RAILWAY RESERVATION SYSTEM

MINI PROJECT

NAME – Neha Sharma

UID - 20BCS4576

20IOT1-A

DATABASE MANAGEMENT SYSTEM LAB

PROJECT DESCRIPTION:

This project is about creating the database about Railway Reservation System.

The railway reservation system facilitates the passengers to enquire about the trains available on the basis of source and destination, booking and cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers. The record of train includes its number, name, source, destination, and days on which it is available, whereas record of train status includes dates for which tickets can be booked, total number of seats available, and number of seats already booked.

Passengers can book their tickets for the train in which seats are available. For this, passenger has to provide the desired train number and the date for which ticket is to be booked. Before booking a ticket for a passenger, the validity of train number and booking date is checked. Once the train number and booking date are validated, it is checked whether the seat is available. If yes, the ticket is booked with confirm status and corresponding ticket ID is generated which is stored along with other details of the passenger. The ticket once booked can be cancelled at any time. For this, the passenger has to provide the ticket ID (the unique key). The ticket ID is searched and the corresponding record is deleted. With this, the first. ticket with waiting status also gets confirmed.

List of Assumption Since the reservation system is very large in reality, it is not feasible to develop the case study to that extent and prepare documentation at the working of the reservation system. To implement this sample case study, some assumptions have been made, which are as follows:

1. The number of trains has been restricted to 5.

2. The booking is open only for next seven days from the current date.

3. Only categories of tickets can be booked, namely, AC and General. 4. The total number of tickets that can be booked in each category (AC and two General) is 10.

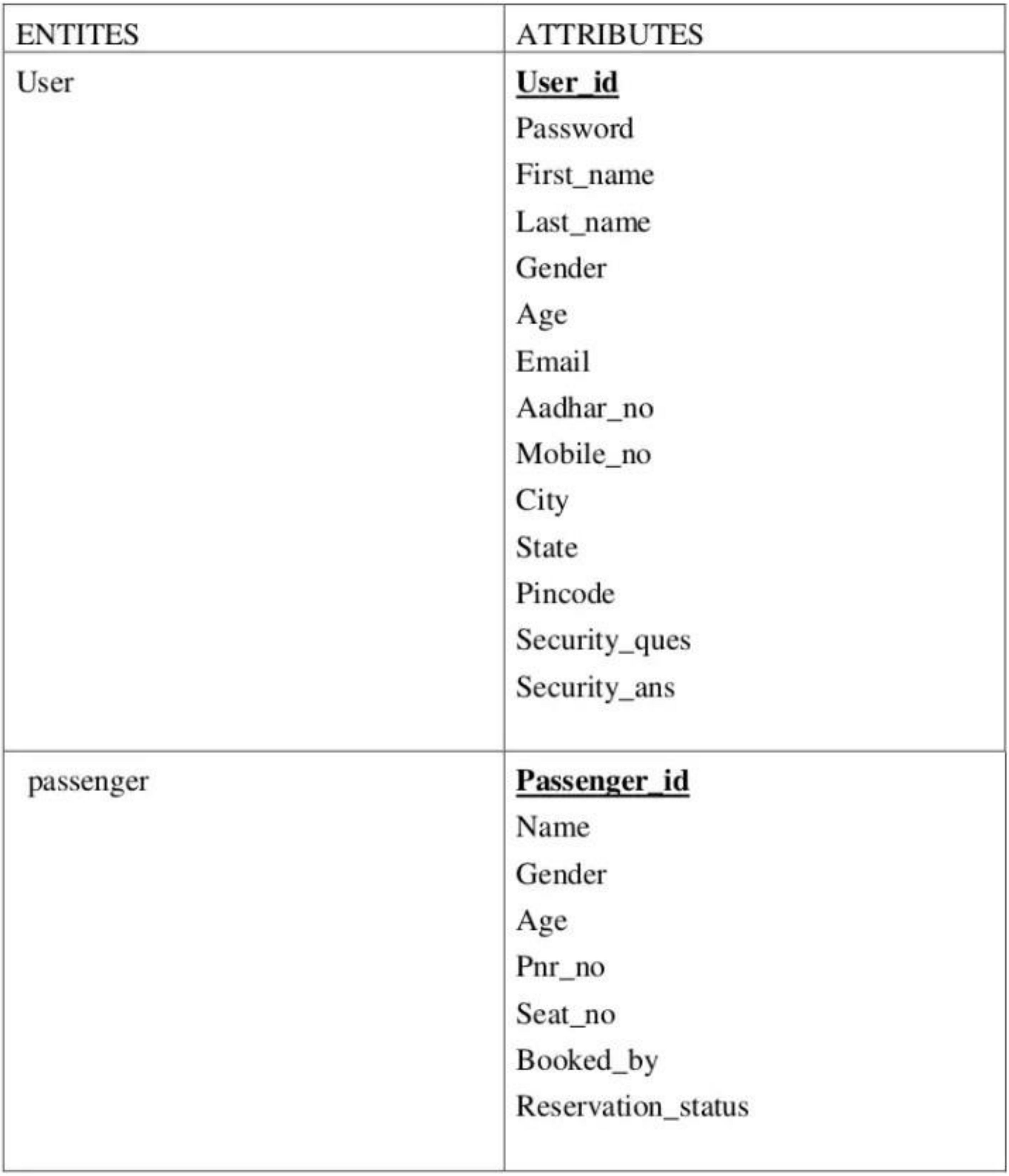
5. The total number of tickets that can be given the status of waiting is 2.

6. The in- between stoppage stations and their bookings are not considered.

List of trains has to be maintained. Detailed Passenger information is to be maintained in the booking procedure, the train number, train date, and category are read from the passenger.

On the basis of the values provided by the passenger, corresponding record is retrieved from the Train Status. If the desired category is AC, then total number of AC seats and number of booked AC seats are compared in order to find whether ticket can be booked or not. Similarly, it can be checked. for the general category. If ticket can be booked, then passenger details are read and stored in the Passenger table. In the cancellation procedure, ticket ID is read from the passenger and corresponding record is searched in the Passenger. If the record exists, it is deleted. After deleting the record (if it is confirmed), first record with waiting status for the same train and same category are searched from the Passenger table and its status is changed to confirm.

