:

Please Enter airport code:

21

Plane Code	Service Code
21	101

Figure 86 Report 8 GUI

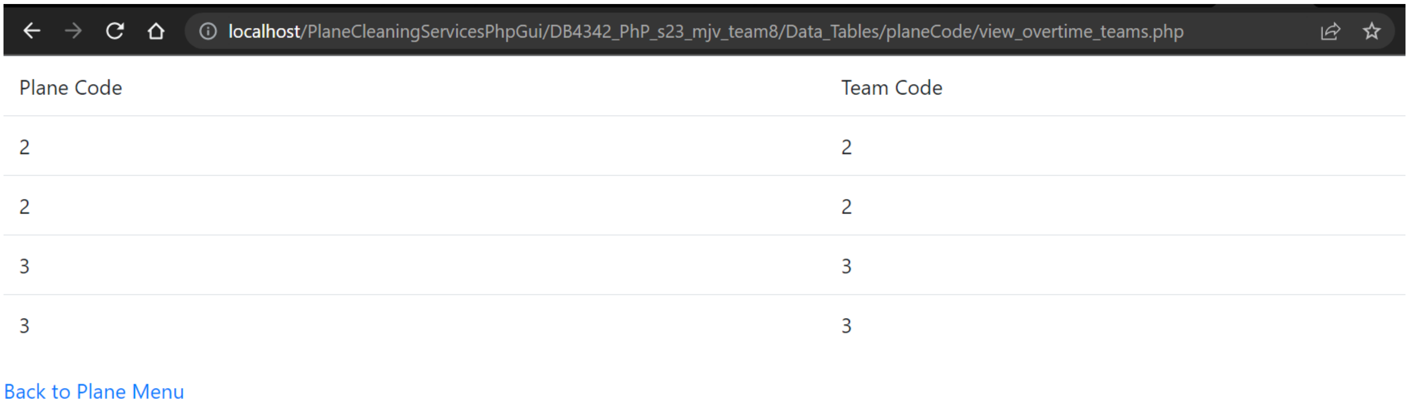


```
 MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT * FROM service_airport WHERE PLairport_code=21;
+----------------+-------------------+
| PLairport_code | SERservice_number |
+----------------+-------------------+
| 21             |               101 |
+----------------+-------------------+
1 row in set (0.0083 sec)
```

Figure 87 Report 8 SQL Query

9. From R18, below you can see what group took longer than expected to clean a plane (over the maximum time).

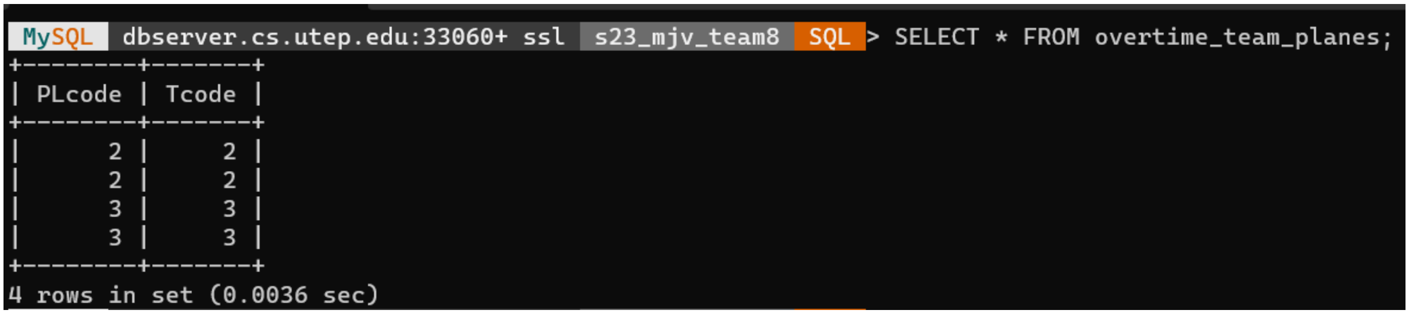
-Plane>> -View Overtime Teams



localhost/PlaneCleaningServicesPhpGui/DB4342_PhP_s23_mjv_team8/Data_Tables/planeCode/view_overtime_teams.php

Plane Code	Team Code
2	2
2	2
3	3
3	3

Figure 88 Report 9 GUI



```
MySQL  dbserver.cs.utep.edu:33060+ ssl  s23_mjv_team8  SQL > SELECT * FROM overtime_team_planes;
+--------+-------+
| PLcode | Tcode |
+--------+-------+
|      2 |     2 |
|      2 |     2 |
|      3 |     3 |
|      3 |     3 |
+--------+-------+
4 rows in set (0.0036 sec)
```

