RAILWAY RESERVATION DATABASE SYSTEM

J COMPONENT PROJECT REPORT

Submitted by

R. MUKESH KANNA (19BCE2385)

S.K. SHRI CHARAN (19BCE2404)

Under the guidance of
Prof. J. Karthikeyan

in partial fulfillment for the award of the degree of

Bachelor of Technology
In
Computer Science and Engineering

Abstract:

This project titled 'Railway Reservation System' is the automation of the whole process of managing the Railway Information. The purpose of Railway Reservation System is to manage the details of Railway bookings and cancellation, trains, passengers and train staffs. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The

purpose of the project is to build a database and manage information regarding the ticket booking and train services. The objective of the proposed project is to increase the efficiency of managing the railway information and to electronically handle the airline related record to enhance the accuracy, flexibility, reliability, to remove the human's error and provide a robust structure that can handle any amount of data. Some of its features are searching facilities based on various factors, editing, adding and updating the records which results in proper resource management of Railway. This system can lead to error-free, secure, reliable and fast management of records. Their valuable data can be stored for a longer period with easy accessing and manipulation of the same. This project is developed using Python as programming language and MySQL as an open source relational database management system.

Introduction:

Database and database systems have become an essential component of everyday life in modern society. In the course of a day, most of us encounter several activities that involve some interaction with the database. For example, if we go to the bank to deposit or withdraw funds or if we make a Hotel or Railway Reservation, chances are that our activities will involve someone accessing a database. The above interactions are examples of what we may call traditional database applications, where most of the application that is stored and accessed is either textual or numeric. In our project we will concentrate on this aspect of computer application. It is obvious that everything that is sustainable would have to go through advancement. In science and technology, the desire for improvement is a constant subject which triggers advancements. This is visible in every industry and the railway industry is not an exemption. The railway industry is a very particular system. Railways provide a service, which is to transport a passenger between two cities at an agreed price. Railways also exhibit very particular economics that, over time, have motivated specific management concepts, tools and practices. Railway information system used to be standalone systems. Each railway has its own system, disconnected from other railways or ticket agents, and usable only by a designated number of railway employees. Today, railway travel information is linked, stored, and retrieved by a network of Computer Reservations Systems (CRS), accessible by multiple railways.

Methodology:

1. Creation of entities:

Passenger

- Ticket

- Train
- Station

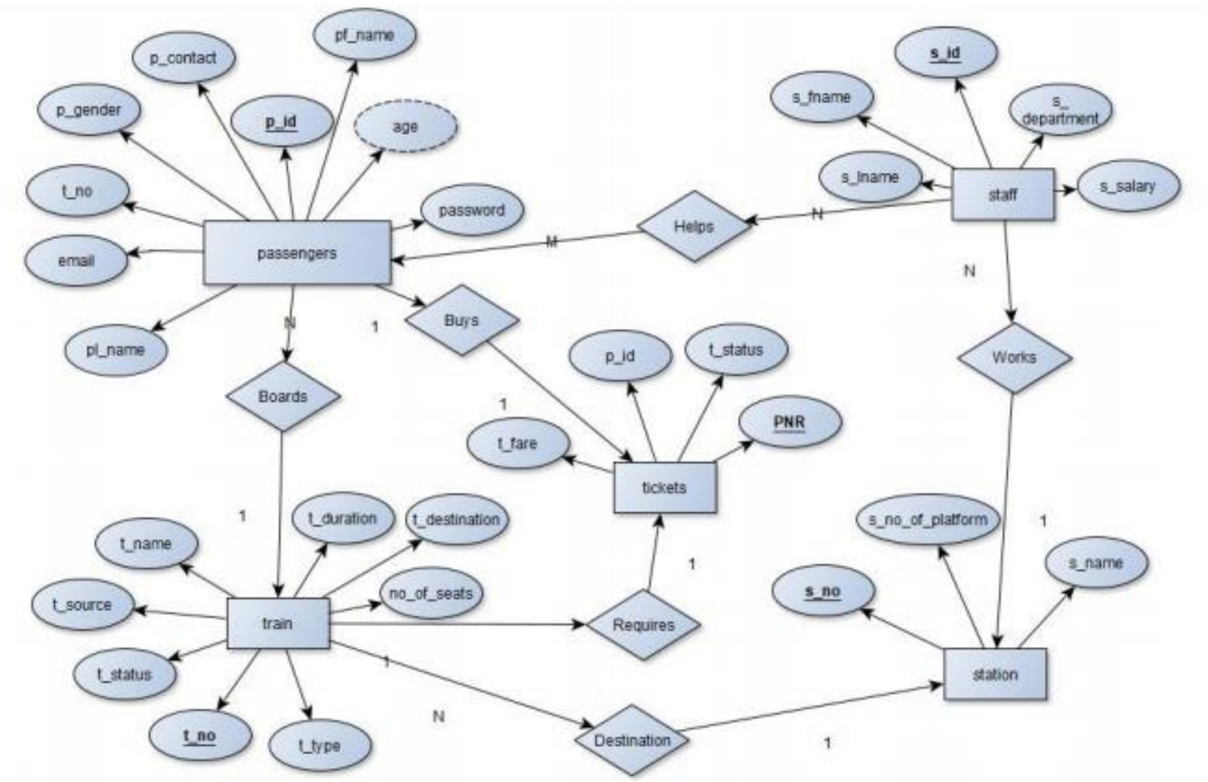
2. Attributes of each entity:

- Passenger – Name (First and Last), Gender, Age, ID, Contact, Email, Password, Train number
- Train-Train duration, Train name, Train source, Train status, Train number, Number of seats, Train type, Train destination
- Station-Name, Number, Number of platform
- Ticket-Passenger ID, Train status, PNR, Ticket fare

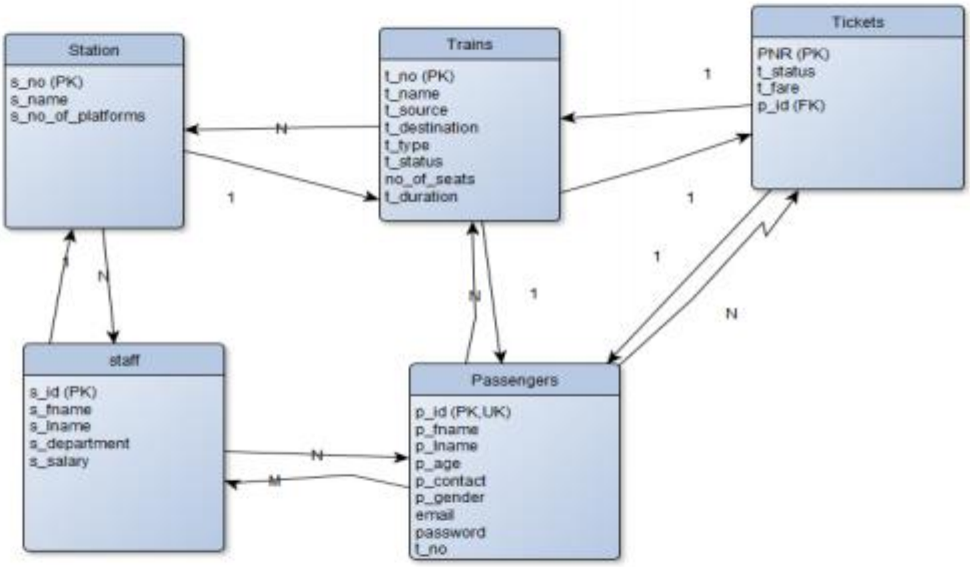
3. Relationships between entities:

- Passenger boards in train and buys tickets.
- Train stops in station and requires tickets to board.

ER Diagram:

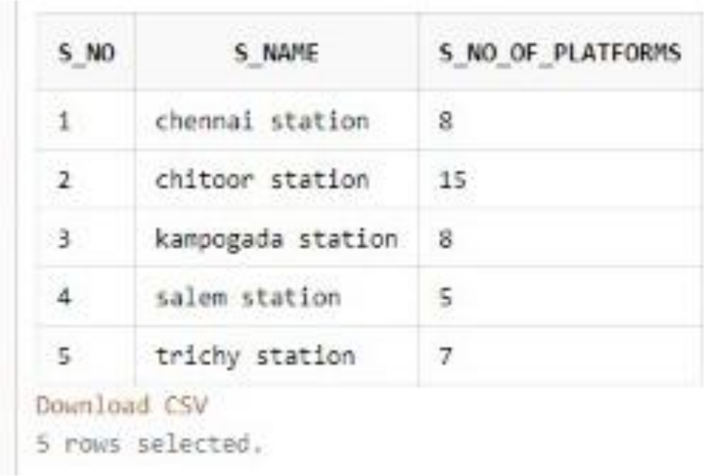


Relational Schema



CREATION OF RELATIONAL TABLE

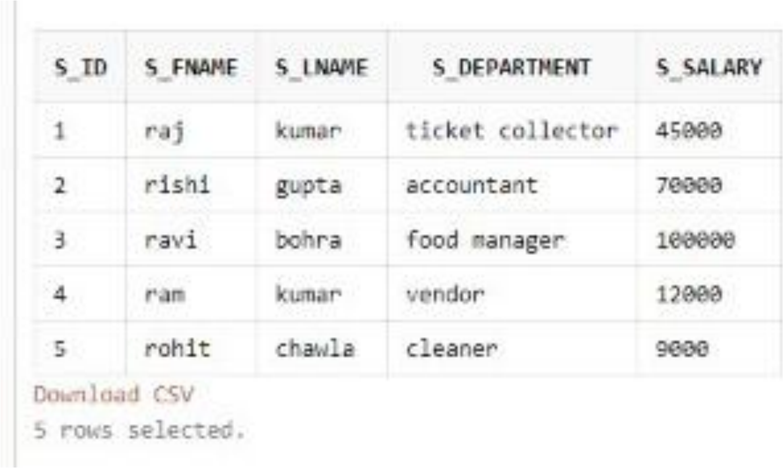
- create table station (s_no number primary key, s_name varchar(20),s_no_of_platforms number);
- create table staff(s_id number primary key,s_fname varchar(20),s_lname varchar(30),s_department varchar(40),s_salary number);
- create table trains(t_no number primary key,t_name varchar(20),t_source varchar(30),t_destination varchar(40),t_type varchar(30),t_status varchar(40),no_of_seats number,t_duration number);
- create table passengers(p_id number primary key,p_fname varchar(20),p_lname varchar(30),p_age number,p_contact number,p_gender varchar(20),email varchar(100),password number,t_no number references trains(t_no));
- create table tickets(PNR number unique,t_status varchar(20),t_fare number,p_id number references passengers(p_id));