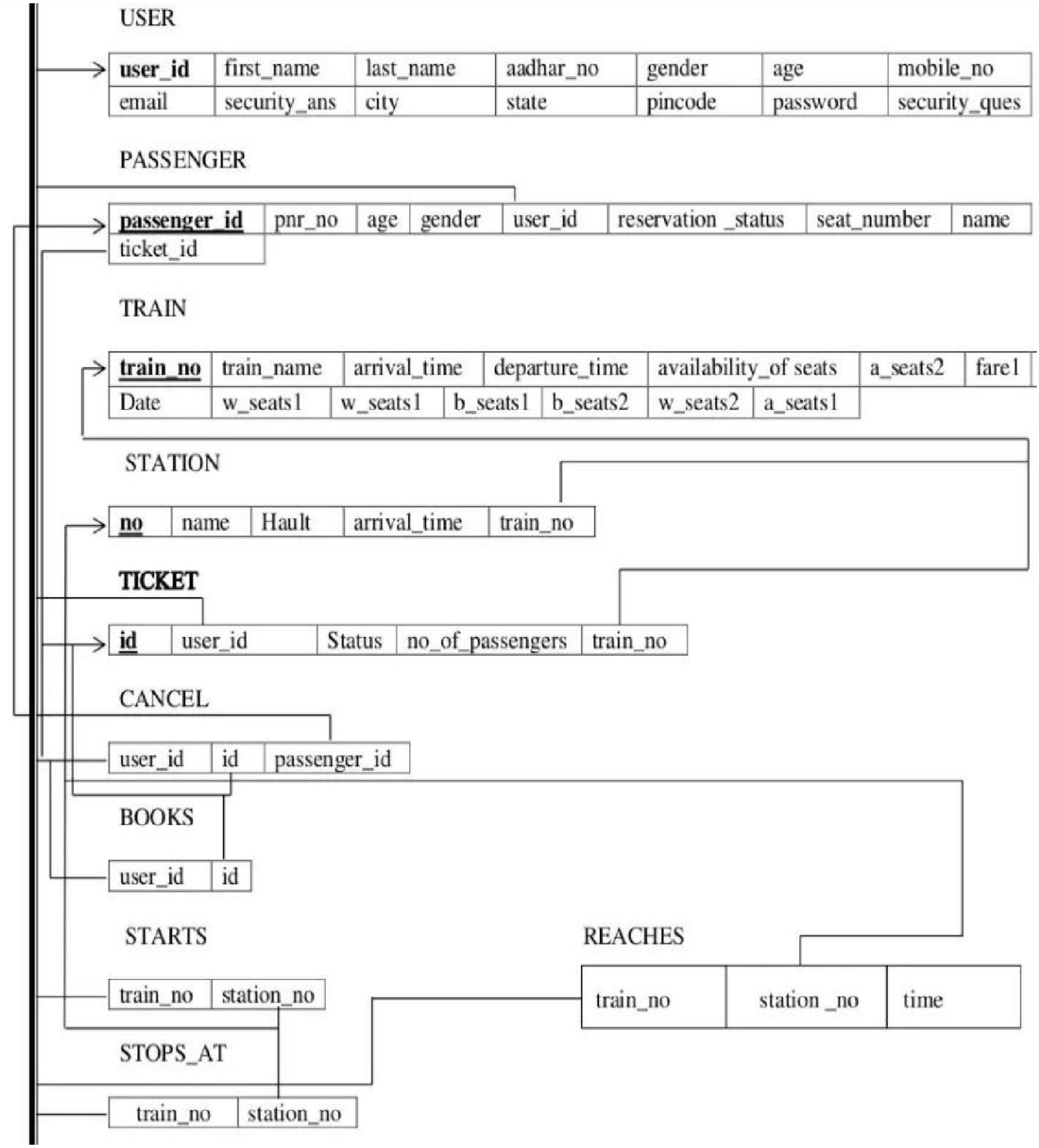
LIST OF ENTITIES & ATTRIBUTES:



ENTITES	ATTRIBUTES
User	**User_id** Password First_name Last_name Gender Age Email Aadhar_no Mobile_no City State Pincode Security_ques Security_ans
passenger	**Passenger_id** Name Gender Age Pnr_no Seat_no Booked_by Reservation_status

Train	**Train_no** Train_name Source Destination Arrival_time Departure_time Avalibility_of_seats Train_no A_seats1 A_seats2 A_seats3 B_seats1 B_seats2 B_seats3 W_Seats1 W_seats2 W_seats3
Station	Name **No** Train_no Arrival_time Hault
Ticket	**Id** Train_no Booked_user Status No_of_passengers

SCHEMA DIAGRAM:



USER

user_id	first_name	last_name	aadhar_no	gender	age	mobile_no
email	security_ans	city	state	pincode	password	security_ques

PASSENGER

passenger_id	pnr_no	age	gender	user_id	reservation _status	seat_number	name
ticket_id							

TRAIN

train_no	train_name	arrival_time	departure_time	availability_of seats	a_seats2	fare1	
Date	w_seats1	w_seats1	b_seats1	b_seats2	w_seats2	a_seats1	

