UNIVERSITY OF ASIA PACIFIC

Department of Computer Science and Engineering

Course Title: Database System Lab

Course Code: CSE212

Project: Railway Management System

Group Name: DATAEX

Submitted by:

1. Name: Nur Habibah Binti Mahbub
   ID: 21201103
2. Name: Maria Akter Khushi
   ID: 21201107
3. Name: Ritu Rani Banik
   ID: 21201108
4. Name: Jannatul Ferdusi
   ID: 21201114
5. Name: Ahsan Ullah Rakib
   ID: 21201117

Submitted To:

Sabiha Tahsin Soha
Lecturer
University of Asia Pacific

Introduction:

The Railway Management System Database is a complete and effective solution created to simplify and improve the operations of railway networks. Train timetables, ticketing data, passenger records, and other elements of railway management are all stored and managed in this database, which acts as a central repository.

The Railway Management System Database facilitates smooth coordination and communication amongst many stakeholders involved in railway operations by utilizing advanced technology along with data management strategies. In addition to ensuring accurate and current information and improving overall operational efficiency, it makes the allocation of resources more effective. This database functions as a centralized location for organizing and storing a variety of railway management-related aspects, such as passenger information, train schedules, and ticketing data.

Entities:

- **Train** --- This entity represents a specific train within the system with its unique train id.

  [Its purpose is to store and manage information related to a train, such as its schedule, route, unique id of the train and its name.]

- **Passenger** --- This entity represents an individual who travels on the train.

  [Its purpose is to store and manage passenger-related information, including personal details.]

- **User** --- This entity represents the individuals who actually interact with the system to perform numerous tasks and access its functionalities.

- **Ticket**--- This entity is a record or document that authorizes a passenger to ride on a certain train. It acts as documentation of the payment and reservation along with date and time for a specific trip.

- **Special Ticket** --- This entity is a record or document that resembles the ticket entity but differs in some manner. This entity offers a special reduction in price to those who are disabled, older citizens, and aliens' .

- **Train Status** --- This entity refers to the current state or condition of a train within the system. It provides information about the availability, booked seats along with its time.
- **Timetable** --- This entity is an organised schedule that shows train departure and arrival times. It is an essential tool for controlling and scheduling train services for both passengers and railway operators.

- **Payment** --- This entity describes the procedure and means through which users or passengers transact money in order to buy tickets provided by the railway network.

- **Station** --- This entity represents a physical location where trains stop to board and disembark passengers.

- **Booking** --- This entity refers to the process of reserving seats or tickets for passengers on a train.

Attributes:

- User
  - u_id(Primary key)
  - u_name
  - password
  - email
  - phone
  - address

- Passenger
  - p_id(Primary key)
  - name
  - age
  - gender
  - u_id(Foreign key)

- Ticket
  - ticket_id(Primary key)
  - p_name(Foreign key)
  - t_id(Foreign key)

- s_id(Foreign key)
- class type
- seat_no
- s_departure
- s_destination
- issue date
- journey date

➔ Special Ticket

- special_id(Primary key)
- ticket_id(Foreign key)
- discount
- special_service

➔ Train status

- status_id(Primary key)
- t_id(Foreign key)
- date
- available seat
- booked seat

➔ Train

- t_id(Primary Key)
- t_name
- class type
- s_departure
- s_destination

➔ Time Table

- time_id(Primary key)
- t_id(Foreign key)
- schedule

➔ Station

- s_id(primary key)
- s_name
- location
- t_id(Foreign key)

➔ Payment

- ➢ pay_id(Primary key)
- ➢ ticket_id (Foreign key)
- ➢ amount
- ➢ pay_type
- ➢ pay date
- ➢ pay time
- ➢ u_id (Foreign key)

➔ Booking

- ➢ book_id (Primary key)
- ➢ u_id(Foreign key)
- ➢ ticket_id (Foreign key)
- ➢ book_date
- ➢ status
- ➢ seat_no

Relationships Sets:

- ✓ 1:1 for one-to-one relationships
- ✓ 1:M for one-to-many relationships
- ✓ M:M for many-to-many relationships

1. User and Passenger (1:M)

- A user can have one or more passengers.
- Passengers are associated with one user.

2. Passenger and Ticket (1:M)

- A passenger can have one or more tickets.
- Tickets are associated with one passenger.

3. Ticket and Special Ticket (1:1)

- A ticket can have one special ticket, and a special ticket is associated with one ticket.

4. Train Status and Train (1:M)

- Train status is associated with one train.
- A train can have multiple statuses.

5. Train and Time Table (1:M)

- A train is associated with one or more entries in the timetable.
- Each timetable entry is associated with one train.

6. Train and Station (M:M)

- A train can serve multiple stations, and a station can be served by multiple trains.
- The relationship table would have the train ID and station ID as foreign keys.

7. User and Payment (1:M)

- A user can make one or more payments.
- Payments are associated with one user.