Figure 89 Report 9 SQL Query

12. REQUIREMENT TRACING

Requirement	Addressed By	Created By
R1. The system should keep track of the personnel information: a unique social security number, name (first name and last name), birth date, gender, phone number, and salary per hour.	Table – Figure 30	Dario Vazquez
R2. The system should not allow employees to be part of more than one group at a time.	Constraint – Figure 31	Dario Vazquez
R3. The system shall generate a report to keep track of the plane's location via airport code and gate.	Report – Figure 71, 72	Dario Vazquez
R4. The system shall generate a report containing the planes that were cleaned for a specific period.	Report Figure 73,74	Dario Vazquez
R5. The system should only allow one employee to supervise a group at a given time.	Constraint – Figure 34	Dario Vazquez
R6. The system should allow the general manager to generate a list of the services provided per plane, airport, group for a specified period.	Report- Figures 63	Maria Vargas
R7. The system should keep track of the services: service number, plane code, airport code, date, time it took to fulfill the service, and group code.	Table-Figure 22	Maria Vargas
R8. The system should allow the general manager to generate information to know the total number of employees and the employees that are available at a specific moment.	Report-Figure 78	Maria Vargas
R9. The system should keep track of each group, its status, supervisor, location and code.	Report-Figure 26	Maria Vargas
R10. The system should allow us to update the group membership of the worker.	Report- Figure 68	Maria Vargas
R11 When assigned to a group the status of a worker is changed to in service and the status of an in service worker cannot be changed to unavailable.	Trigger - 67	Maria Vargas
R12 The system will allow update of which airport and plane is assigned to each group.	Query – Figure 41	Taposh Sarker
R13. The system should generate a report of which group cleaned which plane at which airport at which specific time.	Report – Figure 79,80	Taposh Sarker
R14. The system shall keep records of the exact location (airport, plane, gate) of a group while in-service.	Query – Figure 43	Taposh Sarker
R15. The system shall keep track of which groups are in-service at a particular point of time.	Query Figure 44	Taposh Sarker
R16. The system will make sure that a group is assigned to only one plane at a time.	Constraint – Figure 45	Taposh Sarker
R17. The system will allow the general manager to	Report – Figure 84	Taposh Sarker

generate a list of service provided by airport.		
R18. The system will keep track of the planes' information: category (A, B, C), expected time for it to be cleaned (min and max time), a unique code and compose location (airport code, gate).	Table – Figure 47 Report - Figure 87	Christopher Ibarra
R19. The system will generate a report of the groups that took longer than expected to clean a plane.	Report – Figure 86	Christopher Ibarra
R20. The system will make sure that each group has 7 members for it to be considered operational.	Trigger- Figure 70	Christopher Ibarra
R21. The system will make sure that each group has a supervisor.	Table- Figure 26	Christopher Ibarra

13. GRAPHICAL USER INTERFACE

The following screenshots show the User GUI for the project. They start at the user login and are taken to a menu where they can choose how they would like to interact with the database. This menu has an option for each table, where you can insert, update, and delete entries in that specific table. Each table menu also holds options for specific reports or functional requirements as outlined in the scope.