STATION

no	name	Hault	arrival_time	train_no

TICKET

id	user_id	Status	no_of_passengers	train_no

CANCEL

user_id	id	passenger_id

BOOKS

user_id	id

STARTS

train_no	station_no

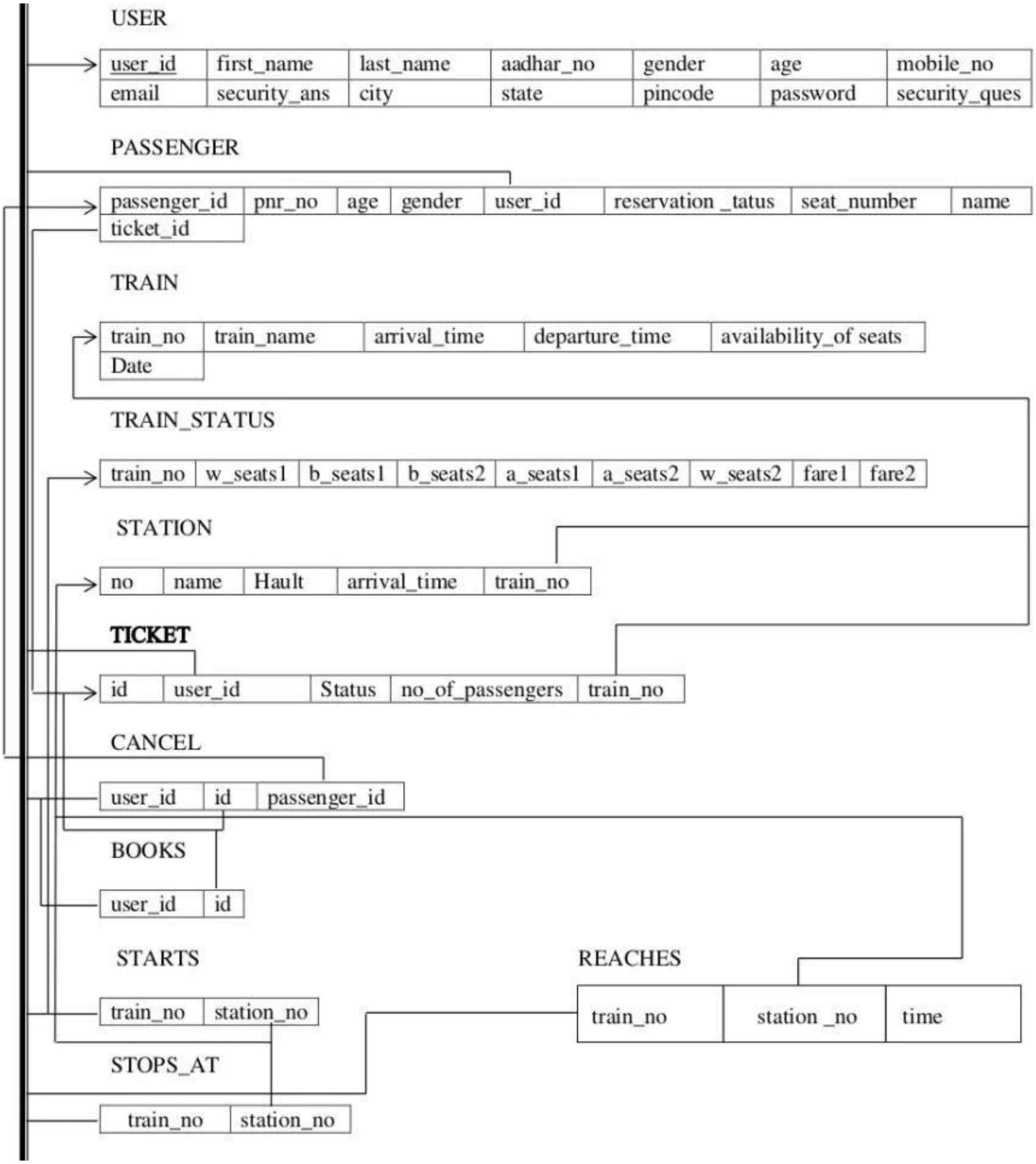
REACHES

train_no	station _no	time

STOPS_AT

train_no	station_no

NORMALIZATION & FINAL LIST OF RELATIONS:



USER

user_id	first_name	last_name	aadhar_no	gender	age	mobile_no
email	security_ans	city	state	pincode	password	security_ques

PASSENGER

passenger_id	pnr_no	age	gender	user_id	reservation _tatus	seat_number	name
ticket_id							

TRAIN

train_no	train_name	arrival_time	departure_time	availability_of seats
Date				

TRAIN_STATUS

train_no	w_seats1	b_seats1	b_seats2	a_seats1	a_seats2	w_seats2	fare1	fare2

STATION

no	name	Hault	arrival_time	train_no

TICKET

id	user_id	Status	no_of_passengers	train_no

CANCEL

user_id	id	passenger_id

BOOKS

user_id	id

STARTS

train_no	station_no

REACHES

train_no	station _no	time

STOPS_AT

train_no	station_no

FINAL LIST OF RELATION SHIPS:

- books - Ternary relationship between USER,TRAIN,PASSENGER and TICKET.
- starts -Between TRAIN and STATION
- reaches-Between TRAIN and STATION
- cancel-Between USER and TICKET
- stops at -Between TRAIN and STATION

CREATE COMMANDS:

create table if not exists USER(user_id int primary key,first_name varchar(50),last_name varchar(50),adhar_no varchar(20), gender char,age int,mobile_no varchar(50), email varchar(50),city varchar(50), state varchar(50),pincode varchar(20), password varchar(50), security_ques varchar(50), security_ans varchar(50));

reate table if not exsists TRAIN(train_no int primary key,train_name varchar(50), arrival_time time,departure_time time, availability of seats char,date date);

create table if not exists STATION(no int,name varchar(50),hault int,arrival_time time,train_noint,primary key(station_no,train_no),constarint foreign key(train_no) references TRAIN(train_no));

create table if not exsists TRAIN_STATUS(train_no int primary key,b_seats 1 int,b_seats2 int,a_seats1 int,a_seats2 int,w_seats1 int,w_seats2 int,farel float,fare2 float);

create table if not exsists TICKET(id int primary key,user_id int,status char,no_of_passengers int,train_no int,constraint foreign key (user_id) references USER(user_id),constraint foreign key (train_no) references TRAIN(train_no)); create table if not exists PASSENGER(passenger_id int primary key,pnr_no int,age int,gender char,user_id int,reservation_status char,seat_number varchar(5),name varchar(50),

```
ticket_id int,constraint foreign key(user_id) references
USER(user_id),constraint foreign key(ticket_id) references TICKET(id));
```

```
create table if not exsists STARTS( train_no int primary key,station_no
int,constraint foreign key(train_no) references TRAIN(train_no),constraint
foreign key(station_no) references STATION(no));
```

```
create table if not exsists STOPS_AT( train_no int station_no
int,constraint foreign key(train_no) references TRAIN(train_no),constraint
foreign key(station_no) references STATION(no));
```

```
create table if not exsists REACHES (train_no int,station_no int,time
time,constraint foreign key (train_no) references
TRAIN(train_no),constraint foreign key(station_no) references
STATION(no));
```

```
create table if not exsists BOOKS (user_id int,id int,constraint foreign
key(user_id) references USER(user_id),constraint foreign key(id)
references TICKET(id));
```

```
create table if not exsists CANCEL(user_id int,id int,passenger_id
int,constraint foreign key(id) references TICKET(id),constraint foreign key
(passenger_id) references PASSENGER (passenger_id),constraint
foreign key(user_id) references USER(user_id));
```