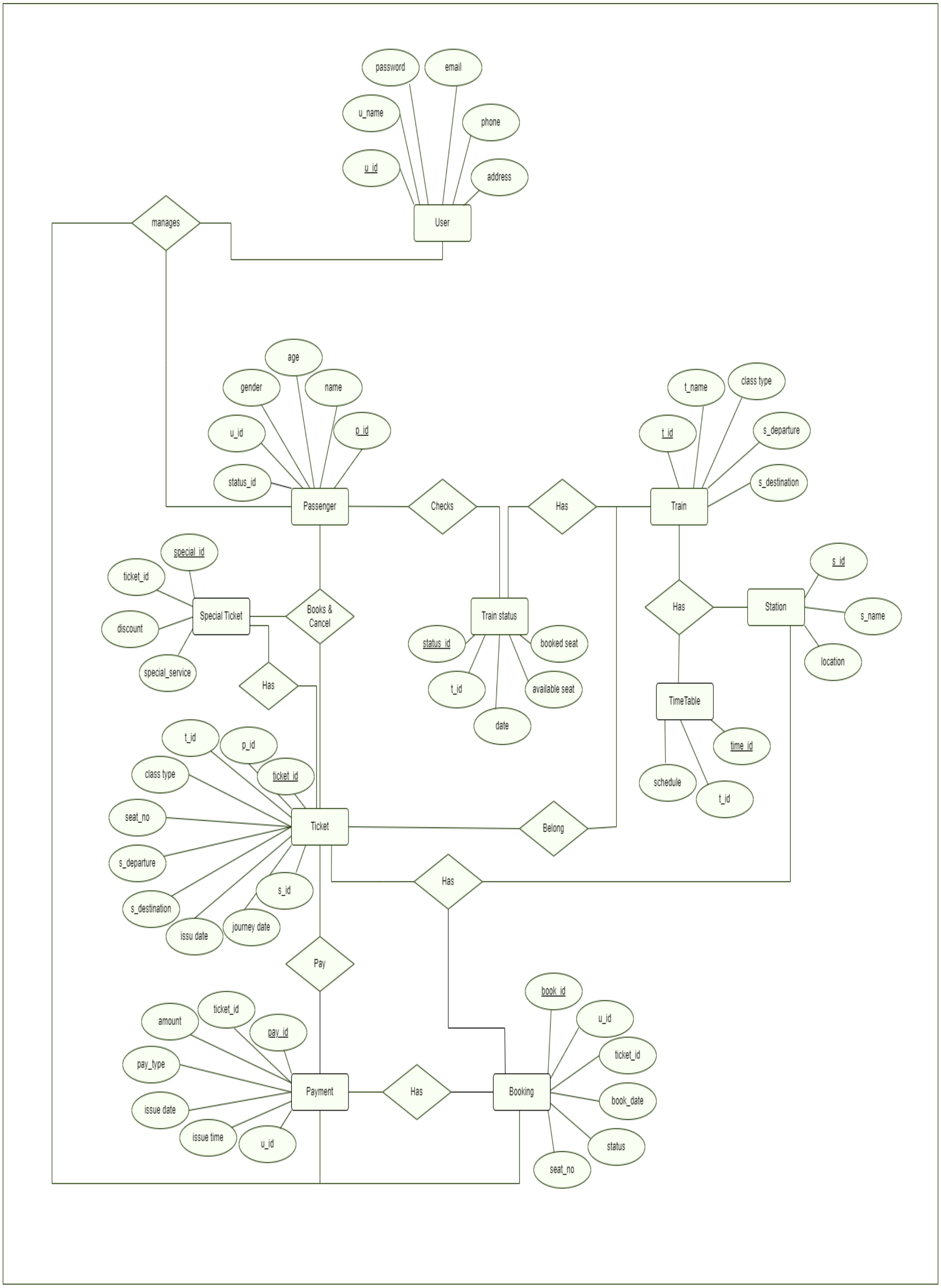
8. User and Booking (1:M)

- A user can have one or more bookings.
- Bookings are associated with one user.