S_NO	S_NAME	S_NO_OF_PLATFORMS
1	chennai station	8
2	chitoor station	15
3	kampogada station	8
4	salem station	5
5	trichy station	7

Download CSV

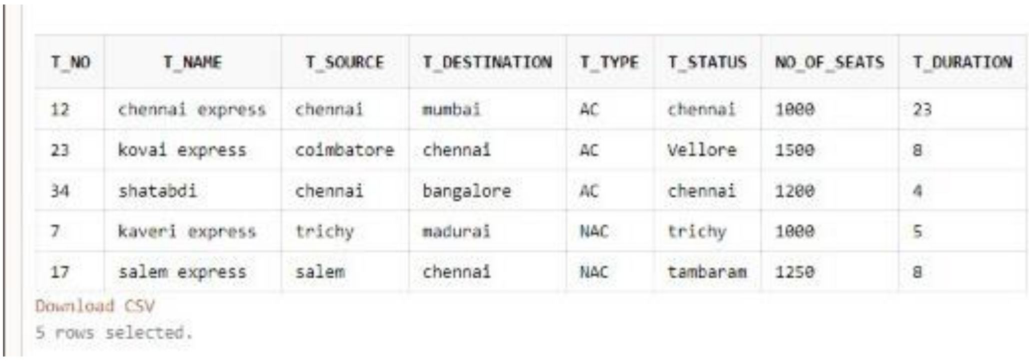
5 rows selected.



S_ID	S_FNAME	S_LNAME	S_DEPARTMENT	S_SALARY
1	raj	kumar	ticket collector	45000
2	rishi	gupta	accountant	70000
3	ravi	bohra	food manager	100000
4	ram	kumar	vendor	12000
5	rohit	chawla	cleaner	9000

Download CSV

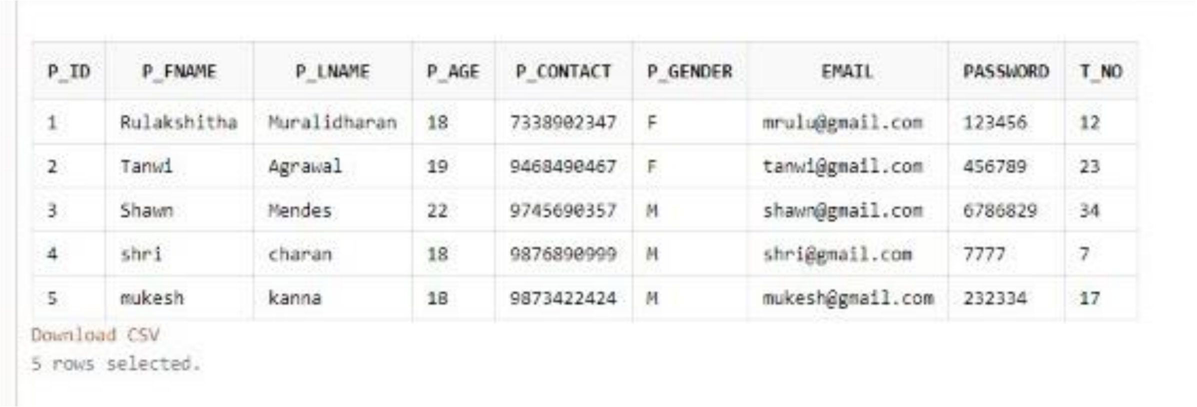
5 rows selected.



T_NO	T_NAME	T_SOURCE	T_DESTINATION	T_TYPE	T_STATUS	NO_OF_SEATS	T_DURATION
12	chennai express	chennai	mumbai	AC	chennai	1000	23
23	kovai express	coimbatore	chennai	AC	Vellore	1500	8
34	shatabdi	chennai	bangalore	AC	chennai	1200	4
7	kaveri express	trichy	madurai	NAC	trichy	1000	5
17	salem express	salem	chennai	NAC	tambaram	1250	8

Download CSV

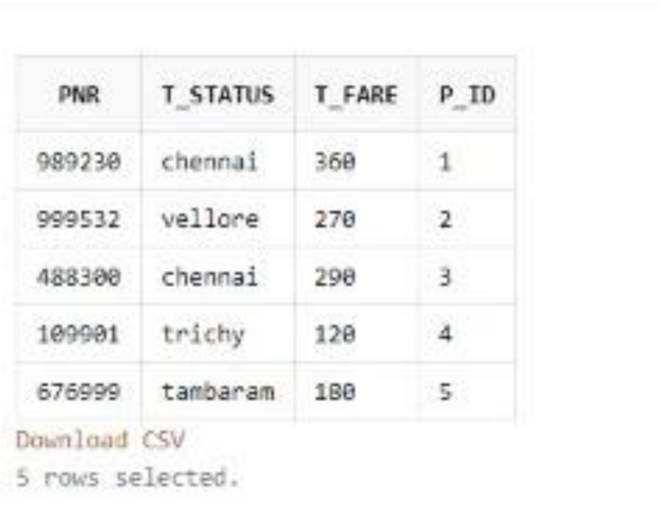
5 rows selected.



P_ID	P_FNAME	P_LNAME	P_AGE	P_CONTACT	P_GENDER	EMAIL	PASSWORD	T_NO
1	Rulakshitha	Muralidharan	18	7338902347	F	mrulu@gmail.com	123456	12
2	Tanwi	Agrawal	19	9468490467	F	tanwi@gmail.com	456789	23
3	Shawn	Mendes	22	9745690357	M	shawn@gmail.com	6786829	34
4	shri	charan	18	9876890999	M	shri@gmail.com	7777	7
5	mukesh	kanna	18	9873422424	M	mukesh@gmail.com	232334	17

Download CSV

5 rows selected.



PNR	T_STATUS	T_FARE	P_ID
989230	chennai	360	1
999532	vellore	270	2
488300	chennai	290	3
109901	trichy	120	4
676999	tambaram	180	5

Download CSV

5 rows selected.

Normalization:

All the tables are normalized upto BCNF.

Functional Dependency:

Table Stations:

s_no -> s_name

s_name -> s_no, s_no_of_platforms

Table Trains:

t_no -> t_name

t_name -> t_no, t_source, t_destination, t_type, t_status, no_of_seats, t_duration

Table Tickets:

PNR -> t_status, t_fare, p_id

p_id -> PNR

Table Staff:

s_id -> s_fname, s_lname

s_fname, s_lname -> s_id, s_department, s_salary

Table Passengers:

p_id -> p_fname, p_lname

pfname, p_lname -> p_id, p_age, p_contact, p_gender, email, password, t_no

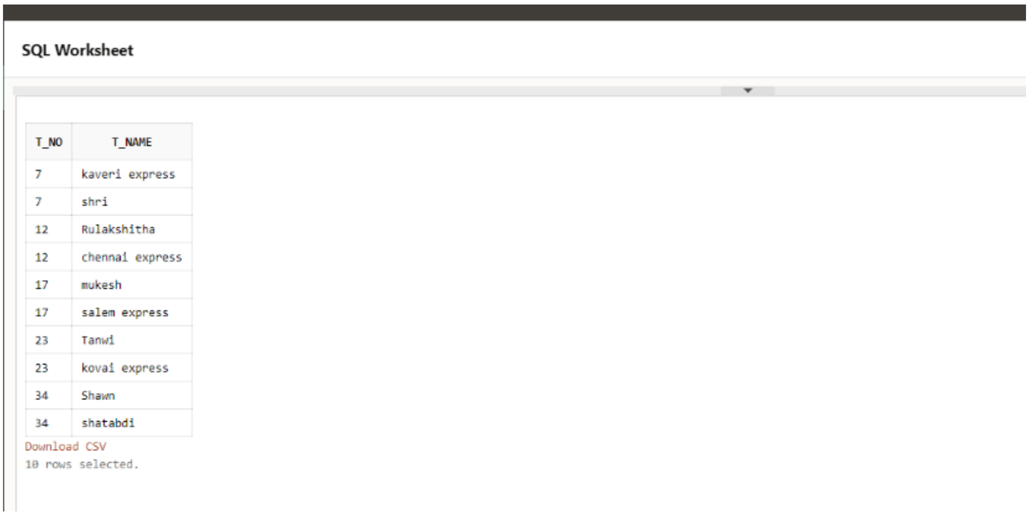
Set Opearators:

a) union

select t_no,t_name from trains

union

select t_no,p_fname from passengers;