INSERT COMMANDS:

```
USER(user_id,first_name,last_name,
aadhar_no,gender,age,mobile_no,email,city,st ate,pincode, password,
security_ques, security_ans)
values(1701,'vijay','sharma','309887340843','M',34,9887786655',
'vijayl@gmail.co m','vijayawada', 'andhrapradesh',
'520001','12345@#','favouritecolour', 'red'), (1702, 'r ohith','kumar',
'456709871234','M',45,'9809666555', 'rohith | kumar@gmail.com','gu
ntur', 'andhrapradesh', '522004','12 @#345','favourite bike','bmw'), (1703,
'manas vi',
'sree','765843210987','F',20,9995550666','manasvi57@gmail.com','guntu
r', 'andhra pradesh','522004','0987hii','favourite flower', 'rose');
```

```
insert into TRAIN(train_no,train_name,arri val_time,departure_time,
availability_of seats,date) values(12711, pinakini
exp','113000','114000','A',20170410),(12315, commandel
exp','124500',125000', 'NA',20170410);
```

```
insert into STATION (no, name,hault,arrival_time,train_no)
values(111,'vijayawada', 10,'113000', 12711), (222,'tirupathi',5,'1
14500',12315);
```

```
insert into TRAIN STATUS(train_no,w_seats
1,b_seats1,b_seats2,a_seats 1,a_seats2.w_seats 2,fare 1,fare2)
values(12711,10,4,0,1,1,0,100,450),(12315,10,5,0,0,2,1,300,600);
```

```
insert into TICKET(id,user_id,status,no_of_passengers,train_no)
values(4001,1701, 'C',1,12711),(4002, 1702, 'NC', 1,123 15);
```

```
insert into PASSENGERS
(passenger_id,pnr_no,age,gender,user_id,reservation
_status,seat_number,name,ticket_id) values(5001,78965,45,'M', 1701,
'C','B6 45','ramesh',4001),(5002,54523,54,F', 1701, W','B3-21',
'surekha',4002); insert into STARTS(train_no,station_no)
values(12711,111),(12315,222); insert into
REACHES(train_no,station_no,time) values (12711,222, 040000'),
```

```
insert into STOPS_AT(train_no,station_no)
values(12711,222),(12315,111);(12315,111,053500');
```

```
insert into BOOKS (user_id,id) values(1701,4001),(1702,4002);
```

```
insert into CANCEL(user_id,id,passenger_id) values (1701,400 1,5001);
```

SQL QUERIES FOR REPORT GENERATION:

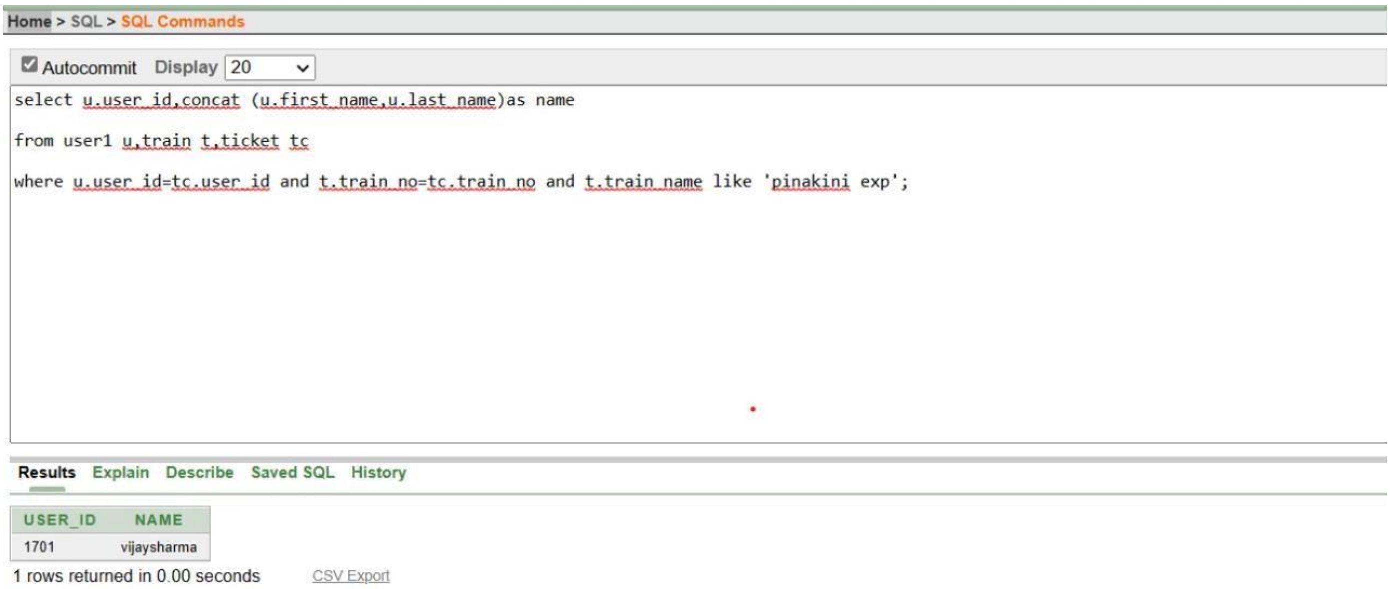
1.print user id and name of all those user who booked ticket for pinakini express

```
select u.user_id,concat (u.first_name,u.last_name)as name

from user u,train t,ticket tc

where u.user_id=tc.user_id and t.train_no=tc.train_no and t.train_name like 'pinakini
exp';
```

OUTPUT:



☑ Autocommit Display 20

```
select u.user_id,concat (u.first_name,u.last_name)as name

from user1 u,train t,ticket tc

where u.user_id=tc.user_id and t.train_no=tc.train_no and t.train_name like 'pinakini exp';
```