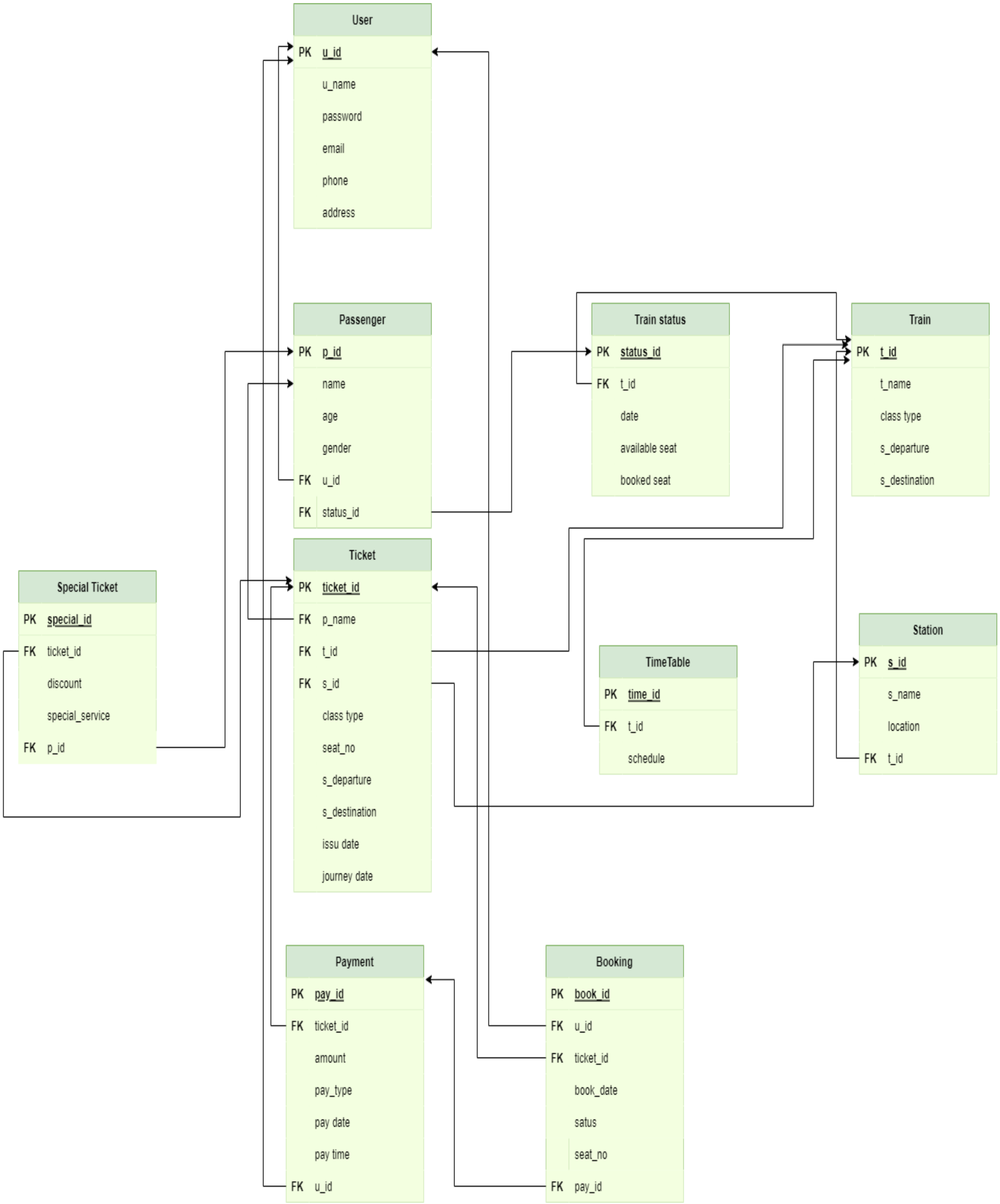
9. Booking and Ticket (1:1)

- Each booking is associated with one ticket, and each ticket is associated with one booking.

ER- Diagram:


password
email
u_name
phone
u_id
address
User
manages
age
gender
name
u_id
p_id
status_id
Passenger
Checks
Has
t_name
class type
t_id
s_departure
s_destination
Train
special_id
ticket_id
Special Ticket
discount
special_service
Books & Cancel
Has
Train status
status_id
booked seat
t_id
available seat
date
Has
Station
s_id
s_name
location
TimeTable
time_id
schedule
t_id
t_id
p_id
class type
ticket_id
seat_no
Ticket
Belong
s_departure
Has
s_id
s_destination
journey date
issu date
Pay
book_id
u_id
ticket_id
amount
pay_id
pay_type
Payment
Has
Booking
book_date
issue date
issue time
u_id
status
seat_no

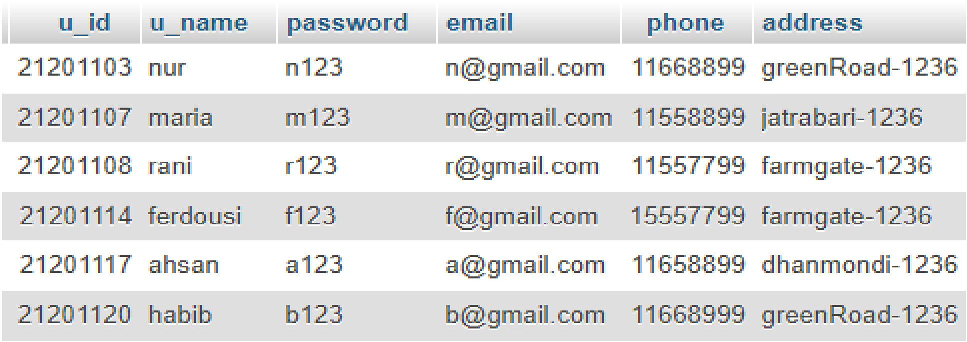
Schema Diagram:



SQL Queries:

1. Show the full list of users.

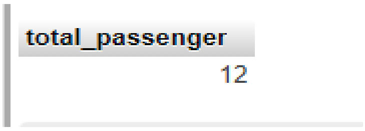
→ SELECT * FROM `users`



u_id	u_name	password	email	phone	address
21201103	nur	n123	n@gmail.com	11668899	greenRoad-1236
21201107	maria	m123	m@gmail.com	11558899	jatrabari-1236
21201108	rani	r123	r@gmail.com	11557799	farmgate-1236
21201114	ferdousi	f123	f@gmail.com	15557799	farmgate-1236
21201117	ahsan	a123	a@gmail.com	11658899	dhanmondi-1236
21201120	habib	b123	b@gmail.com	11668999	greenRoad-1236

2. Find the total number of passengers. Each passenger cannot be counted twice.

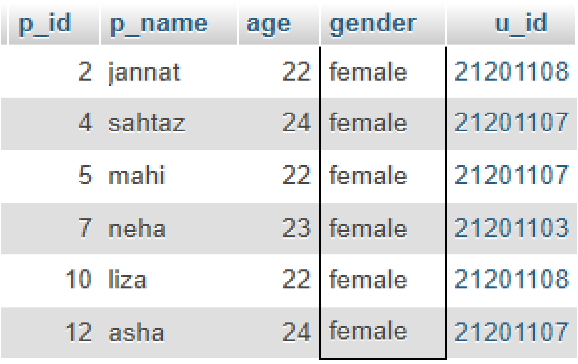
→ SELECT count(distinct `p_id`) as total_passenger FROM `passenger` ;



total_passenger
12

3. Find out the list of all female passengers in the passenger table.

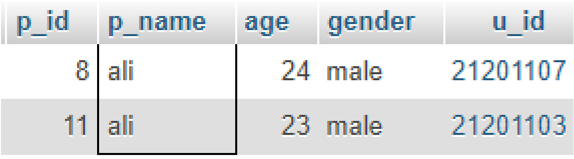
```
→ SELECT `p_id`, `p_name`, `age`, `gender`, `u_id`
  FROM `passenger`
  WHERE gender='female'
```



p_id	p_name	age	gender	u_id
2	jannat	22	female	21201108
4	sahtaz	24	female	21201107
5	mahi	22	female	21201107
7	neha	23	female	21201103
10	liza	22	female	21201108
12	asha	24	female	21201107

4. Find the list of passengers named ‘ali’.

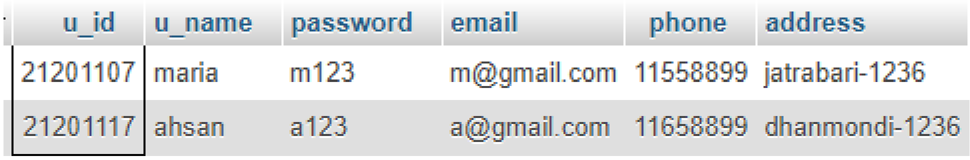
→
```
SELECT `p_id`, `p_name`, `age`, `gender`, `u_id`
FROM `passenger`
WHERE p_name='ali';
```



p_id	p_name	age	gender	u_id
8	ali	24	male	21201107
11	ali	23	male	21201103

5. Find the user IDs that start with 2 and ends with 7 from users table.

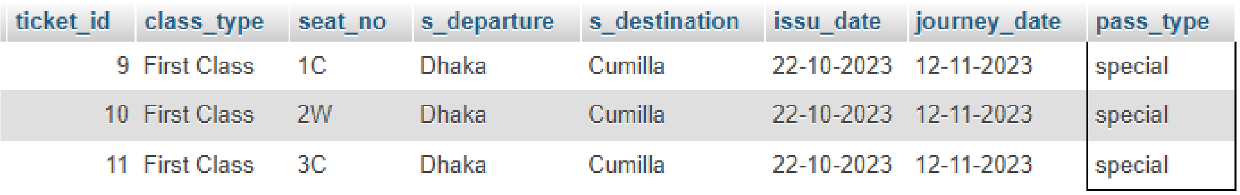
→
```
SELECT `u_id`, `u_name`, `password`, `email`, `phone`, `address`
FROM `users`
WHERE u_id like  '2______7';
```



u_id	u_name	password	email	phone	address
21201107	maria	m123	m@gmail.com	11558899	jatrabari-1236
21201117	ahsan	a123	a@gmail.com	11658899	dhanmondi-1236