T_NO	T_NAME
7	kaveri express
7	shri
12	Rulakshitha
12	chennai express
17	mukesh
17	salem express
23	Tanwi
23	kovai express
34	Shawn
34	shatabdi

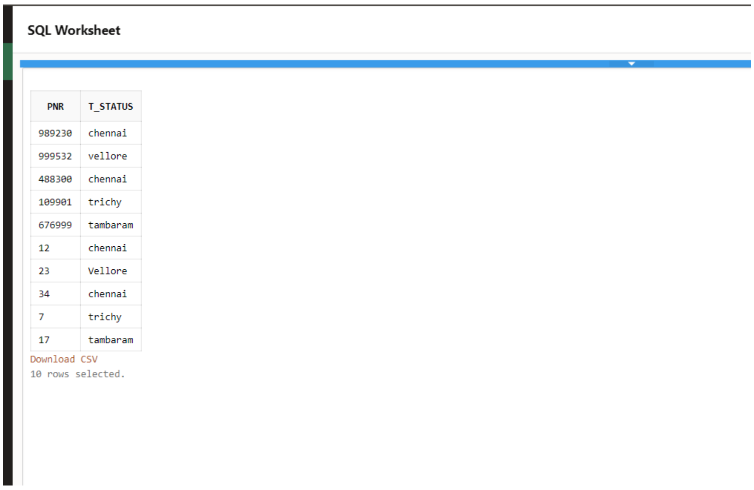
```
10 rows selected.
```

b) union all

select pnr,t_status from tickets

union all

select t_no,t_status from trains;



SQL Worksheet

PNR	T_STATUS
989230	chennai
999532	vellore
488300	chennai
109901	trichy
676999	tambaram
12	chennai
23	Vellore
34	chennai
7	trichy
17	tambaram

Download CSV

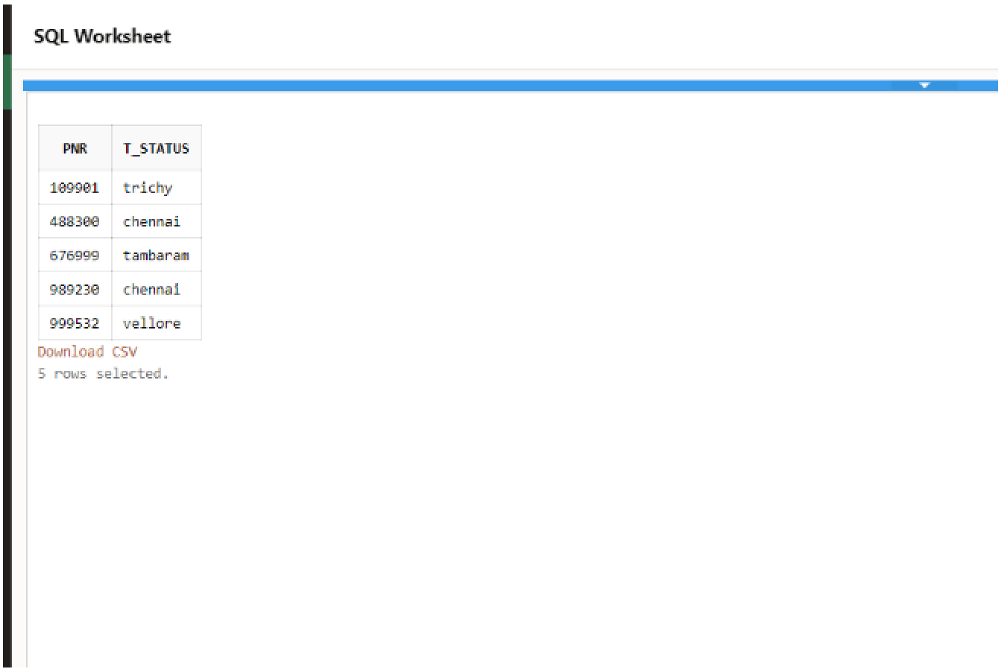
10 rows selected.

c) minus

select pnr,t_status from tickets

minus

select t_no,t_status from trains;



SQL Worksheet

PNR	T_STATUS
109901	trichy
488300	chennai
676999	tambaram
989230	chennai
999532	vellore

Download CSV

5 rows selected.

d) intersect

select t_no,t_name from trains

intersect

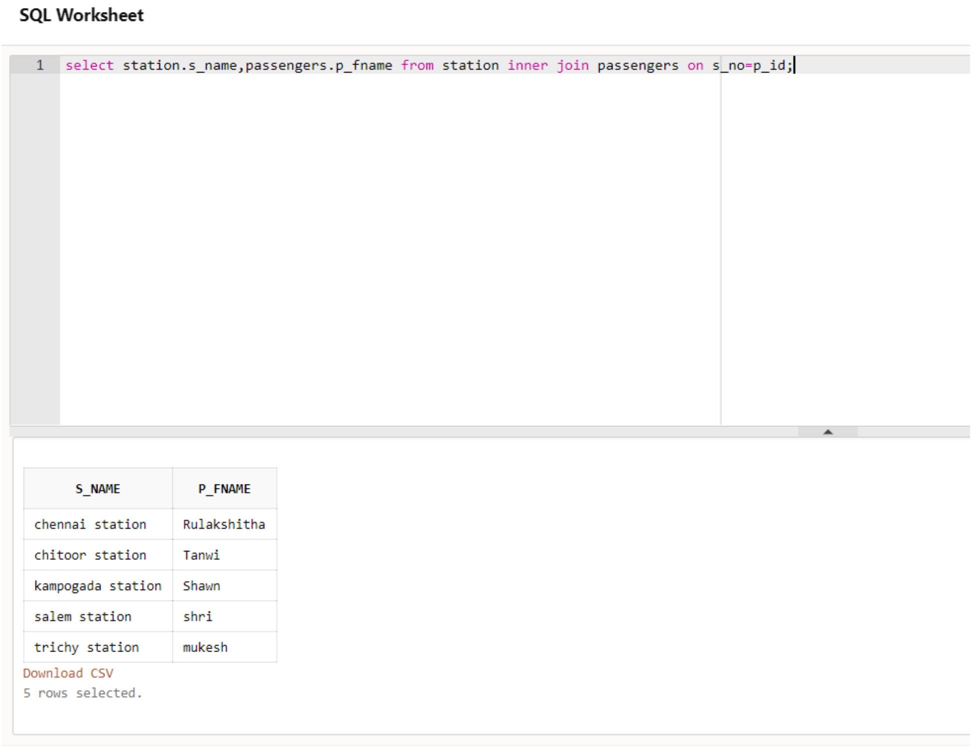
select t_no,p_fname from passengers;

```
no data found
```

Join querries:

a) **List the station name and the passenger's first when passenger Id and the station number are same using inner join.**

select station.s_name,passengers.p_fname from station inner join passengers on s_no=p_id;



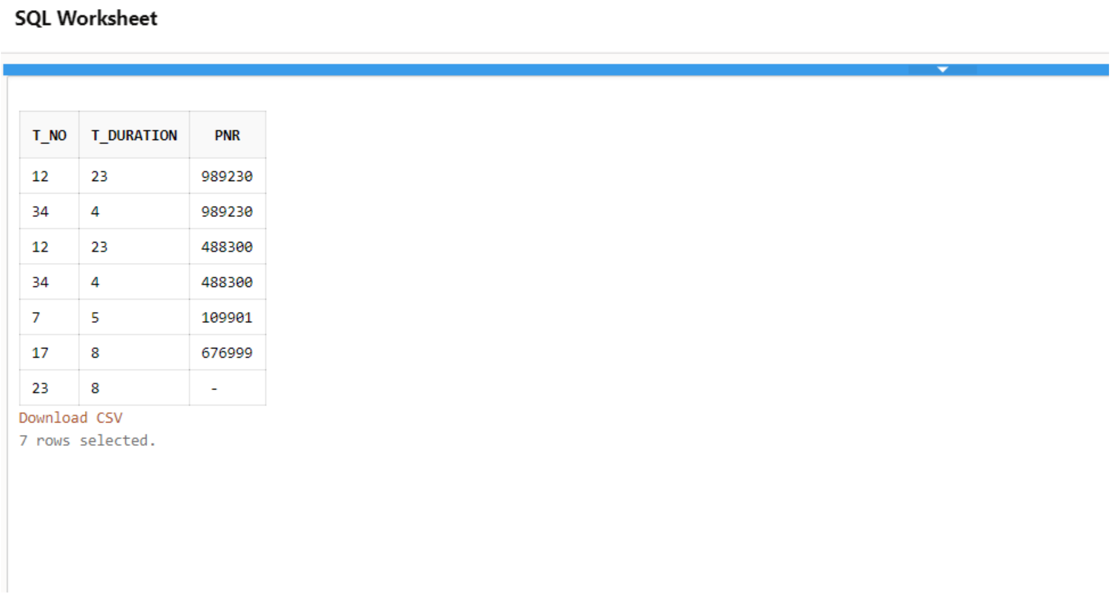
```
select station.s_name,passengers.p_fname from station inner join passengers on s_no=p_id;
```

