

## **DBMS PROJECT - RAILWAY MANAGEMENT SYSTEM**

### **UPDATED DDL**

create table User

(user\_ID int NOT NULL primary key,

user\_password varchar(50) not null,

contact\_number int8 not null check(contact\_number>=1000000000 AND contact\_number<10000000000),

DOB date not null,

gender char(10) not null,

first\_name char(50) not null,

last\_name char(50) not null,

house\_number varchar(15) not null,

locality varchar(50) not null,

city varchar(50) not null,

state varchar(50) not null,

pincode int not null check(pincode>=100000 AND pincode<1000000),

age int not null, check(age>=0));

create table Payment

(payment\_id int NOT NULL primary key,

amount int not null, check(amount>0),

payment\_type varchar(20), check(payment\_type in ("UPI", "CASH", "BANK TRANSFER", "NET BANKING")));

create table Train

(train\_ID int NOT NULL,

```
train_type varchar(50) not null,  
price_per_km float not null, check(price_per_km>0.0),  
primary key(train_ID));
```

```
create table Station  
(station_id int NOT NULL primary key,  
station_name char(50) not null);
```

```
create table Train_distance_from  
(source_id int,  
destination_id int,  
distance int not null, check(distance>0),  
primary key(source_id, destination_id),  
foreign key(source_id) references Station(station_id),  
foreign key(destination_id) references Station(station_id));
```

```
create table Train_stops  
(train_ID int, station_ID int,  
arrival_day int not null,  
arrival_hour int not null, check(arrival_hour>=0 AND arrival_hour<24),  
arrival_minute int not null, check(arrival_minute>=0 AND arrival_minute<60),  
departure_day int not null, check(departure_day>=0 AND departure_day<=1),  
departure_hour int not null, check(departure_hour>=0 AND departure_hour<24),  
departure_minute int not null, check(departure_minute>=0 AND departure_minute<60),  
primary key(train_id, station_id),  
foreign key(train_ID) references Train(train_ID),  
foreign key(station_id) references Station(station_id));
```

```
create table train_classes
(train_id int, class_type varchar(20),
count_of_seats int not null, check(count_of_seats>0),
price_multiplier int not null, check(price_multiplier>0),
primary key(train_id, class_type),
foreign key(train_id) references Train(train_id));
```

```
create table Train_booking
(train_id int , source_id int, destination_id int, user_id int, ticket_id int, payment_id int, train_date date,
train_class varchar(20),
primary key(train_id, source_id, destination_id, ticket_id, user_id, train_date, train_class),
foreign key(train_id) references Train(train_ID),
foreign key(source_id) references Station(station_ID),
foreign key(destination_ID) references Station(station_ID),
foreign key(user_ID) references User(user_ID),
foreign key(payment_id) references Payment(payment_id),
foreign key(train_id, train_class) references train_classes(train_id, class_type));
```

```
create table Bus
(bus_ID int NOT NULL primary key,
bus_name varchar(50) not null,
count_of_seats int not null, check(count_of_seats>0),
bus_type varchar(50) not null,
price_per_km int not null, check(price_per_km>0));
```

```
create table Bus_stop
(bus_stop_id int NOT NULL primary key,
bus_stop_name char(50) not null) ENGINE=InnoDB;
```

```
create table Bus_stops_at
(bus_ID int, bus_stop_ID int,
arrival_day int not null, check(arrival_day>=0 AND arrival_day<=1),
arrival_hour int not null, check(arrival_hour>=0 AND arrival_hour<24),
arrival_minute int not null, check(arrival_minute>=0 AND arrival_minute<60),
departure_day int not null, check(departure_day>=0 AND departure_day<=1),
departure_hour int not null, check(departure_hour>=0 AND departure_hour<24),
departure_minute int not null, check(departure_minute>=0 AND departure_minute<60),
primary key(bus_ID, bus_stop_ID),
foreign key(bus_stop_id) references Bus_stop(bus_stop_id),
foreign key(bus_id) references Bus(bus_id));
```

```
create table Bus_distance_from
(source_id int,
destination_id int,
distance int not null, check(distance>0),
primary key(source_id, destination_id),
foreign key(source_id) references Bus_stop(bus_stop_id),
foreign key(destination_id) references Bus_stop(bus_stop_id)) ENGINE=InnoDB;
```

```
create table Bus_booking
(bus_id int , source_id int, destination_id int, user_id int, ticket_id int, payment_id int, bus_date date,
primary key(bus_id, source_id, destination_id, ticket_id, user_id, bus_date),
foreign key(bus_id) references Bus(bus_ID),
foreign key(source_id) references Bus_stop(bus_stop_ID),
foreign key(destination_ID) references Bus_stop(bus_stop_ID),
foreign key(user_ID) references User(user_ID),
foreign key(payment_id) references Payment(payment_id));
```

```
create table Airport
(airport_ID int NOT NULL primary key,
airport_name char(50) not null);
```

```
create table Flight
(flight_ID int, flight_name varchar(30), src_ID int not null, dest_ID int not null,
base_price int not null, check(base_price>0),
departure_hour int not null, check(departure_hour>=0 AND departure_hour<24),
departure_minute int not null, check(departure_minute>=0 AND departure_minute<60),
arrival_hour int not null, check(arrival_hour>=0 AND arrival_hour<24),
arrival_minute int not null, check(arrival_minute>=0 AND arrival_minute<60),
primary key(flight_ID),
foreign key(src_ID) references Airport(airport_id),
foreign key(dest_ID) references Airport(airport_id));
```

```
create table flight_classes
(flight_id int, class_type varchar(20),
count_of_seats int not null, check(count_of_seats>0),
price_multiplier int not null, check(price_multiplier>0),
primary key(flight_id, class_type),
foreign key(flight_id) references Flight(flight_id)) ENGINE=InnoDB;
```

```
create table Flight_booking
(flight_id int, source_id int, destination_id int, user_id int, ticket_id int, payment_id int, flight_date date,
flight_class varchar(20),
primary key(flight_id, source_id, destination_id, user_id, ticket_id, payment_id, flight_date, flight_class),
foreign key(flight_id) references Flight(flight_id),
foreign key(source_id) references Airport(airport_id),
```

```
foreign key(destination_id) references Airport(airport_id),  
foreign key(user_id) references User(user_id),  
foreign key(payment_id) references Payment(payment_id),  
foreign key(flight_id, flight_class) references flight_classes(flight_id, class_type));
```

```
create table Train_cancelled  
(ticket_id int primary key,  
train_id int not null,  
source_id int not null,  
destination_id int not null,  
user_id int not null,  
payment_id int not null,  
train_date date not null,  
train_class varchar(20) not null,  
foreign key(train_id) references Train(train_id),  
foreign key(source_id) references Station(station_id),  
foreign key(destination_id) references Station(station_id),  
foreign key(user_id) references User(user_id),  
foreign key(payment_id) references Payment(payment_id),  
foreign key(train_id, train_class) references train_classes(train_id, class_type));  
  
create Table Bus_cancelled  
(ticket_id int primary key,  
bus_id int not null,  
source_id int not null,  
destination_id int not null,  
user_id int not null,  
payment_id int not null,  
bus_date date not null,
```

```
foreign key(bus_id) references Bus(bus_id),  
foreign key(source_id) references Bus_stop(bus_stop_id),  
foreign key(destination_id) references Bus_stop(bus_stop_id),  
foreign key(user_id) references User(user_id),  
foreign key(payment_id) references Payment(payment_id));
```

```
create table Flight_cancelled  
(ticket_id int primary key,  
flight_id int not null,  
source_id int not null,  
destination_id int not null,  
user_id int not null,  
payment_id int not null,  
flight_date date not null,  
flight_class varchar(20) not null,  
foreign key(flight_id) references Flight(flight_id),  
foreign key(source_id) references Airport(airport_id),  
foreign key(destination_id) references Airport(airport_id),  
foreign key(user_id) references User(user_id),  
foreign key(payment_id) references Payment(payment_id),  
foreign key(flight_id, flight_class) references flight_classes(flight_id, class_type));
```