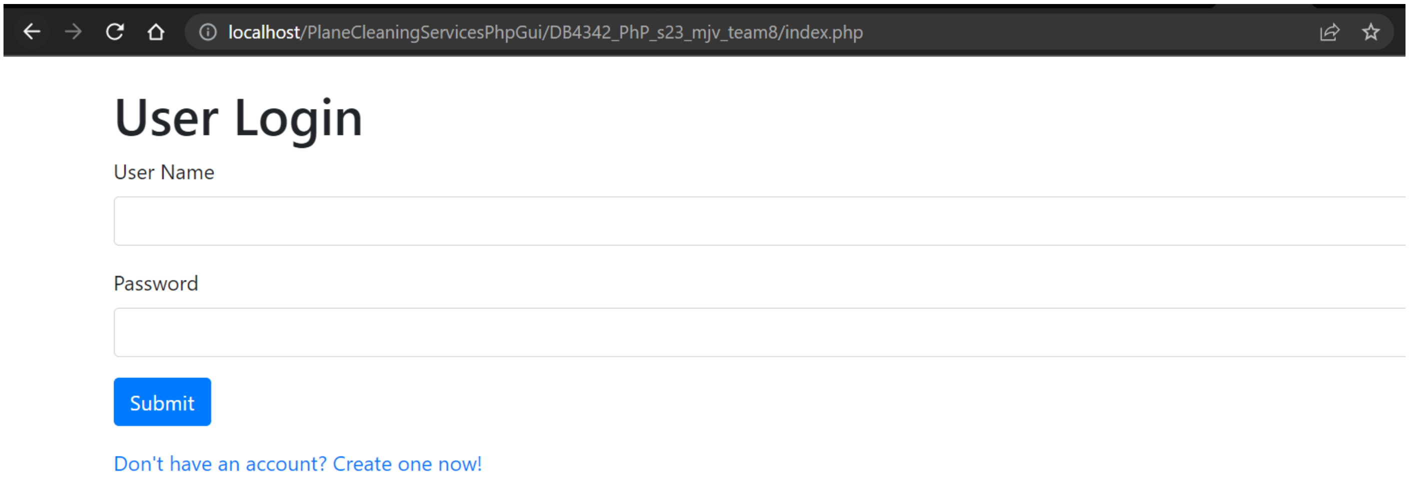
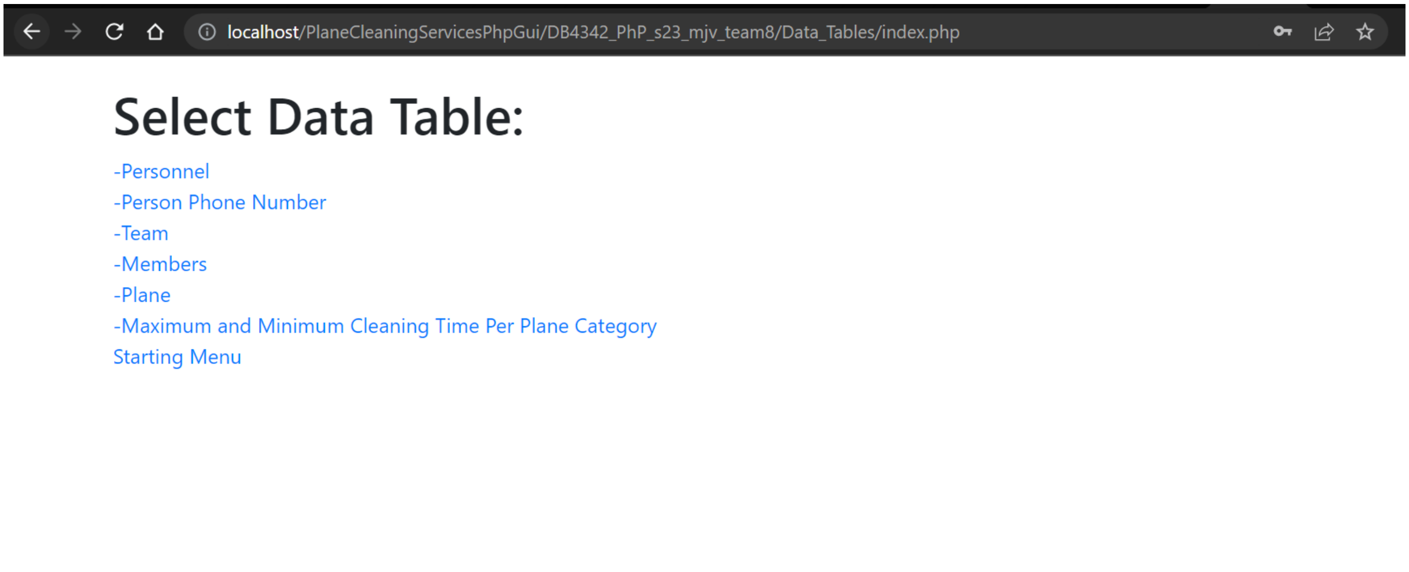

localhost/PlaneCleaningServicesPhpGui/DB4342_PhP_s23_mjv_team8/index.php
User Login
User Name
Password
Submit
Don't have an account? Create one now!

Figure 90 User Login Screen

For example, after logging in, we can click on “-Personnel”