USER_ID	NAME
1701	vijaysharma

1 rows returned in 0.00 seconds

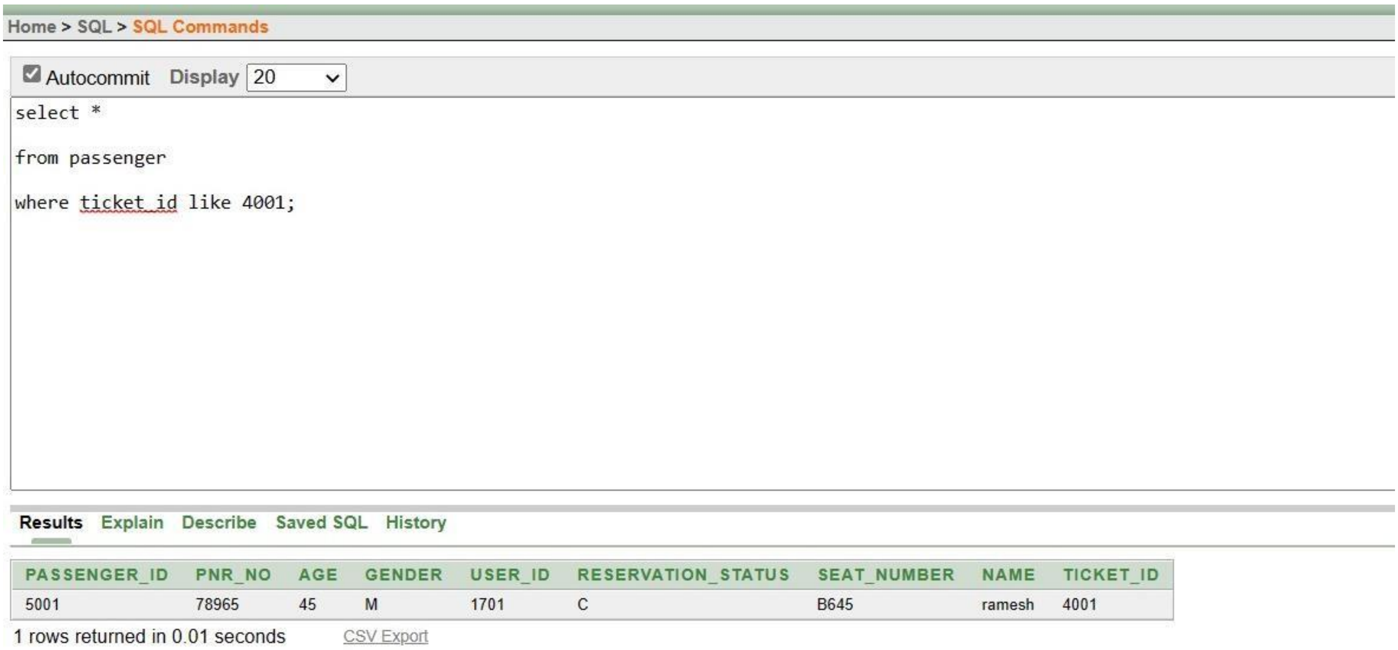
2. print details of passengers travelling under ticket no 4001

select *

from passenger

where ticket_id like 4001;

OUTPUT:



☑ Autocommit Display 20

```
select *

from passenger

where ticket_id like 4001;
```

PASSENGER_ID	PNR_NO	AGE	GENDER	USER_ID	RESERVATION_STATUS	SEAT_NUMBER	NAME	TICKET_ID
5001	78965	45	M	1701	C	B645	ramesh	4001

1 rows returned in 0.01 seconds

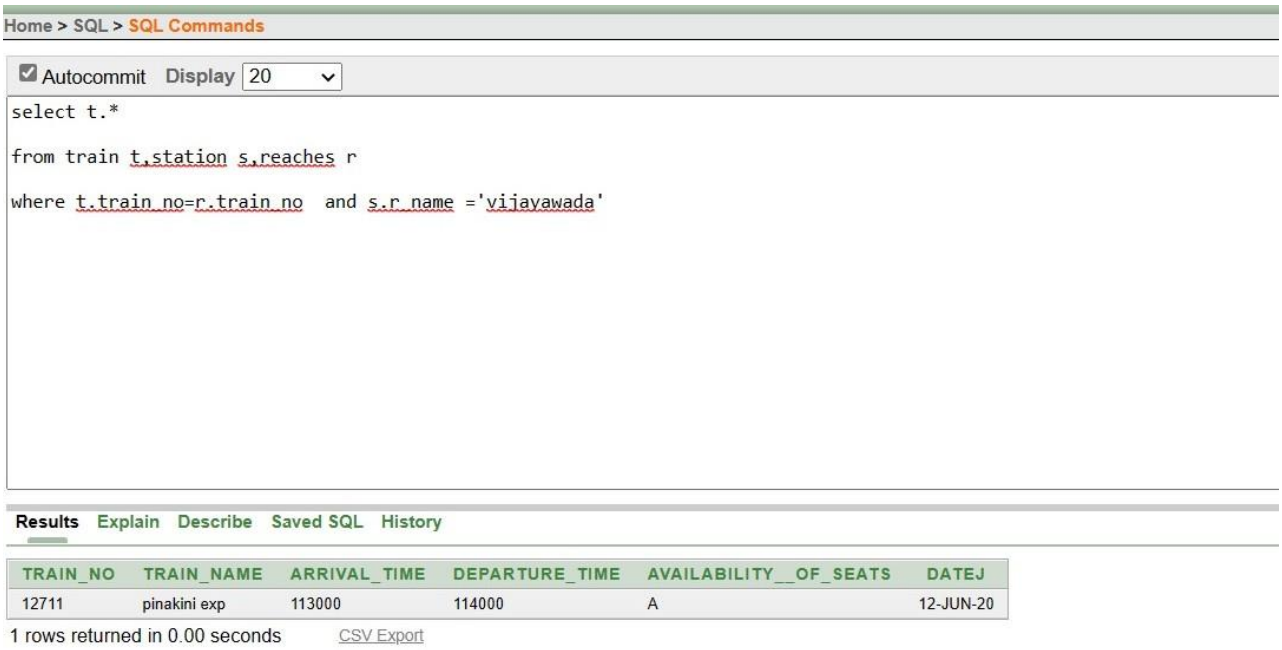
3. display all those train no's which reach station no

select t.*

from train t,station s,reaches r

where t.train_no=r.train_no and r.station_no=s.no and s.name like 'vijayawada';

OUTPUT:


Home > SQL > SQL Commands
☑ Autocommit Display 20
Results Explain Describe Saved SQL History
CSV Export

```
select t.*

from train t,station s,reaches r

where t.train_no=r.train_no  and s.r_name ='vijayawada'
```