## QUERIES

1. Printing the available transport ID, price, timings for the following selected parameters with available seats for TRAIN
  - a. Mode of transport
  - b. Source and Destination
  - c. Date of travel
  - d. Class

```
select t.train_id, t.train_type, ts1.departure_hour, ts1.departure_minute,
ts2.arrival_hour, ts2.arrival_minute, ts2.departure_hour,
ts2.departure_minute, (t.price_per_km*tdf.distance*tc.price_multiplier)

from Train t, train_classes tc, Train_distance_from tdf, Train_stops
ts1, Train_stops ts2
where t.train_id=tc.train_id AND tc.class_type='{0}' AND
count_of(tc.train_id, tc.class_type, '{1}')<tc.count_of_seats AND
((tdf.source_id='{2}'
AND tdf.destination_id='{3}') OR (tdf.source_id='{3}' AND
tdf.destination_id='{2}')) AND ts1.train_id=t.train_id AND
ts2.train_id=t.train_id AND ts1.station_id='{2}' AND ts2.station_id='{3}'
AND ((ts1.departure_day<ts2.arrival_day ) OR
(ts1.departure_day=ts2.arrival_day AND ts1.departure_hour<ts2.arrival_hour)
OR (ts1.departure_day=ts2.arrival_day AND
ts1.departure_hour=ts2.arrival_hour AND
ts1.departure_minute<ts2.arrival_minute));
```