6. Find the ‘special’ pass type from ticket table.

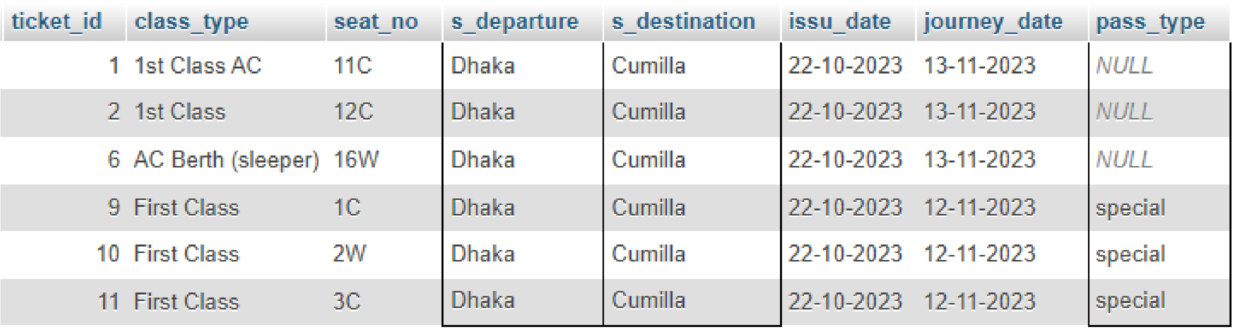
→
```
SELECT  `ticket_id`,  `class_type`,  `seat_no`,  `s_departure`,  `s_destination`,  `issu_date`,
`journey_date`, `pass_type`
 FROM `ticket`
WHERE pass_type = 'special';
```



ticket_id	class_type	seat_no	s_departure	s_destination	issu_date	journey_date	pass_type
9	First Class	1C	Dhaka	Cumilla	22-10-2023	12-11-2023	special
10	First Class	2W	Dhaka	Cumilla	22-10-2023	12-11-2023	special
11	First Class	3C	Dhaka	Cumilla	22-10-2023	12-11-2023	special

7. Find out the list of ticket_id where the train departs from Dhaka to the destination Cumilla from ticket table.

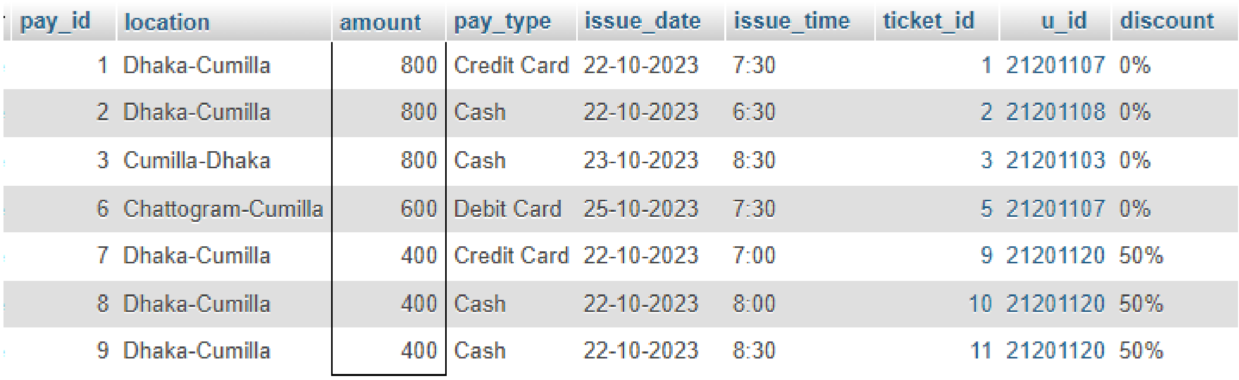
→
```
SELECT `ticket_id`, `class_type`, `seat_no`, `s_departure`, `s_destination`, `issu_date`,
`journey_date`, `pass_type`
FROM `ticket`
WHERE  s_departure = 'Dhaka' and s_destination = 'Cumilla';
```



ticket_id	class_type	seat_no	s_departure	s_destination	issu_date	journey_date	pass_type
1	1st Class AC	11C	Dhaka	Cumilla	22-10-2023	13-11-2023	NULL
2	1st Class	12C	Dhaka	Cumilla	22-10-2023	13-11-2023	NULL
6	AC Berth (sleeper)	16W	Dhaka	Cumilla	22-10-2023	13-11-2023	NULL
9	First Class	1C	Dhaka	Cumilla	22-10-2023	12-11-2023	special
10	First Class	2W	Dhaka	Cumilla	22-10-2023	12-11-2023	special
11	First Class	3C	Dhaka	Cumilla	22-10-2023	12-11-2023	special