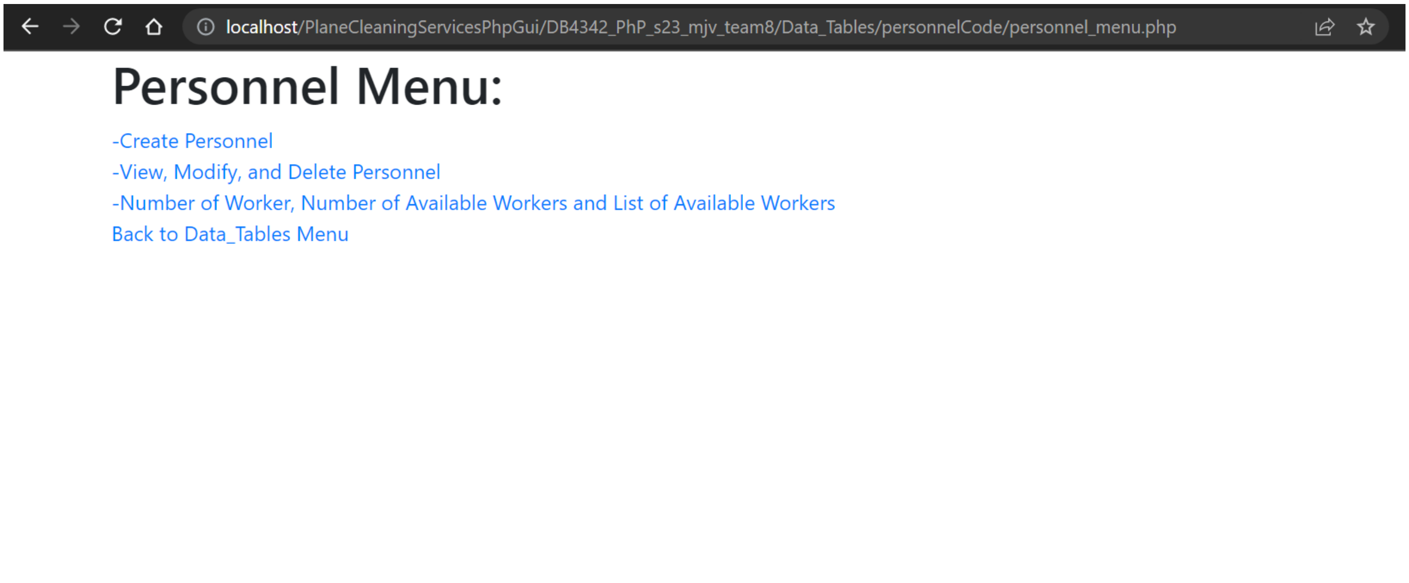
localhost/PlaneCleaningServicesPhpGui/DB4342_PhP_s23_mjv_team8/Data_Tables/index.php

Select Data Table:

-Personnel
-Person Phone Number
-Team
-Members
-Plane
-Maximum and Minimum Cleaning Time Per Plane Category
Starting Menu

Figure 91 Main Menu

This takes us to modifying the table in the database or looking at a report.



localhost/PlaneCleaningServicesPhpGui/DB4342_PhP_s23_mjv_team8/Data_Tables/personnelCode/personnel_menu.php

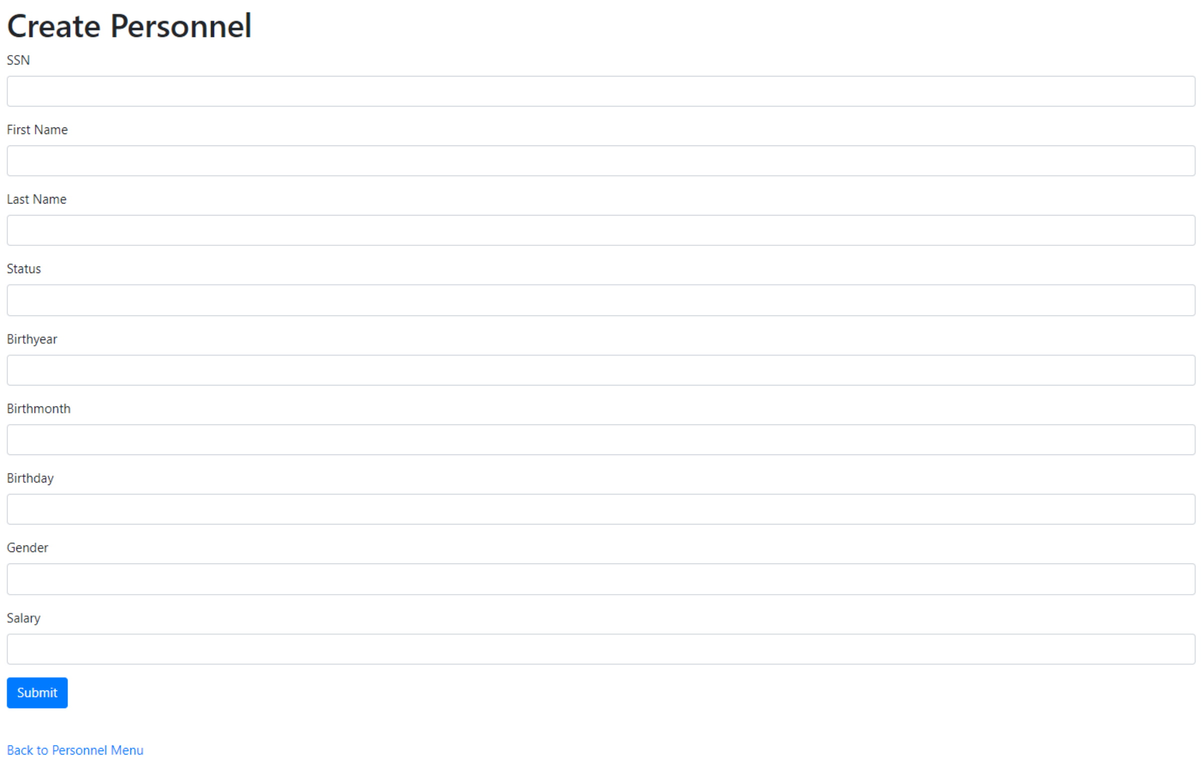
Personnel Menu:

-Create Personnel
-View, Modify, and Delete Personnel
-Number of Worker, Number of Available Workers and List of Available Workers
Back to Data_Tables Menu

Figure 92 Personnel Menu

We can create a record with “-Create Personnel” or view the PERSONNEL table and update or delete a record by clicking on “-View, Modify, and Delete Personnel” and choosing either “Update” or “Delete” on the right as shown below.

-Create Personnel



Create Personnel

SSN

First Name

Last Name

Status

Birthyear

Birthmonth

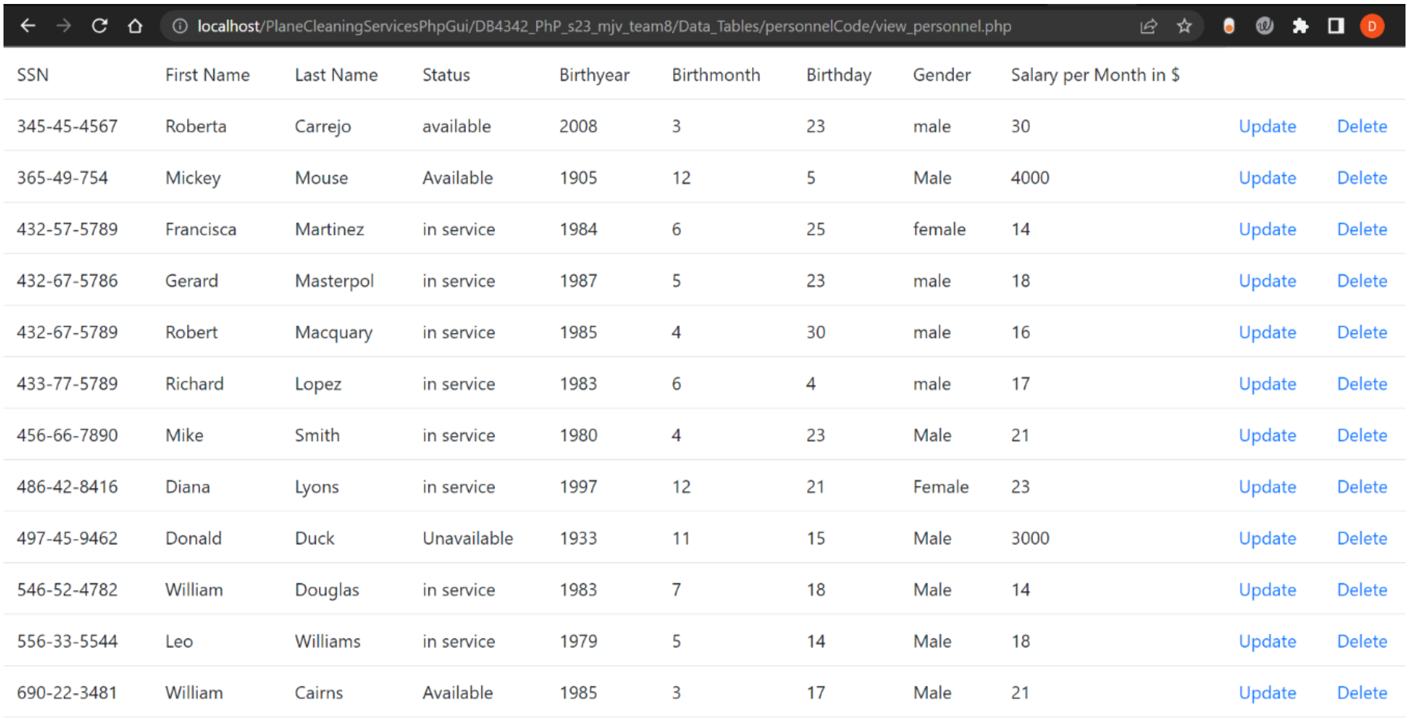
Birthday

Gender

Salary

Figure 93 Create Personnel Link

-View, Modify, and Delete Personnel



localhost/PlaneCleaningServicesPhpGui/DB4342_PhP_s23_mjv_team8/Data_Tables/personnelCode/view_personnel.php