2. Printing the available transport ID, price, timings for the following selected parameters with available seats for BUS
  - a. Mode of transport
  - b. Source and Destination
  - c. Date of travel
  - d. Class

```
select b.bus_id, b.bus_type, bs1.departure_hour, bs1.departure_minute,
bs2.arrival_hour, bs2.arrival_minute, bs2.departure_hour,
bs2.departure_minute, (b.price_per_km*bdf.distance)
from Bus b, Bus_distance_from bdf, Bus_stops_at bs1, Bus_stops_at bs2
where count_of_bus(b.bus_id, '{0}')<b.count_of_seats AND
((bdf.source_id='{1}' AND bdf.destination_id='{2}') OR (bdf.source_id='{2}'
AND bdf.destination_id='{1}'))
AND bs1.bus_id=b.bus_id AND bs2.bus_id=b.bus_id AND bs1.bus_stop_id='{1}'
AND bs2.bus_stop_id='{2}' AND ((bs1.departure_day<bs2.arrival_day ) OR
(bs1.departure_day=bs2.arrival_day AND bs1.departure_hour<bs2.arrival_hour))
```



OR (bs1.departure\_day=bs2.arrival\_day AND bs1.arrival\_hour=bs2.arrival\_hour  
AND bs1.departure\_minute<bs2.arrival\_minute));

**3. Printing the available transport ID, price, timings for the following selected parameters with available seats for FLIGHT**

- a. Mode of transport
- b. Source and Destination
- c. Date of travel
- d. Class

```
select f.flight_ID, f.flight_name, f.departure_hour, f.departure_minute,
f.arrival_hour, f.arrival_minute, (f.base_price*fc.price_multiplier)
from Flight f, flight_classes fc
where f.flight_ID=fc.flight_id AND f.src_ID='{0}' AND f.dest_id='{1}' AND
count_of_flight(f.flight_id, fc.class_type, '{2}')<fc.count_of_seats AND
fc.class_type='{3}';
```

**4. Printing the booked ticket details by taking ticketID that have not yet been canceled for TRAIN-**

```
select tb.ticket_id, t.train_id, t.train_type, tb.train_class, tb.source_id,
ts1.departure_hour, ts1.departure_minute, tb.destination_id,
ts2.arrival_hour, ts2.arrival_minute, tb.train_date, p.amount
    from Train t, Train_booking tb, Train_stops ts1, Train_stops ts2,
Payment p
    where t.train_id=tb.train_id AND ts1.train_id=t.train_id AND
ts2.train_id=t.train_id AND tb.payment_id=p.payment_id AND
tb.ticket_id='{0}' AND
    tb.source_id=ts1.station_id AND tb.destination_id=ts2.station_id;
```

**5. Printing the booked ticket details by taking ticketID that have not yet been canceled for BUS-**

```
select bb.ticket_id, b.bus_id, b.bus_type, bb.source_id, bs1.departure_hour,
bs1.departure_minute, bb.destination_id, bs2.arrival_hour,
bs2.arrival_minute, bb.bus_date, p.amount
    from Bus b, Bus_booking bb, Bus_stops_at bs1, Bus_stops_at bs2, Payment
p
    where b.bus_id=bb.bus_id AND bs1.bus_id=b.bus_id AND bs2.bus_id=b.bus_id
AND bb.payment_id=p.payment_id AND bb.ticket_id='{0}' AND
    bb.source_id=bs1.bus_stop_ID AND bb.destination_id=bs2.bus_stop_id;
```

**6. Printing the booked ticket details by taking ticketID that have not yet been canceled for FLIGHT-**

```

select fb.ticket_id, f.flight_ID, f.flight_name, fc.class_type, f.src_id,
f.departure_hour, f.departure_minute, f.dest_id, f.arrival_hour,
f.arrival_minute, fb.flight_date, p.amount
    from Flight f, Flight_booking fb, Payment p, flight_classes fc
    where f.flight_ID=fb.flight_id AND fb.payment_id=p.payment_id AND
fb.ticket_id='{0}' AND f.flight_id=fc.flight_id AND
fb.flight_class=fc.class_type;

```

**7. Show all kids(<18 age) traveling on the bus for a particular date.**

```

Select User.user_ID, User.first_name, User.last_name, Bus_booking.ticket_id,
Bus_booking.payment_id,
Payment.amount from User, Bus_booking, Payment where
Bus_booking.user_ID = User.user_ID and Bus_booking.payment_id =
Payment.payment_id and Bus_booking.bus_ID = 2021 and
Bus_booking.source_id = 203 and Bus_booking.destination_id= 206 AND
Bus_booking.bus_date='2022-04-19';