8. Find out the list of pay_id from payment table where the payment amount is between 400 to 800.

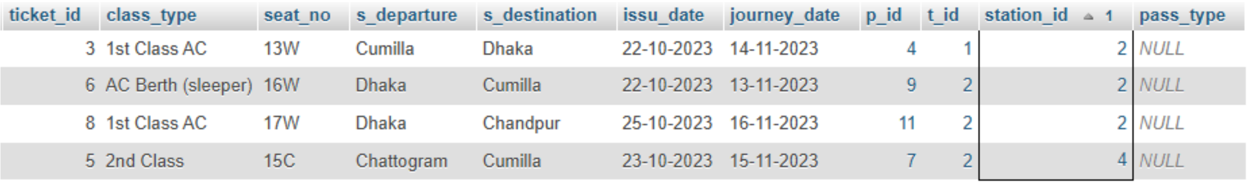
→
```
SELECT `pay_id`, `location`, `amount`, `pay_type`, `issue_date`, `issue_time`, `ticket_id`, `u_id`,
`discount`
FROM `payment`
WHERE amount between 400 and 800;
```



pay_id	location	amount	pay_type	issue_date	issue_time	ticket_id	u_id	discount
1	Dhaka-Cumilla	800	Credit Card	22-10-2023	7:30	1	21201107	0%
2	Dhaka-Cumilla	800	Cash	22-10-2023	6:30	2	21201108	0%
3	Cumilla-Dhaka	800	Cash	23-10-2023	8:30	3	21201103	0%
6	Chattogram-Cumilla	600	Debit Card	25-10-2023	7:30	5	21201107	0%
7	Dhaka-Cumilla	400	Credit Card	22-10-2023	7:00	9	21201120	50%
8	Dhaka-Cumilla	400	Cash	22-10-2023	8:00	10	21201120	50%
9	Dhaka-Cumilla	400	Cash	22-10-2023	8:30	11	21201120	50%

9. Find out the list of station_id where the station ids' are 2 and 4 and order it by ASC order from ticket table.

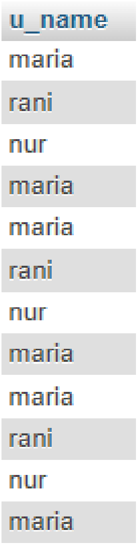
→ SELECT * FROM `ticket` WHERE station_id in (2,4) order by station_id;



ticket_id	class_type	seat_no	s_departure	s_destination	issu_date	journey_date	p_id	t_id	station_id ▲ 1	pass_type
3	1st Class AC	13W	Cumilla	Dhaka	22-10-2023	14-11-2023	4	1	2	NULL
6	AC Berth (sleeper)	16W	Dhaka	Cumilla	22-10-2023	13-11-2023	9	2	2	NULL
8	1st Class AC	17W	Dhaka	Chandpur	25-10-2023	16-11-2023	11	2	2	NULL
5	2nd Class	15C	Chattogram	Cumilla	23-10-2023	15-11-2023	7	2	4	NULL

10. Find out all the u_name where user id is same as the passesnger id.

SELECT u_name` FROM `users`,passenger WHERE users.u_id = passenger.u_id ;



u_name
maria
rani
nur
maria
maria
rani
nur
maria
maria
rani
nur
maria

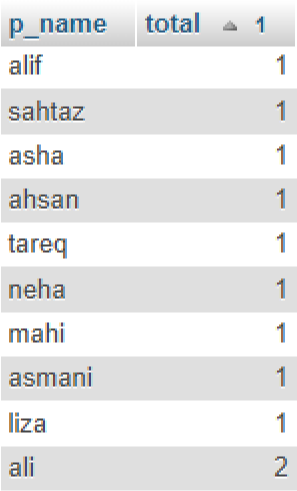
11. Find out the total name of the passengers who have the same passenger id and ticket id and also count the number of the passengers here.

SELECT `p_name`, COUNT(*) as total

FROM `passenger`,ticket

WHERE passenger.p_id = ticket.p_id

group by p_name order by total;

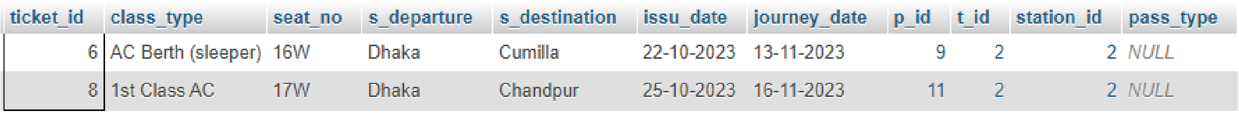


p_name	total ▲ 1
alif	1
sahtaz	1
asha	1
ahsan	1
tareq	1
neha	1
mahi	1
asmani	1
liza	1
ali	2

12. Find out the list of id of the ticket which are not in the ticket table.

→

```
SELECT * FROM ticket
WHERE ticket_id
NOT IN ( SELECT DISTINCT ticket_id FROM payment);
```

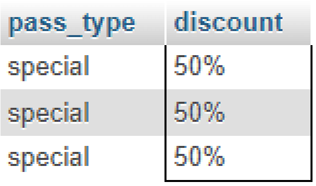


ticket_id	class_type	seat_no	s_departure	s_destination	issu_date	journey_date	p_id	t_id	station_id	pass_type
6	AC Berth (sleeper)	16W	Dhaka	Cumilla	22-10-2023	13-11-2023	9	2	2	NULL
8	1st Class AC	17W	Dhaka	Chandpur	25-10-2023	16-11-2023	11	2	2	NULL