S_NAME	P_FNAME
chennai station	Rulakshitha
chitoor station	Tanwi
kampogada station	Shawn
salem station	shri
trichy station	mukesh

```
5 rows selected.
```

b) using left outer join list the train number,train duration,pnr by matching the train status

select trains.t_no,trains.t_duration,tickets.pnr from trains left outer join tickets on trains.t_status=tickets.t_status;

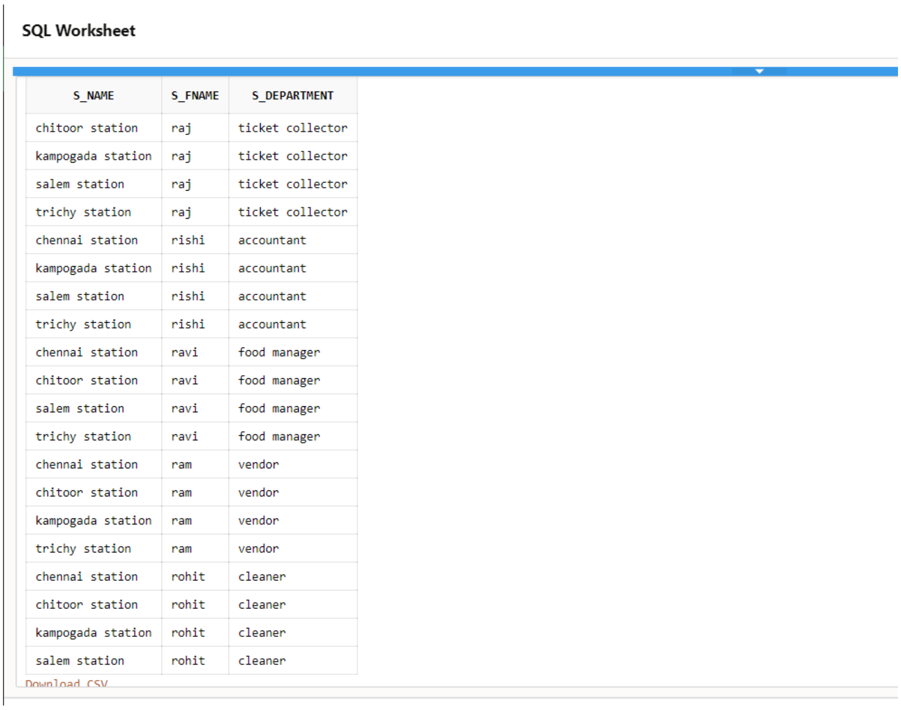


T_NO	T_DURATION	PNR
12	23	989230
34	4	989230
12	23	488300
34	4	488300
7	5	109901
17	8	676999
23	8	-

```
Download CSV
7 rows selected.
```

c) list the staff's first name and department in all stations using right outer join by matching the station number and stff id

select station.s_name,staff.s_fname,staff.s_department from station right outer join staff on station.s_no<>staff.s_id;



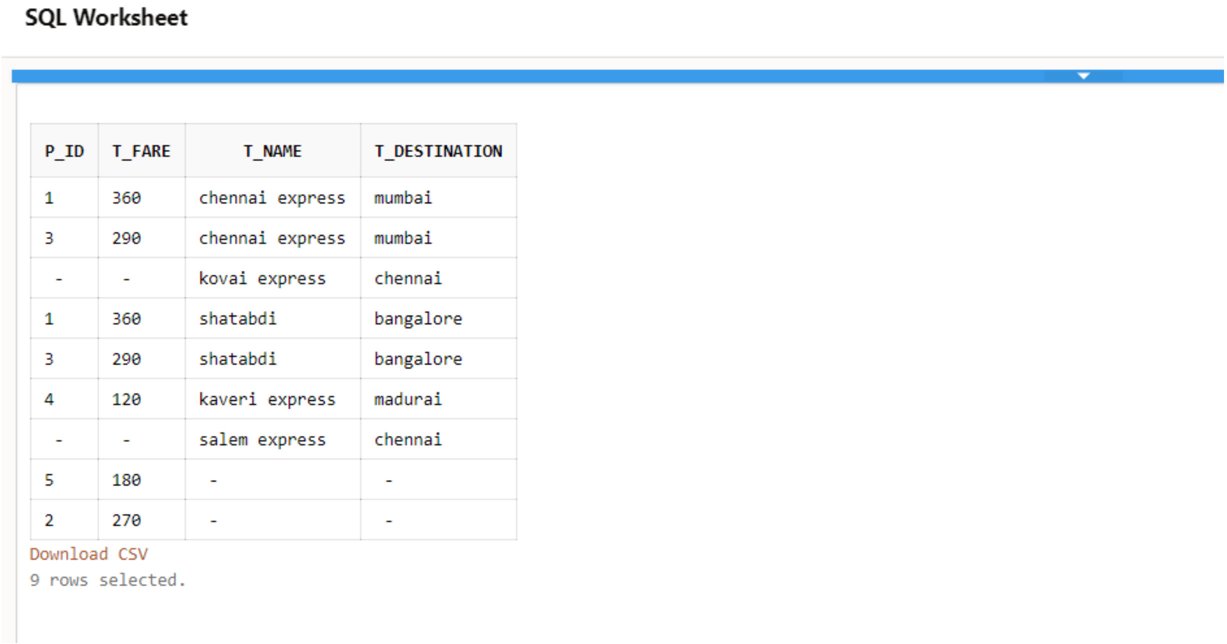
SQL Worksheet

S_NAME	S_FNAME	S_DEPARTMENT
chitoor station	raj	ticket collector
kampogada station	raj	ticket collector
salem station	raj	ticket collector
trichy station	raj	ticket collector
chennai station	rishi	accountant
kampogada station	rishi	accountant
salem station	rishi	accountant
trichy station	rishi	accountant
chennai station	ravi	food manager
chitoor station	ravi	food manager
salem station	ravi	food manager
trichy station	ravi	food manager
chennai station	ram	vendor
chitoor station	ram	vendor
kampogada station	ram	vendor
trichy station	ram	vendor
chennai station	rohit	cleaner
chitoor station	rohit	cleaner
kampogada station	rohit	cleaner
salem station	rohit	cleaner

Download CSV

d) list out platform id,ticket fare,train name,train destination using full outer join by matching the train status and train source.

select tickets.p_id,tickets.t_fare,trains.t_name,trains.t_destination from tickets full outer join trains on tickets.t_status=trains.t_source;



SQL Worksheet

P_ID	T_FARE	T_NAME	T_DESTINATION
1	360	chennai express	mumbai
3	290	chennai express	mumbai
-	-	kovai express	chennai
1	360	shatabdi	bangalore
3	290	shatabdi	bangalore
4	120	kaveri express	madurai
-	-	salem express	chennai
5	180	-	-
2	270	-	-

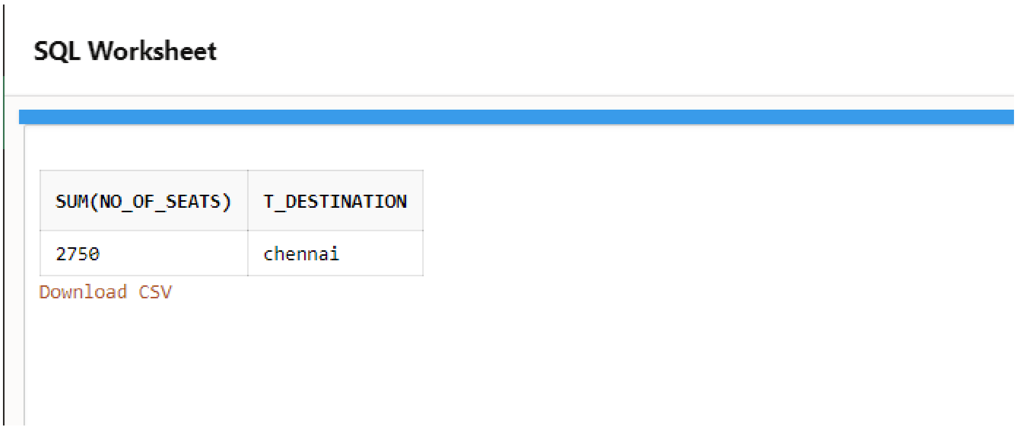
Download CSV

9 rows selected.