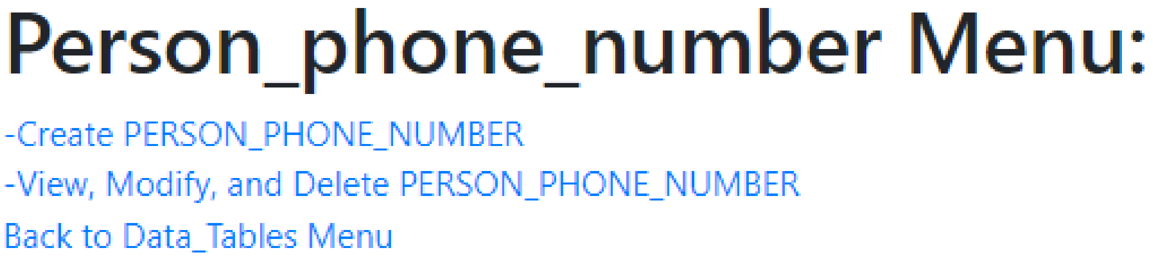
SSN	First Name	Last Name	Status	Birthyear	Birthmonth	Birthday	Gender	Salary per Month in $		
345-45-4567	Roberta	Carrejo	available	2008	3	23	male	30	Update	Delete
365-49-754	Mickey	Mouse	Available	1905	12	5	Male	4000	Update	Delete
432-57-5789	Francisca	Martinez	in service	1984	6	25	female	14	Update	Delete
432-67-5786	Gerard	Masterpol	in service	1987	5	23	male	18	Update	Delete
432-67-5789	Robert	Macquary	in service	1985	4	30	male	16	Update	Delete
433-77-5789	Richard	Lopez	in service	1983	6	4	male	17	Update	Delete
456-66-7890	Mike	Smith	in service	1980	4	23	Male	21	Update	Delete
486-42-8416	Diana	Lyons	in service	1997	12	21	Female	23	Update	Delete
497-45-9462	Donald	Duck	Unavailable	1933	11	15	Male	3000	Update	Delete
546-52-4782	William	Douglas	in service	1983	7	18	Male	14	Update	Delete
556-33-5544	Leo	Williams	in service	1979	5	14	Male	18	Update	Delete
690-22-3481	William	Cairns	Available	1985	3	17	Male	21	Update	Delete

Figure 94 View Personnel Link

The other tables have the same functionalities as well as specific reports for the functional requirements.

Here are the rest of the menus for each table:

-Person Phone Number



Person_phone_number Menu:

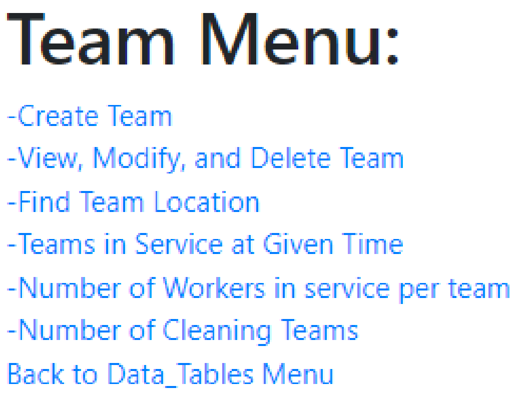
-Create PERSON_PHONE_NUMBER

-View, Modify, and Delete PERSON_PHONE_NUMBER

Back to Data_Tables Menu

Figure 95 Person_phone_number Menu

-Team



Team Menu:

-Create Team

-View, Modify, and Delete Team

-Find Team Location

-Teams in Service at Given Time

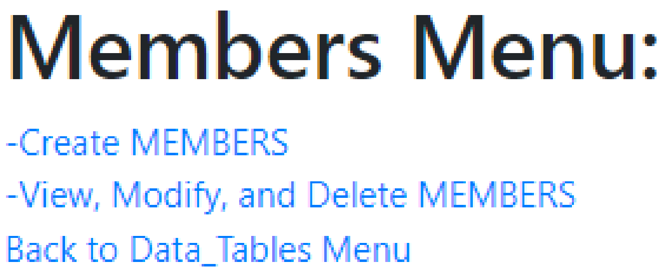
-Number of Workers in service per team

-Number of Cleaning Teams

Back to Data_Tables Menu

Figure 96 Team Menu

-Members



Members Menu:

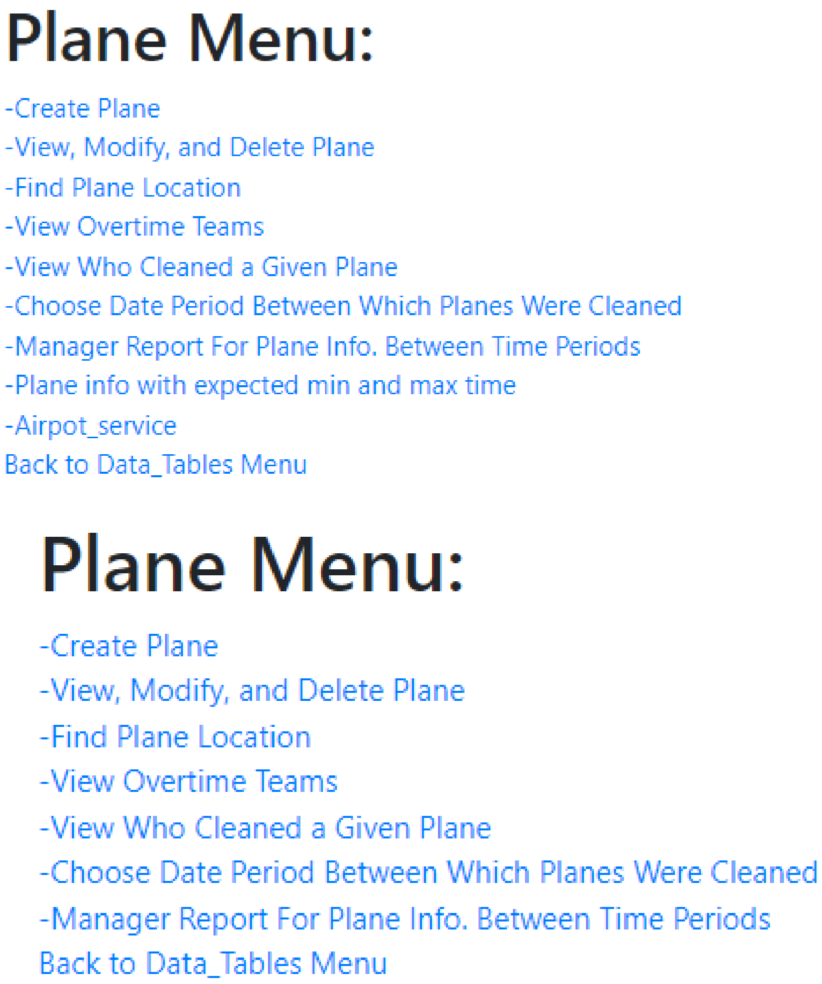
-Create MEMBERS

-View, Modify, and Delete MEMBERS

Back to Data_Tables Menu

Figure 97 Members Menu

-Plane



Plane Menu:

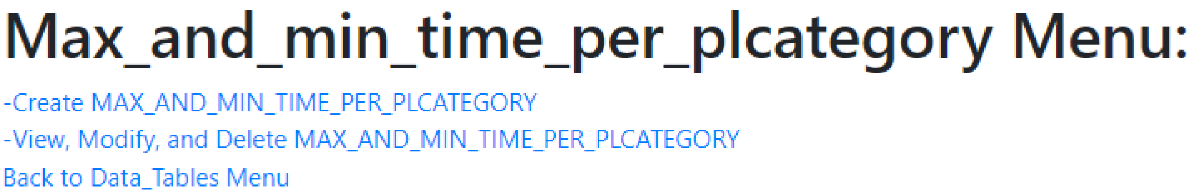
-Create Plane
-View, Modify, and Delete Plane
-Find Plane Location
-View Overtime Teams
-View Who Cleaned a Given Plane
-Choose Date Period Between Which Planes Were Cleaned
-Manager Report For Plane Info. Between Time Periods
-Plane info with expected min and max time
-Airpot_service
Back to Data_Tables Menu

Plane Menu:

-Create Plane
-View, Modify, and Delete Plane
-Find Plane Location
-View Overtime Teams
-View Who Cleaned a Given Plane
-Choose Date Period Between Which Planes Were Cleaned
-Manager Report For Plane Info. Between Time Periods
Back to Data_Tables Menu

Figure 98 Plane Menu

-Maximum and Minimum Cleaning Time Per Plane Category



Max_and_min_time_per_plcategory Menu:

-Create MAX_AND_MIN_TIME_PER_PLCATEGORY
-View, Modify, and Delete MAX_AND_MIN_TIME_PER_PLCATEGORY
Back to Data_Tables Menu

Figure 99 Max_and_min_time_per_plcategory Menu

The specific report descriptions, outputs and types of inputs can be referenced in section 11.