TRAIN_NO	TRAIN_NAME	ARRIVAL_TIME	DEPARTURE_TIME	AVAILABILITY__OF_SEATS	DATEJ
12711	pinakini exp	113000	114000	A	12-JUN-20

1 rows returned in 0.00 seconds

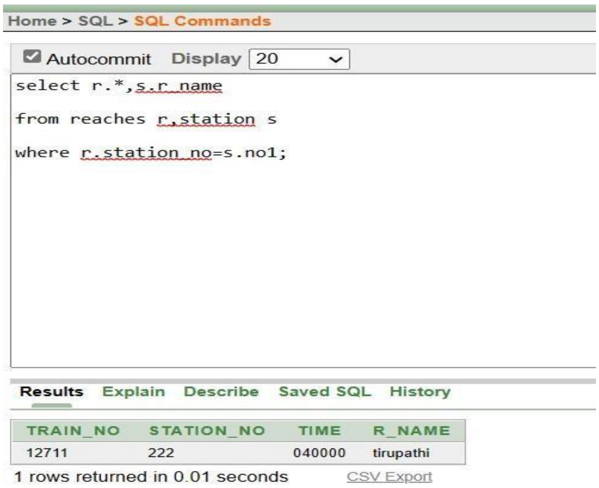
4. display time at which train no

select r.*,s.name

from reaches r,station s

where r.station_no=s.no;

OUTPUT:


Home > SQL > SQL Commands
☑ Autocommit Display 20
Results Explain Describe Saved SQL History
CSV Export

```
select r.*,s.r_name

from reaches r,station s

where r.station_no=s.no1;
```

TRAIN_NO	STATION_NO	TIME	R_NAME
12711	222	040000	tirupathi

1 rows returned in 0.01 seconds

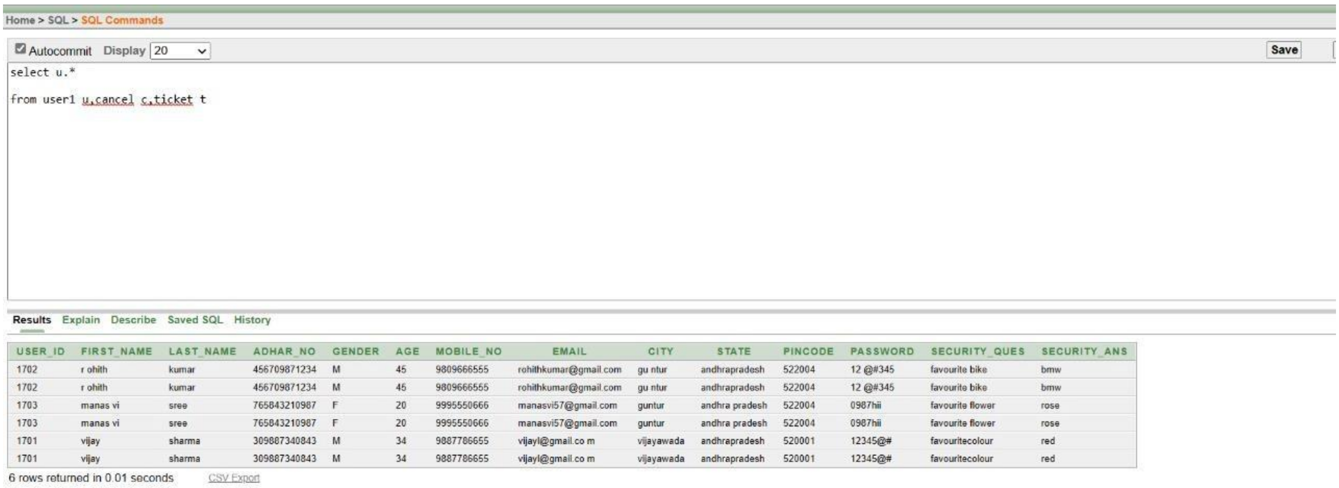
5. display details of all those users who cancelled tickets for train no

select u. *

from user u,cancel c,ticket t

where c.user_id=u.user_id and c.id=t.id and t.train_no like 12711;

OUTPUT:



Home > SQL > SQL Commands
☑ Autocommit Display 20
Save
Results Explain Describe Saved SQL History
CSV Export

```
select u.*

from user1 u,cancel c,ticket t
```

USER_ID	FIRST_NAME	LAST_NAME	ADHAR_NO	GENDER	AGE	MOBILE_NO	EMAIL	CITY	STATE	PINCODE	PASSWORD	SECURITY_QUES	SECURITY_ANS
1702	r ohith	kumar	456709871234	M	45	9809666555	rohithkumar@gmail.com	gu ntur	andhrapradesh	522004	12 @#345	favourite bike	bmw
1702	r ohith	kumar	456709871234	M	45	9809666555	rohithkumar@gmail.com	gu ntur	andhrapradesh	522004	12 @#345	favourite bike	bmw
1703	manas vi	sree	765843210987	F	20	9995550666	manasvi57@gmail.com	guntur	andhra pradesh	522004	0987hii	favourite flower	rose
1703	manas vi	sree	765843210987	F	20	9995550666	manasvi57@gmail.com	guntur	andhra pradesh	522004	0987hii	favourite flower	rose
1701	vijay	sharma	309887340843	M	34	9887786655	vijayl@gmail.co m	vijayawada	andhrapradesh	520001	12345@#	favouritecolour	red
1701	vijay	sharma	309887340843	M	34	9887786655	vijayl@gmail.co m	vijayawada	andhrapradesh	520001	12345@#	favouritecolour	red