13. Find the special passenger from ticket table whose discount is 50%.

→

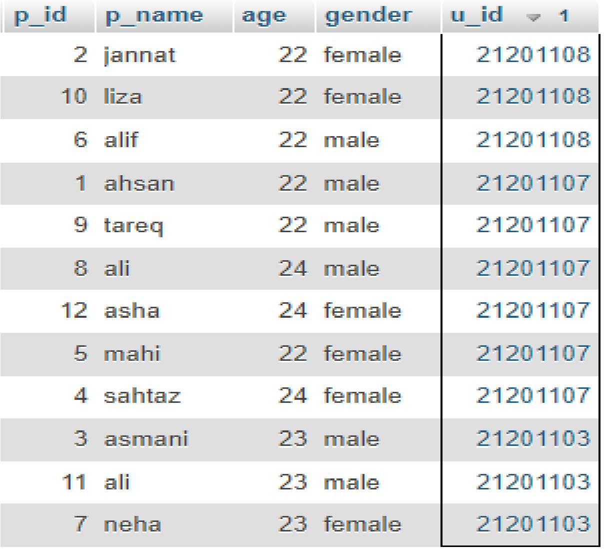
```
SELECT `pass_type`, payment.discount
FROM `ticket`,payment
WHERE ticket.ticket_id = payment.ticket_id
and discount='50%' ;
```



pass_type	discount
special	50%
special	50%
special	50%

14. Find the u_id from passenger table in descender order.

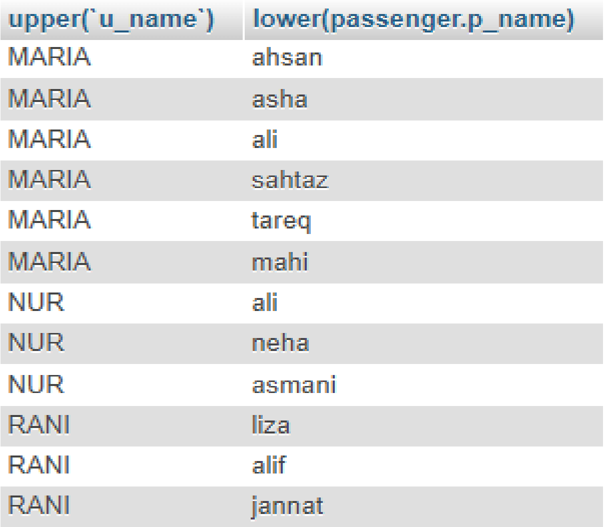
```
→ SELECT `p_id`, `p_name`, `age`, `gender`, `u_id`
   FROM `passenger`
  WHERE u_id order by u_id desc ;
```



p_id	p_name	age	gender	u_id ▼ 1
2	jannat	22	female	21201108
10	liza	22	female	21201108
6	alif	22	male	21201108
1	ahsan	22	male	21201107
9	tareq	22	male	21201107
8	ali	24	male	21201107
12	asha	24	female	21201107
5	mahi	22	female	21201107
4	sahtaz	24	female	21201107
3	asmani	23	male	21201103
11	ali	23	male	21201103
7	neha	23	female	21201103

15. Show the upper user names from user table and lower passenger names from passenger table.

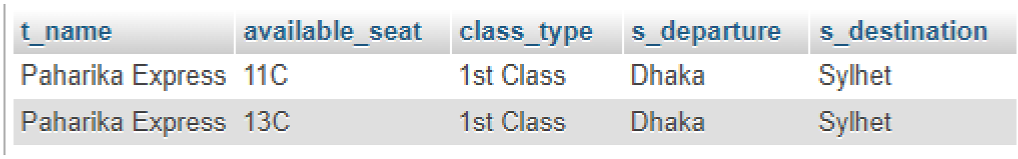
```
→ SELECT upper(`u_name`), lower(passenger.p_name)
  FROM `users`,passenger
  WHERE users.u_id = passenger.u_id order by users.u_name asc;
```



upper(`u_name`)	lower(passenger.p_name)
MARIA	ahsan
MARIA	asha
MARIA	ali
MARIA	sahtaz
MARIA	tareq
MARIA	mahi
NUR	ali
NUR	neha
NUR	asmani
RANI	liza
RANI	alif
RANI	jannat