Group by:

a) Retrieve the train destination with sum of seats more than 2000.

select sum(no_of_seats),t_destination from trains group by t_destination having sum(no_of_seats)>2000;



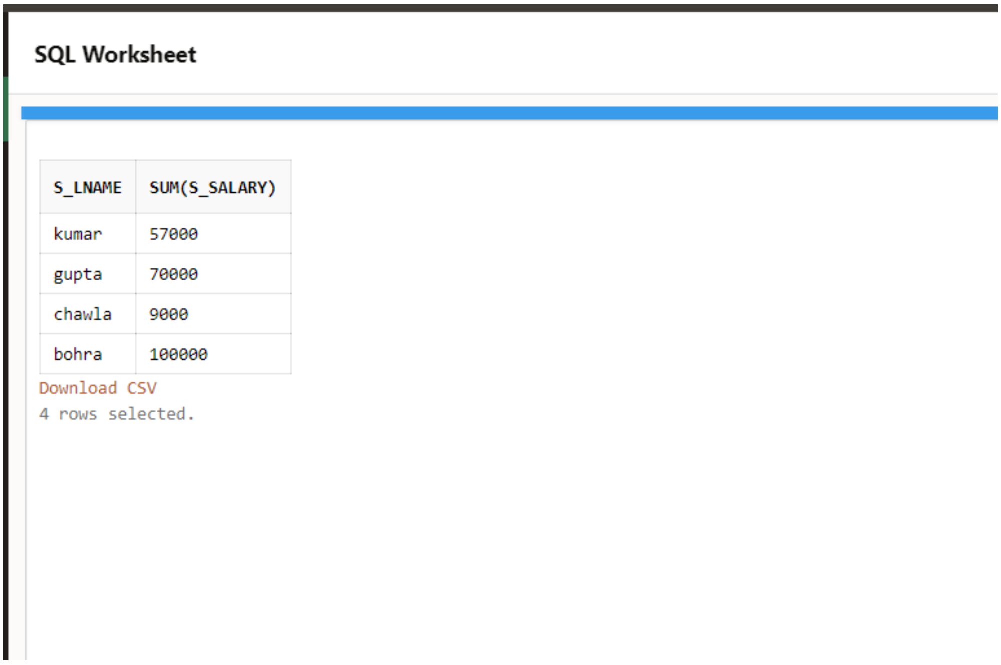
SQL Worksheet

SUM(NO_OF_SEATS)	T_DESTINATION
2750	chennai

Download CSV

b) Retrieve the sum of salaries of staffs by grouping their last names.

select s_lname,sum(s_salary) from staff group by s_lname;



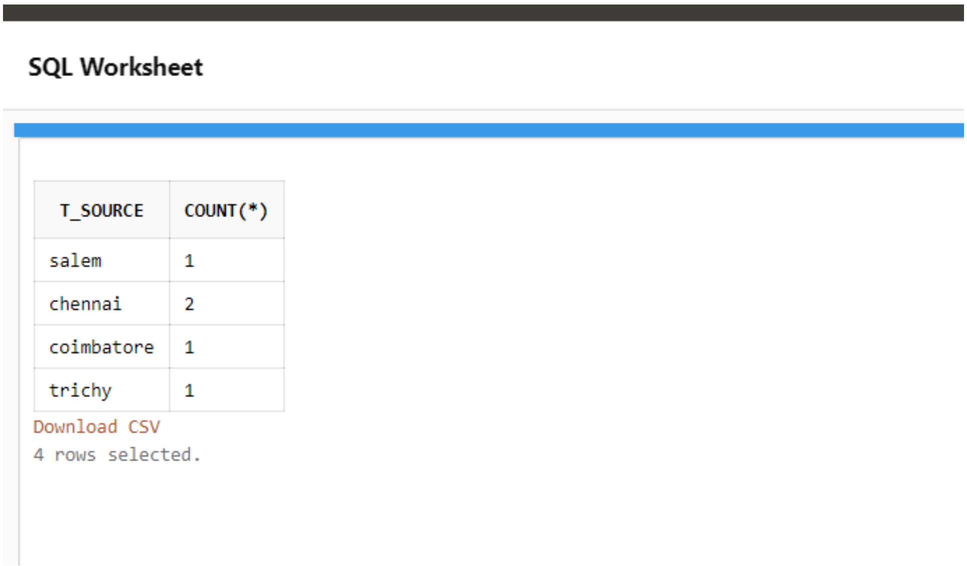
SQL Worksheet

S_LNAME	SUM(S_SALARY)
kumar	57000
gupta	70000
chawla	9000
bohra	100000

Download CSV
4 rows selected.

c) List all the train’s starting point and count the same by grouping them

select t_source,count(*) from trains group by t_source;



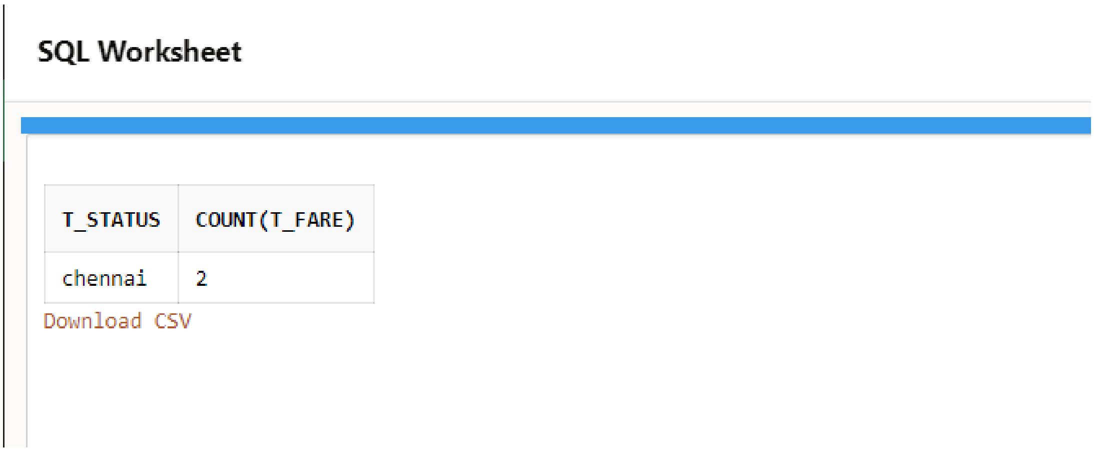
SQL Worksheet

T_SOURCE	COUNT(*)
salem	1
chennai	2
coimbatore	1
trichy	1

```
4 rows selected.
```

d) List all the train’s status with more than one ticket fare

select t_status,count(t_fare) from tickets group by t_status having count(t_fare)>1;n



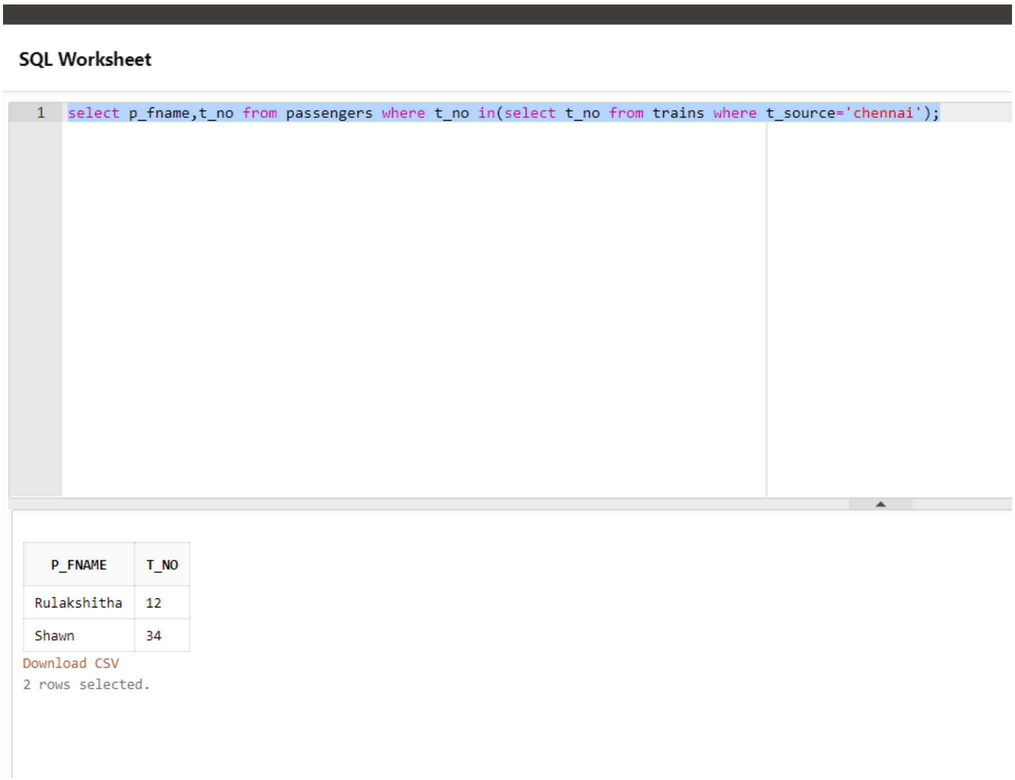
SQL Worksheet

T_STATUS	COUNT(T_FARE)
chennai	2

Nested queries:

a) Retrieve the passenger’s name,train number for those who have the train starting point as chennai

select p_fname,t_no from passengers where t_no in(select t_no from trains where t_source='chennai');



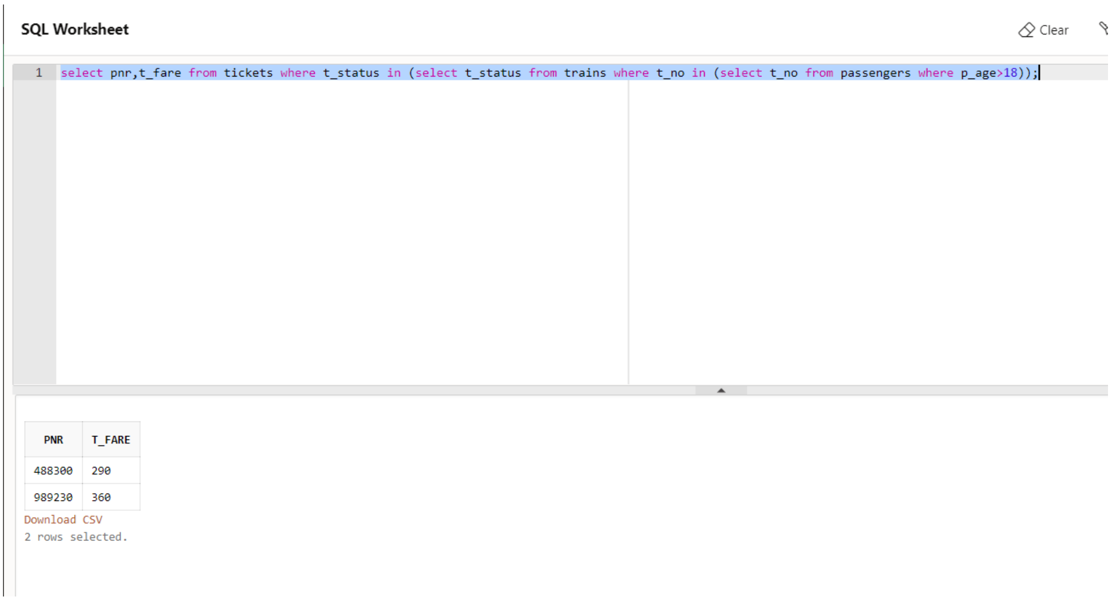
```
select p_fname,t_no from passengers where t_no in(select t_no from trains where t_source='chennai');
```