6 rows returned in 0.01 seconds

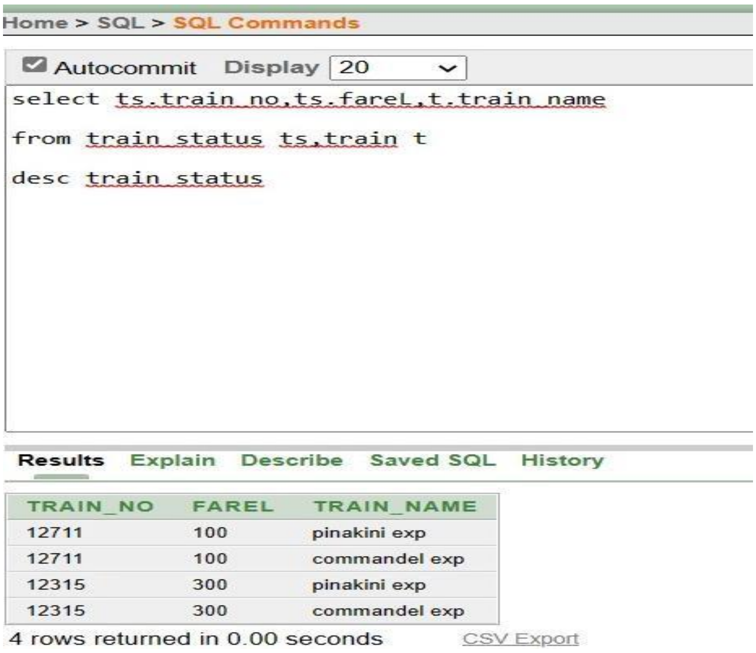
6. display the train no with increasing order of the fares of class 1

select ts.train_no,ts.fare 1,t.train_name

from train status ts,train t

where t.train_no=ts.train_no order by fare l asc;

OUTPUT:



Home > SQL > SQL Commands
☑ Autocommit Display 20
Results Explain Describe Saved SQL History
CSV Export

```
select ts.train_no,ts.fareL,t.train_name

from train_status ts,train t

desc train_status
```

TRAIN_NO	FAREL	TRAIN_NAME
12711	100	pinakini exp
12711	100	commandel exp
12315	300	pinakini exp
12315	300	commandel exp

4 rows returned in 0.00 seconds

7.display passenger details for train pinakini.

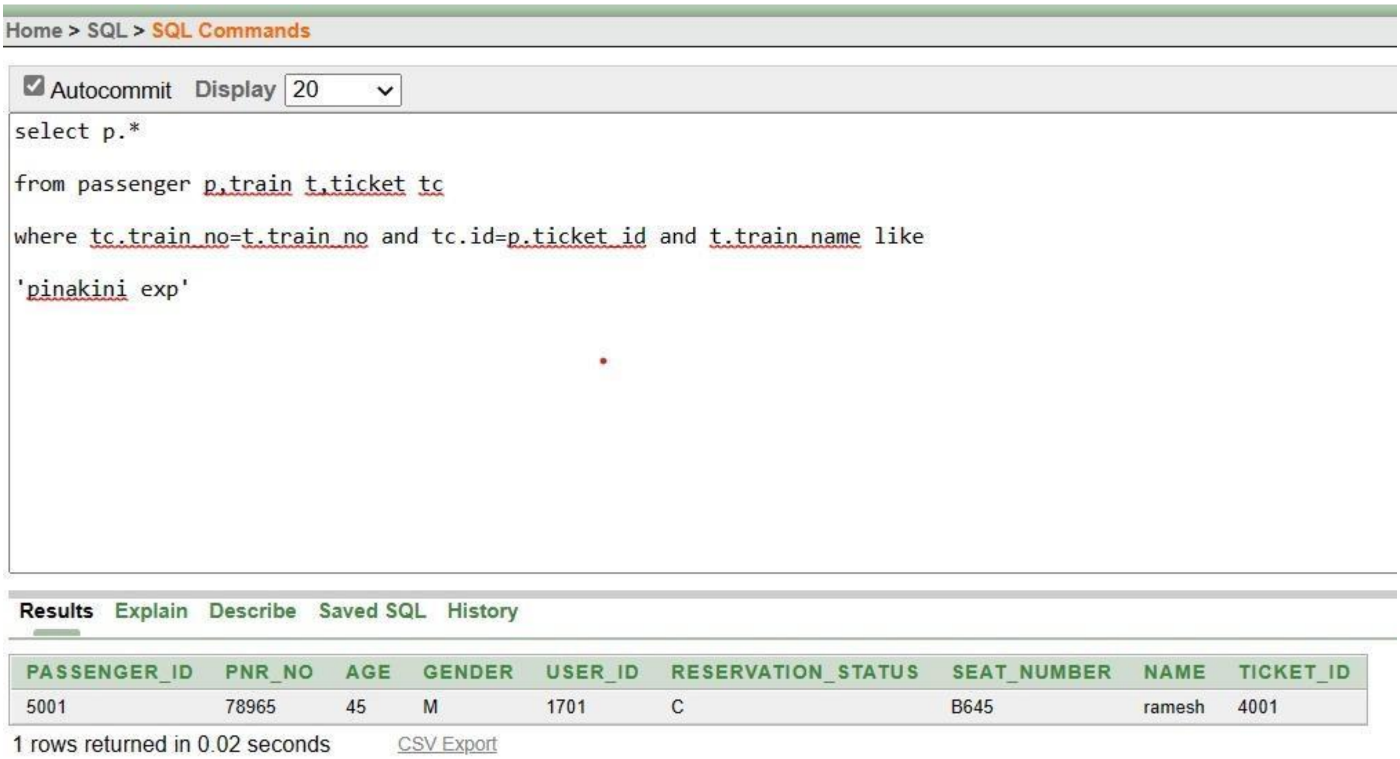
select p.*

from passenger p,train t,ticket tc

where tc.train_no=t.train_no and tc.id=p.ticket_id and t.train_name like

'pinakini exp';

OUTPUT:


Home > SQL > SQL Commands
☑ Autocommit Display 20
Results Explain Describe Saved SQL History
CSV Export

```
select p.*

from passenger p,train t,ticket tc

where tc.train_no=t.train_no and tc.id=p.ticket_id and t.train_name like

'pinakini exp'
```

PASSENGER_ID	PNR_NO	AGE	GENDER	USER_ID	RESERVATION_STATUS	SEAT_NUMBER	NAME	TICKET_ID
5001	78965	45	M	1701	C	B645	ramesh	4001