14. IIS WEB SERVER AND GUI

The interface files reside in the utep servers per the instructions. The GUI can be accessed with the following credentials:

- **id: user**

- **pwd:user1**

11. REFERENCES

- Villanueva-Rosales, N.; G. Jimenez, M. (2023). Lecture 2 - Project Requirements and Assumptions [PowerPointslides].
- Villanueva-Rosales, N.; G. Jimenez, M. (2023). Lecture 3– Database Management - Datamodels & E/R Diagrams [PowerPointslides].
- Villanueva-Rosales, N.; G. Jimenez, M. (2023). Lecture 5 & 6 - Database Management E/R diagram to Relational model [PowerPointslides].
- Villanueva-Rosales, N. (2023). Lecture 5 – Database Management - Normalization [PowerPointslides].

APPENDIX A. ATTRIBUTION INFORMATION

The contributions of the team members for this assignment are as follows:

1. Dario Vazquez
   1.1. Contributed to the refinement of the scope.
   1.2. Created the rough draft for the E/R diagrams
   1.3. Contributed to the refinement of the E/R diagram
   1.4. Contributed to the refinement of requirements
   1.5. Contributed to the refinement of the relational model.
   1.6. Corrected the ER diagram from Assignment 1.
   1.7. Populated some tables in SQL
   1.8. Completed SQL subqueries for R1-R5
   1.9. Created user in GUI interface.
   1.10. Tested Team interface
   1.11. Added an appropriate header to all PHP files
   1.12. Added R18 to GUI
   1.13. Corrected query for R14
   1.14. Implemented R14 in PHP for interface
   1.15. Corrected query for R13
   1.16. Implemented R13 in PHP for interface
   1.17. Corrected query for R15
   1.18. Implemented R15 for interface
   1.19. Corrected query for R18
   1.20. Implemented R18 for interface
   1.21. Implemented R1 procedure in PHP for interface
   1.22. Implemented R3 procedure in PHP for interface
   1.23. Implemented R4 procedure in PHP for interface
   1.24. Contributed to sections 11 and 13 of report
   1.25. Created overtime_teams view

2. Maria Vargas
   2.1. Contributed to the first draft and refinement of the scope.
   2.2. Created and refined requirements R6 to R11 helped with assumptions.
   2.3. Contributed to the integration of the E/R diagram and help match it to the scope.
   2.4. Contributed to the refinement of the relational model.
   2.5. Normalized the relational model.
   2.6. Created tables: Members, Plane and Personnel in SQL and populated some of them.
   2.7. Completed subqueries in SQL for requirements R6 to R11.
   2.8. Created user in GUI interface.
   2.9. Created the procedures, views and triggers for all the requirements of this project. Some were used in PHP some were not.
   2.10. Revise two of the tables in GUI team and personnel.
   2.11. Implemented two reports in PHP

2.12. Took the screenshots of many of the sql views, procedures and triggers.

2.13. Did most of the presentation.

2.14. Changed the colors of the letters in the Relational and Normalized diagrams to be visible.

3. Taposh Sarker

3.1. Contributed to the first draft of the scope.

3.2. Created and refined requirements R8 and R12.

3.3. Created the E/R diagrams for the Plame entity (Figure 1, in green).

3.4. Contributed to the integration of the E/R diagram.

3.5. Provided feedback for their relational model.

3.6. Help with normalization of the relational model.

3.7. Changed ER diagram after changes in scope.

3.8. Completed subqueries in SQL for R8 – R12.

3.9. Helped in development environment setup

3.10. Helped in database backup and experiment

3.11. Helped in code version management (git) and deployment

3.12. Helped in overall GUI development.

4. Christopher Ibarra

4.1. Contributed to creating the un-normalized and normalized relational schemas diagrams.

4.2. Helped normalize the relational schema.

4.3. Created the tables TEAM, PERSON_PHONE_NUMBER, and MAX_AND_MIN_TIME_PER_PLCATEGORY

4.4. Helped populate the tables PERSONNEL, MEMBERS, PERSON_PHONE_NUMBER, and MAX_AND_MIN_TIME_PER_PLCATEGORY

4.5. Implemented the queries for the functional requirements R16 to R20.

4.6. Created some assumptions.

4.7. Contributed to the E/R diagram.

4.8. Help with the refinement of the scope.

4.9. Created some of the procedures, triggers and views.

4.10. Took snapshots for the reports and GUI sections.

4.11. Guided with the GUI interface.

TEAM JOURNAL

Dario: I agree with the contribution of my teammates stated in this journal entry.

Maria: I agree with the contribution of my teammates stated in this journal entry.

Taposh: I agree with the contribution of my teammates stated in this journal entry.

Christopher: I agree with the contribution of my teammates stated in this journal entry.