P_FNAME	T_NO
Rulakshitha	12
Shawn	34

```
2 rows selected.
```

b) List the pnr and ticket fare of passengers who are above the age of 18

select pnr,t_fare from tickets where t_status in (select t_status from trains where t_no in (select t_no from passengers where p_age>18));



```
select pnr,t_fare from tickets where t_status in (select t_status from trains where t_no in (select t_no from passengers where p_age>18));
```

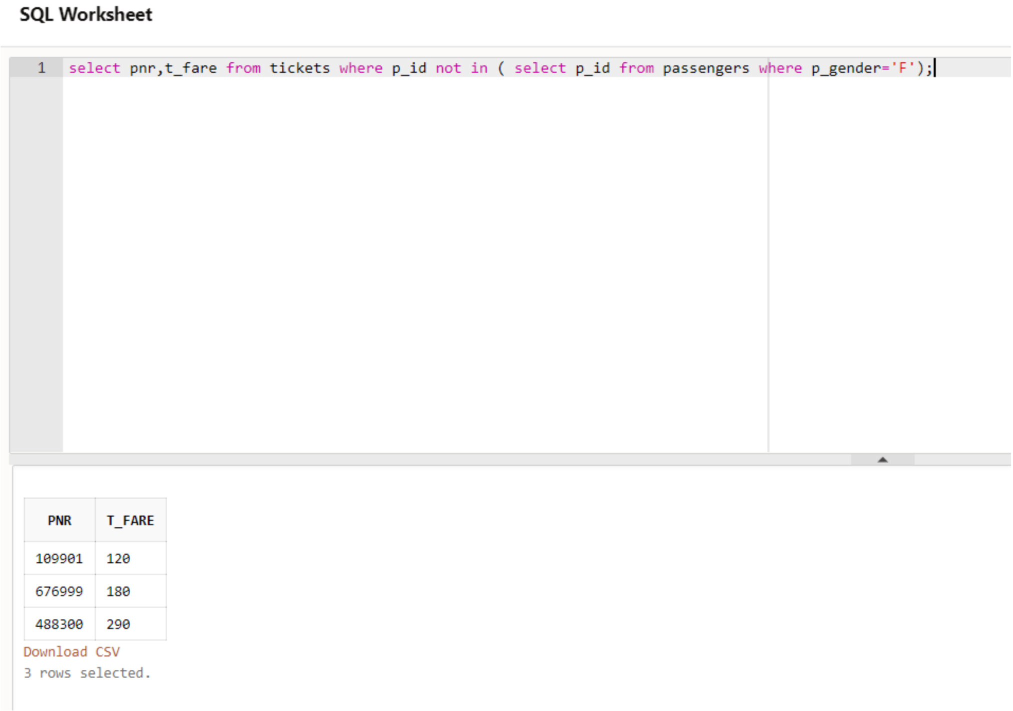
PNR	T_FARE
488300	290
989230	360

Download CSV

2 rows selected.

c) List the pnr ,ticket price of passengers who are not female.

select pnr,t_fare from tickets where p_id not in (select p_id from passengers where p_gender='F');



```
select pnr,t_fare from tickets where p_id not in ( select p_id from passengers where p_gender='F');
```

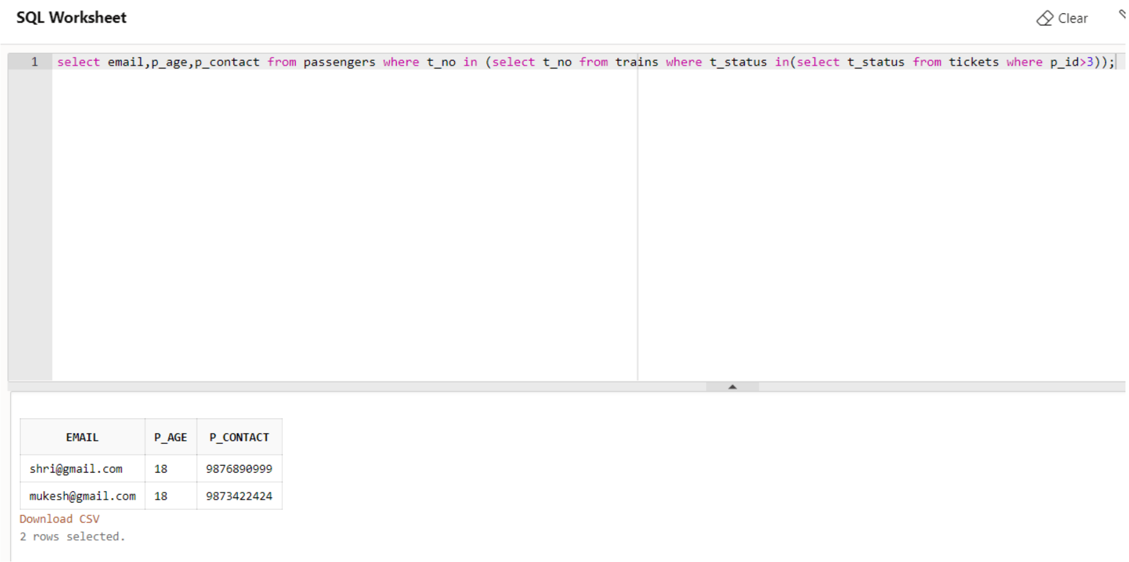
PNR	T_FARE
109901	120
676999	180
488300	290

Download CSV

3 rows selected.

d) Retreive the email ID,age ,contact of the passengers whose passenger ID’s are more than 3.

select email,p_age,p_contact from passengers where t_no in (select t_no from trains where t_status in(select t_status from tickets where p_id>3));



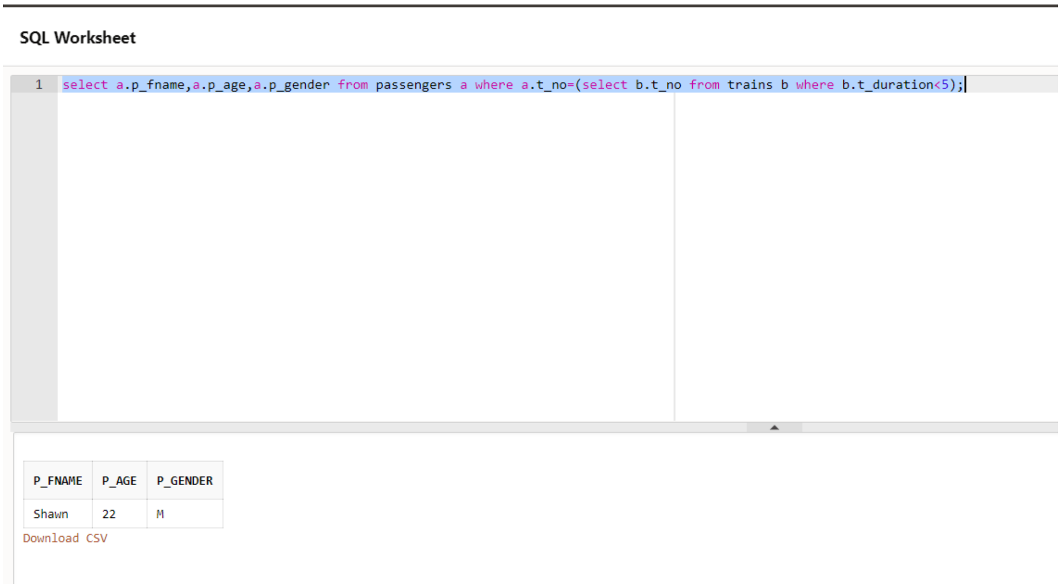
```
select email,p_age,p_contact from passengers where t_no in (select t_no from trains where t_status in(select t_status from tickets where p_id>3));
```

EMAIL	P_AGE	P_CONTACT
shri@gmail.com	18	9876890999
mukesh@gmail.com	18	9873422424

Download CSV
2 rows selected.