16. Show the train status of train ID two.

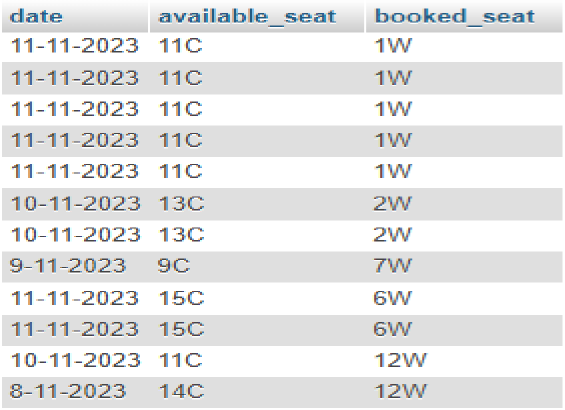
```
→ SELECT  `t_name`, train_status.available_seat, `class_type`, `s_departure`, `s_destination`
  FROM `train`,train_status
  WHERE train.t_id = train_status.t_id and train.t_id=2 ;
```



t_name	available_seat	class_type	s_departure	s_destination
Paharika Express	11C	1st Class	Dhaka	Sylhet
Paharika Express	13C	1st Class	Dhaka	Sylhet

17. Find out the list of the train seats which are booked , available and the dates including them.

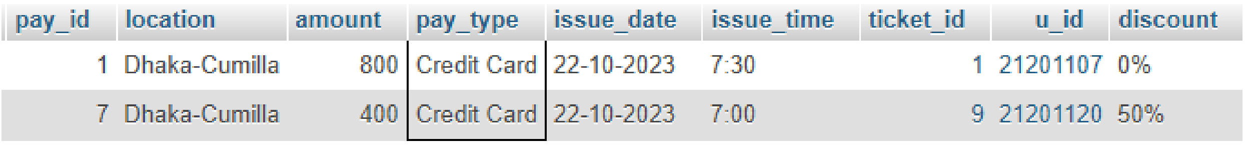
```
 SELECT `date`, `available_seat`, `booked_seat`

FROM `train_status`,passenger

WHERE train_status.status_id = passenger.status_id ;
```



date	available_seat	booked_seat
11-11-2023	11C	1W
11-11-2023	11C	1W
11-11-2023	11C	1W
11-11-2023	11C	1W
11-11-2023	11C	1W
10-11-2023	13C	2W
10-11-2023	13C	2W
9-11-2023	9C	7W
11-11-2023	15C	6W
11-11-2023	15C	6W
10-11-2023	11C	12W
8-11-2023	14C	12W

18. Find out the list of pay_id where pay type starts from 'C' and ends with 'd' from payment table.

```
→ SELECT `pay_id`, `location`, `amount`, `pay_type`, `issue_date`, `issue_time`, `ticket_id`, `u_id`,
  `discount`
  FROM `payment`
  WHERE pay_type like '%d' and pay_type like 'c%';
```

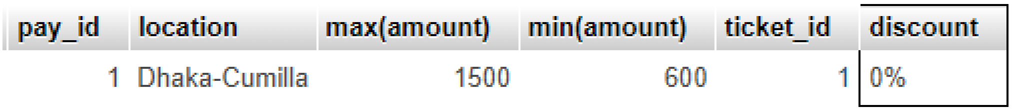


pay_id	location	amount	pay_type	issue_date	issue_time	ticket_id	u_id	discount
1	Dhaka-Cumilla	800	Credit Card	22-10-2023	7:30	1	21201107	0%
7	Dhaka-Cumilla	400	Credit Card	22-10-2023	7:00	9	21201120	50%

19. When discount is not 50%, find out the maximum and minimum amount spent while buying ticket from payment table.

→

```
SELECT `pay_id`, `location`, max(amount), min(amount), `ticket_id`,  `discount`
FROM `payment`
WHERE discount <> '50%' ;
```

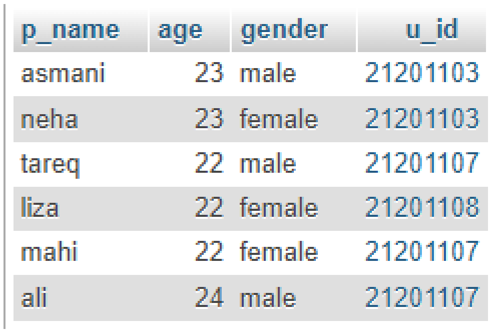


pay_id	location	max(amount)	min(amount)	ticket_id	discount
1	Dhaka-Cumilla	1500	600	1	0%

20. Show the list after cancellation of the ticket.

→

```
SELECT `p_name`, `age`, `gender`, `u_id`
FROM `passenger`,ticket
WHERE passenger.p_id = ticket.p_id and not ticket.ticket_id in (1,3,4,8,11) order by
ticket.ticket_id ;
```

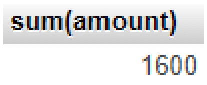


p_name	age	gender	u_id
asmani	23	male	21201103
neha	23	female	21201103
tareq	22	male	21201107
liza	22	female	21201108
mahi	22	female	21201107
ali	24	male	21201107

21. What is the total cost of two tickets?

→

```
SELECT sum(amount) FROM `payment` where ticket_id in(1,2) ;
```



sum(amount)
1600