1 rows returned in 0.02 seconds

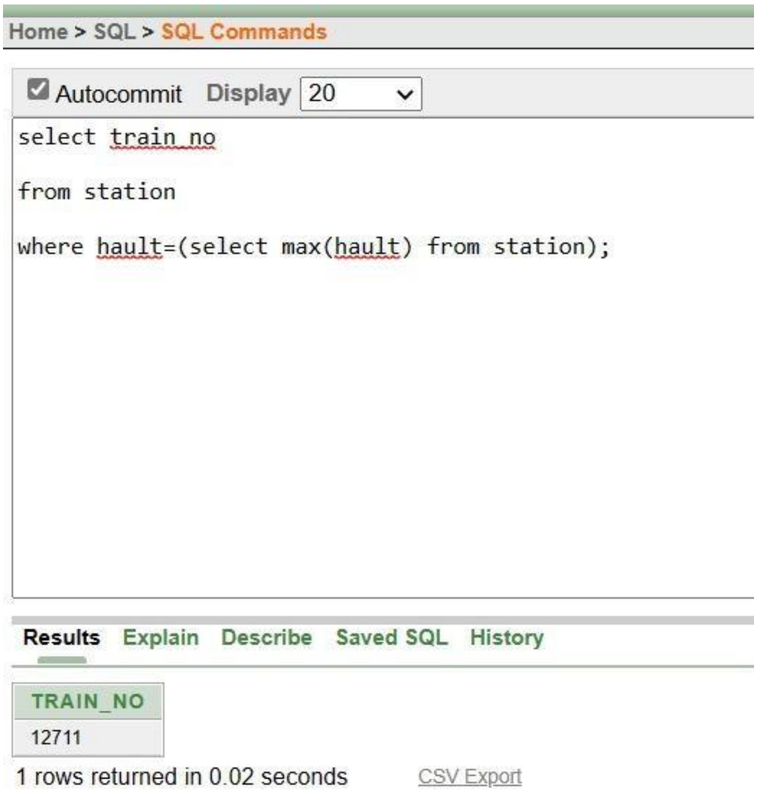
8. display the train no which haults for more time in station no

select train_no

from station

having max(hault);

OUTPUT:


Home > SQL > SQL Commands
☑ Autocommit Display 20
Results Explain Describe Saved SQL History
CSV Export

```
select train_no

from station

where hault=(select max(hault) from station);
```

TRAIN_NO
12711

1 rows returned in 0.02 seconds

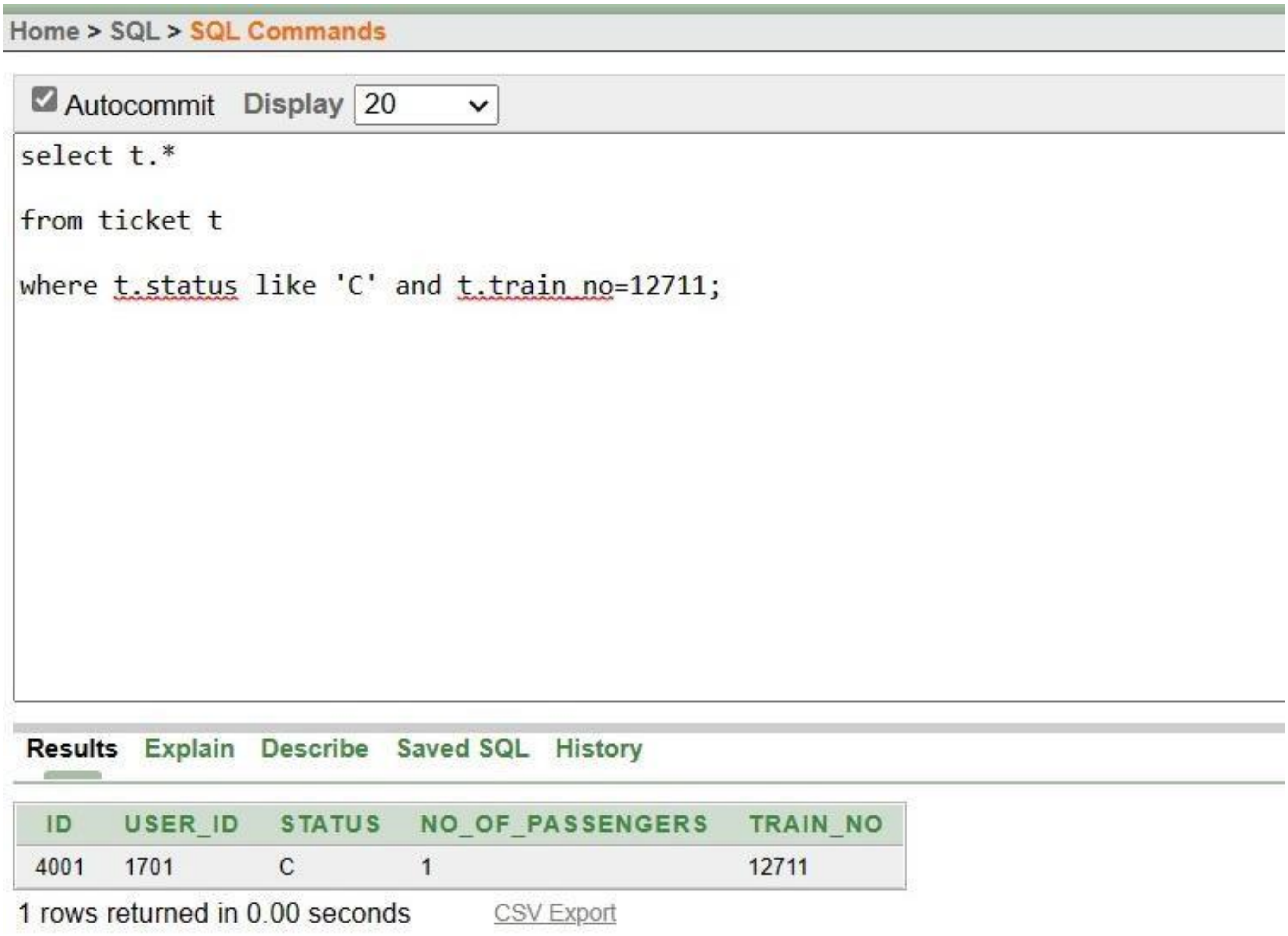
9. display details of all those passengers whose status is confirmed for train no

select t.*

from ticket t

where t.status like 'c' and t.train_no=12711;

OUTPUT:



```
select t.*

from ticket t

where t.status like 'C' and t.train_no=12711;
```

ID	USER_ID	STATUS	NO_OF_PASSENGERS	TRAIN_NO
4001	1701	C	1	12711

1 rows returned in 0.00 seconds

CONCLUSION:

In our project Railway reservation system, we have stored all the information about the Trains scheduled and the user’s booking tickets and even status of trains, seats etc. This data base is helpful for the applications which facilitate 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 railway station for each and every query they get. We had considered the most important requirements only, many more features and details cand 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.