Correlated Querries:

a) **List the name,age,gender of the passengers whose train duration is less than 5 hours.**
select a.p_fname,a.p_age,a.p_gender from passengers a where a.t_no=(select b.t_no from trains b where b.t_duration<5);



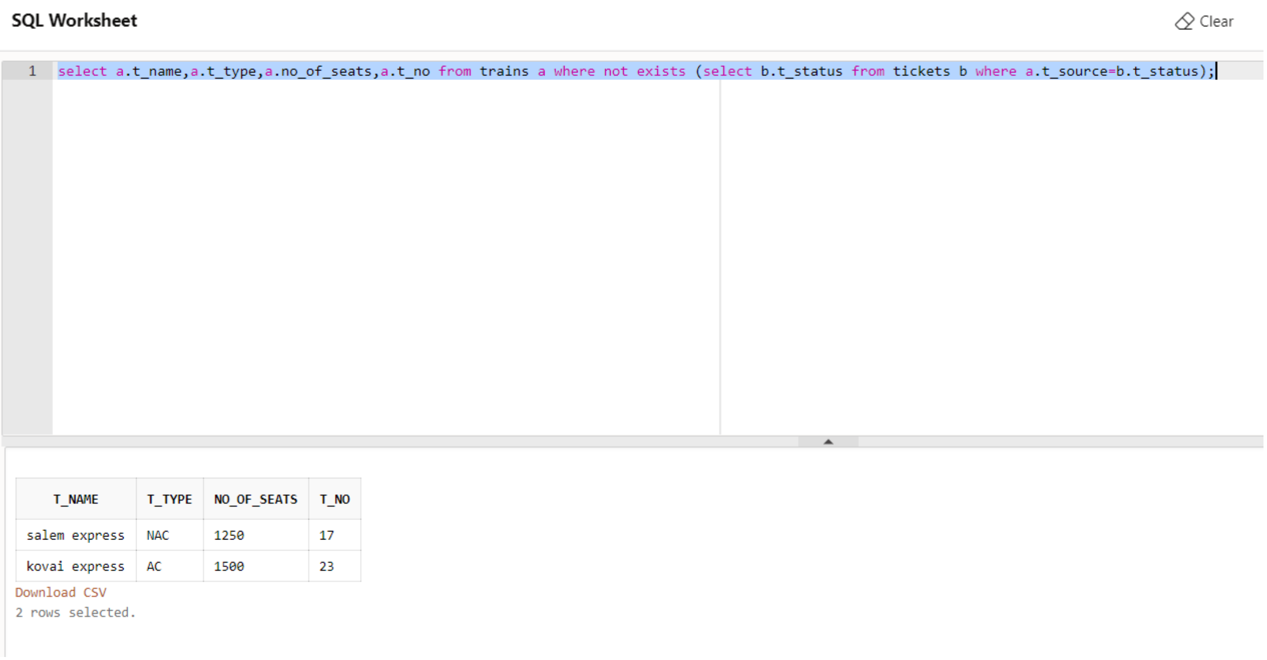
```
select a.p_fname,a.p_age,a.p_gender from passengers a where a.t_no=(select b.t_no from trains b where b.t_duration<5);
```

P_FNAME	P_AGE	P_GENDER
Shawn	22	M

Download CSV

b) List the name,train type ,trains's capacity,train number of passenger's for those train status and train source are same

select a.t_name,a.t_type,a.no_of_seats,a.t_no from trains a where not exists (select b.t_status from tickets b where a.t_source=b.t_status);



```
select a.t_name,a.t_type,a.no_of_seats,a.t_no from trains a where not exists (select b.t_status from tickets b where a.t_source=b.t_status);
```

T_NAME	T_TYPE	NO_OF_SEATS	T_NO
salem express	NAC	1250	17
kovai express	AC	1500	23

Download CSV

2 rows selected.

CURSORS:

1)DISPLAY THE STATION DETAILS OF STATION NUMBER 5

DECLARE

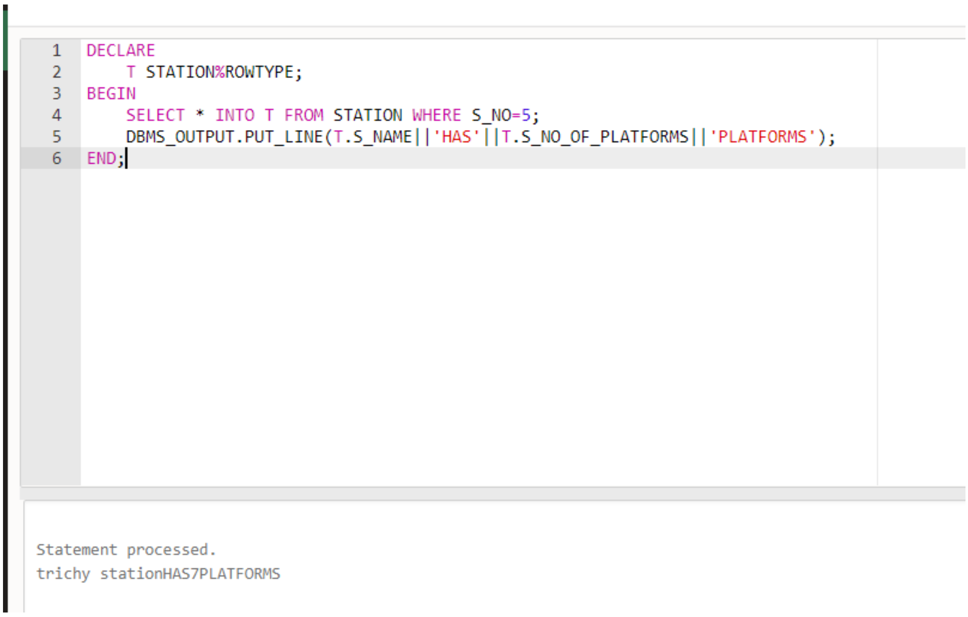
T STATION%ROWTYPE;

BEGIN

SELECT * INTO T FROM STATION WHERE S_NO=5;

DBMS_OUTPUT.PUT_LINE(T.S_NAME||'HAS'||T.S_NO_OF_PLATFORMS ||'PLATFORMS');

END;

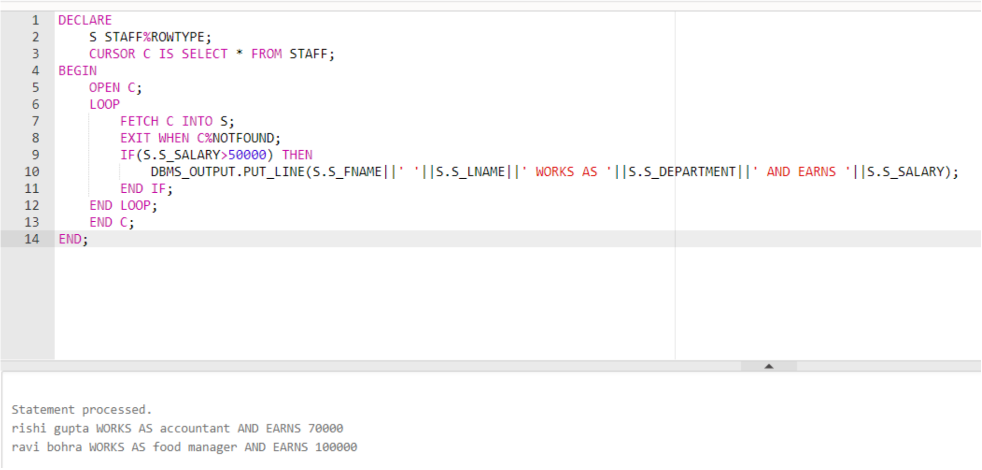


```
DECLARE
    T STATION%ROWTYPE;
BEGIN
    SELECT * INTO T FROM STATION WHERE S_NO=5;
    DBMS_OUTPUT.PUT_LINE(T.S_NAME||'HAS'||T.S_NO_OF_PLATFORMS||'PLATFORMS');
END;
```

```
Statement processed.
trichy stationHAS7PLATFORMS
```

2)SHOW THE STAFF DETAILS FOR THOSE WHO EARN MORE THAN 50000

```
DECLARE
   S STAFF%ROWTYPE;
   CURSOR C IS SELECT * FROM STAFF;
BEGIN
   OPEN C;
   LOOP
      FETCH C INTO S;
      EXIT WHEN C%NOTFOUND;
      IF(S.S_SALARY>50000) THEN
         DBMS_OUTPUT.PUT_LINE(S.S_FNAME||' '||S.S_LNAME||' WORKS AS '||S.S_DEPARTMENT||' AND EARNS '||S.S_SALARY);
      END IF;
   END LOOP;
   END C;
END;
```



```
DECLARE
    S STAFF%ROWTYPE;
    CURSOR C IS SELECT * FROM STAFF;
BEGIN
    OPEN C;
    LOOP
        FETCH C INTO S;
        EXIT WHEN C%NOTFOUND;
        IF(S.S_SALARY>50000) THEN
            DBMS_OUTPUT.PUT_LINE(S.S_FNAME||' '||S.S_LNAME||' WORKS AS '||S.S_DEPARTMENT||' AND EARNS '||S.S_SALARY);
        END IF;
    END LOOP;
    END C;
END;
```

```
Statement processed.
rishi gupta WORKS AS accountant AND EARNS 70000
ravi bohra WORKS AS food manager AND EARNS 100000
```

TRIGGERS

1)CREATE A TRIGGER WHEN THE TRAIN STATUS IS UPDATED

CREATE OR REPLACE

TRIGGER ALERT

AFTER

UPDATE ON TRAINS

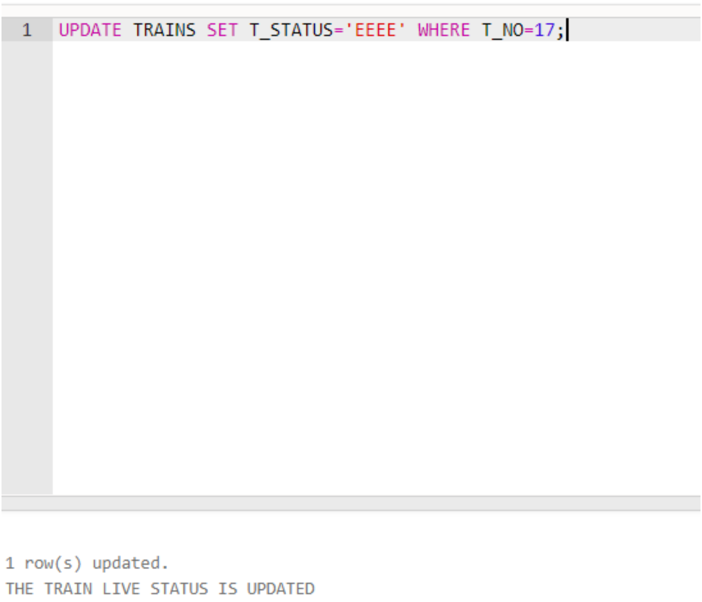
FOR EACH ROW

BEGIN

DBMS_OUTPUT.PUT_LINE('THE TRAIN LIVE STATUS IS UPDATED');

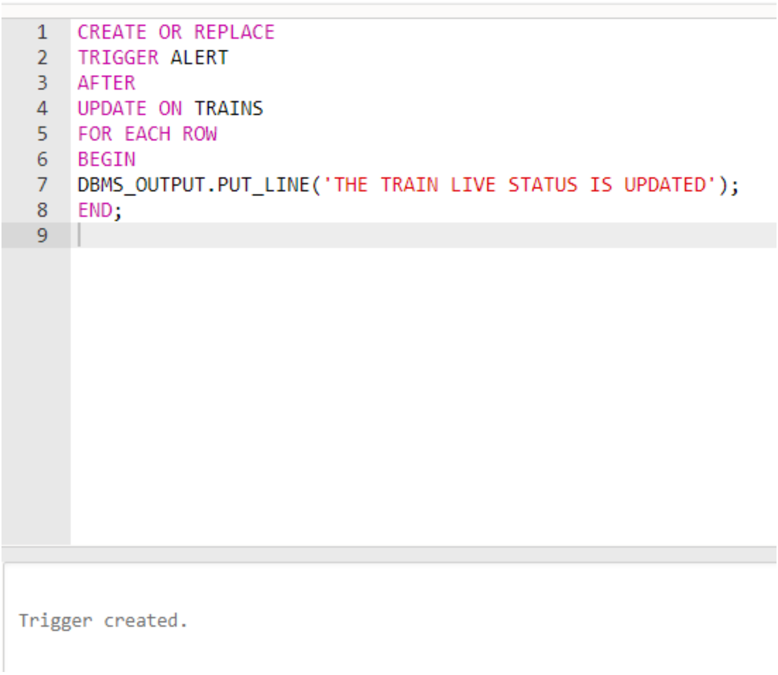
END;

UPDATE TRAINS SET T_STATUS='EEEE' WHERE T_NO=17;



```
UPDATE TRAINS SET T_STATUS='EEEE' WHERE T_NO=17;
```

```
1 row(s) updated.
THE TRAIN LIVE STATUS IS UPDATED
```



```
CREATE OR REPLACE
TRIGGER ALERT
AFTER
UPDATE ON TRAINS
FOR EACH ROW
BEGIN
DBMS_OUTPUT.PUT_LINE('THE TRAIN LIVE STATUS IS UPDATED');
END;
```

```
Trigger created.
```

2)CREATE A TRIGGER TO PROMPT ERROR MESSAGE WHEN A TRAIN’S STATION NAME,SOURCEOR DESTINATION IS UPDATED.

CREATE OR REPLACE

TRIGGER AL

BEFORE UPDATE OF T_NAME,T_SOURCE,T_DESTINATION ON TRAINS

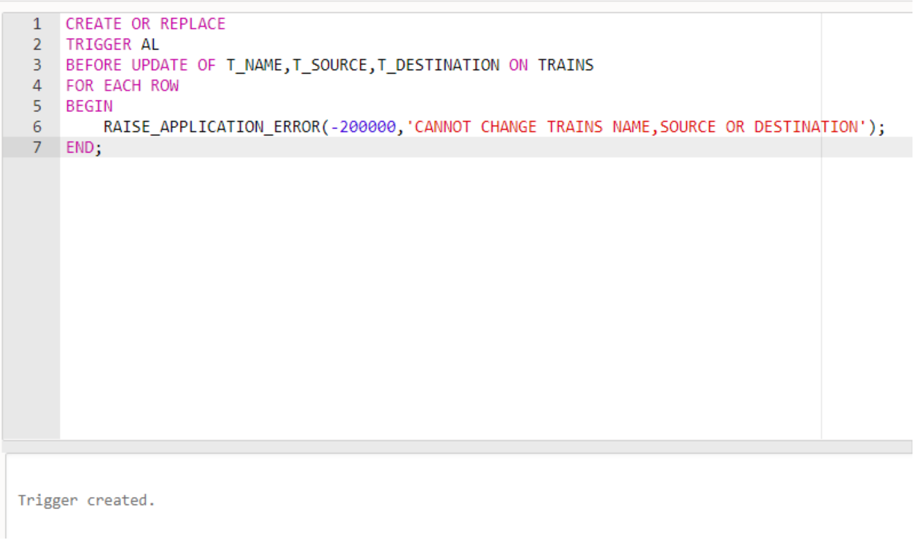
FOR EACH ROW

BEGIN

    RAISE_APPLICATION_ERROR(-200000,'CANNOT CHANGE TRAINS NAME,SOURCE OR DESTINATION');

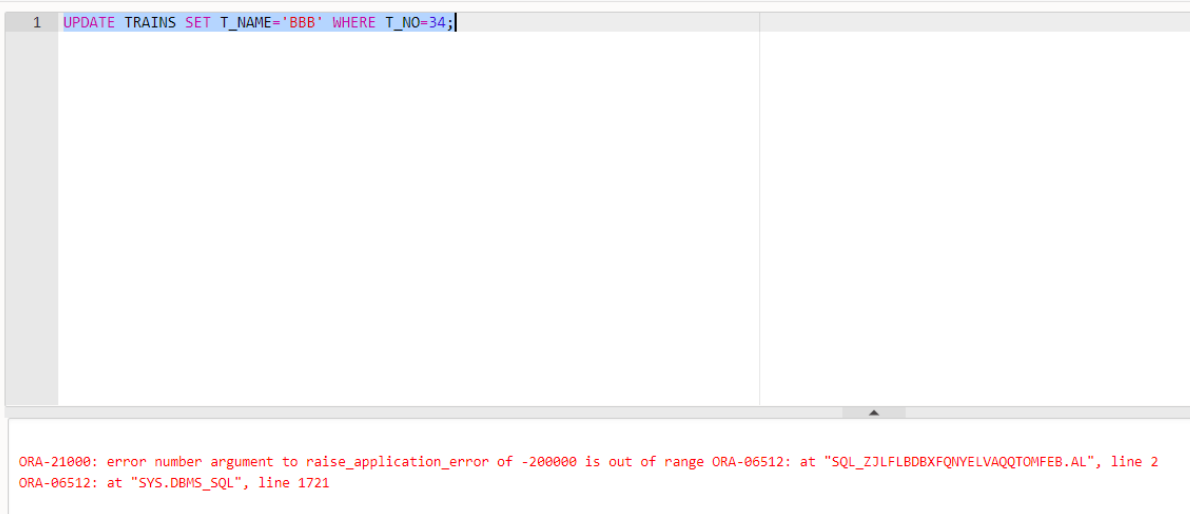
END;

UPDATE TRAINS SET T_NAME='BBB' WHERE T_NO=34;



```
CREATE OR REPLACE
TRIGGER AL
BEFORE UPDATE OF T_NAME,T_SOURCE,T_DESTINATION ON TRAINS
FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-200000,'CANNOT CHANGE TRAINS NAME,SOURCE OR DESTINATION');
END;
```

```
Trigger created.
```



```
UPDATE TRAINS SET T_NAME='BBB' WHERE T_NO=34;
```

```
ORA-21000: error number argument to raise_application_error of -200000 is out of range ORA-06512: at "SQL_ZJLFLBDBXFQNYELVAQQTOMFEB.AL", line 2
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

CONCLUSION:

Our project is a humble venture to satisfy the needs to manage railway information. This project shall prove to be effective and reliable. The program is error-free. The information stored in the form of database is secure. The system design has been done keeping user friendliness and efficiency in mind. In our project Railway reservation system we have also stored the information about the Trains scheduled and the users booking the tickets and even the status of the train, seats etc. This database is helpful for the applications which facilitates passengers to book the train tickets and check the details of trains and their status from their place itself it avoids inconveniences of going to the railway station for each and every query they get. We had considered the most important requirements only, many more features and details can be added to our project in order to obtain even more user friendly applications. These applications are already in progress and in future they can be upgraded and may become part of amazing technology.

REFERENCES:

1. Ramez Elmasri and Shamkant B. Navathe, ‘Fundamentals of Database Systems’ Addison Wesley Publishing company

2. Allan Leake, (2000), Definition of database

https://searchsqlserver.techtarget.com/definition/database

3. http://www.tutorialspoint.com/mysql/

4. SQL Bible, 2nd edition (Paperback)

5. http://www.slideshare.net

6.www.indianrail.gov.in

7.Indian Railways Vision 2020