```

**8. Given a bus, and 2 bus stops, print details of all those users traveling between those 2 stops along with their payment details.**

```

Select User.user_ID, User.first_name, User.last_name, Bus_booking.ticket_id,
Bus_booking.payment_id,
Payment.amount from User, Bus_booking, Payment where
Bus_booking.user_ID = User.user_ID and Bus_booking.payment_id =
Payment.payment_id and Bus_booking.bus_ID = 2021 and
Bus_booking.source_id = 203 and Bus_booking.destination_id= 206 AND
Bus_booking.bus_date='2022-04-19';

```

**9. User inputs time of arrival and station, print all trains traveling away from that station and having departure time after the arrival time of user (Eg: station-102, arrival time-05:10)**

```

Select Train.train_ID, Train.train_type from Train inner join Train_stops on
Train.train_ID = Train_stops.train_ID where Train_stops.station_ID=102 and
trainEnding(Train.train_ID)!=102
and ((Train_stops.arrival_hour>5) OR (Train_stops.arrival_hour=5 AND
Train_stops.arrival_minute>=10));

```

**10. Given a train and a date, give all the bookings that have been done.**

```

select tb.train_id, tb.train_date, sum(p.amount), p.payment_type
from Train_booking tb join Payment p on tb.payment_id=p.payment_id
where tb.train_id='{}' AND tb.train_date='{}'
group by p.payment_type;

```

**11. Given a Bus and a date, give all the bookings that have been done.**

```

select bb.bus_id, bb.bus_date, sum(p.amount), p.payment_type

```

```

from Bus_booking bb join Payment p on bb.payment_id=p.payment_id
where bb.bus_id='{0}' AND bb.bus_date='{1}'
group by p.payment_type;

```

## 12. Given a Flight and a date, give all the bookings that have been done.

```

select fb.flight_id, fb.flight_date, sum(p.amount), p.payment_type
from Flight_booking fb join Payment p on fb.payment_id=p.payment_id
where fb.flight_id='{0}' AND fb.flight_date='{1}'
group by p.payment_type;

```

## 13. Give all the bookings with given source, destination and travel date for train

```

select tb.ticket_id, tb.train_id, tb.user_id, tb.train_class, tb.payment_id,
p.amount
from Train_booking tb join Payment p on tb.payment_id=p.payment_id
where tb.train_date='{0}' AND tb.source_id='{1}' AND tb.destination_id='{2}'

```

## 14. Give all the bookings with given source, destination and travel date for bus

```

select bb.ticket_id, bb.bus_id, bb.user_id, bb.payment_id, p.amount
from Bus_booking bb join Payment p on bb.payment_id=p.payment_id
where bb.bus_date='{0}' AND bb.source_id='{1}' AND bb.destination_id='{2}';

```

## 15. Give all the bookings with given source, destination and travel date for flight

```

select fb.ticket_id, fb.flight_id, fb.user_id, fb.flight_class,
fb.payment_id, p.amount
from Flight_booking fb join Payment p on fb.payment_id=p.payment_id
where fb.flight_date='{0}' AND fb.source_id='{1}' AND
fb.destination_id='{2}';

```

# GRANTS AND REVOKES

GRANT SELECT, DELETE

ON User

TO 'Admin'@'localhost';

REVOKE INSERT, UPDATE, ALTER, DROP

ON User

TO 'Admin'@'localhost';

GRANT SELECT, INSERT

ON User

TO 'Client'@'localhost';

REVOKE DELETE, UPDATE, ALTER, DROP

ON User

TO 'Client'@'localhost';

GRANT SELECT, DELETE, INSERT

ON Payment

TO 'Admin'@'localhost';

REVOKE UPDATE, ALTER, DROP

ON Payment

TO 'Admin'@'localhost';

GRANT SELECT, DELETE, INSERT

ON Payment

TO 'Client'@'localhost';

REVOKE UPDATE, ALTER, DROP  
ON Payment  
TO 'Client'@'localhost';

GRANT SELECT, UPDATE  
ON Train  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, ALTER, DROP  
ON Train  
TO 'Admin'@'localhost';

GRANT SELECT  
ON Train\_stops  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Train\_stops  
TO 'Admin'@'localhost';

GRANT SELECT  
ON Station  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Station  
TO 'Admin'@'localhost';

GRANT SELECT  
ON Train\_distance\_from  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Train\_distance\_from  
TO 'Admin'@'localhost';

GRANT SELECT, UPDATE  
ON Train\_classes  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, ALTER, DROP  
ON Train\_classes

TO 'Admin'@'localhost';

GRANT SELECT, DELETE  
ON Train\_booking  
TO 'Admin'@'localhost';

REVOKE INSERT, UPDATE, ALTER, DROP  
ON Train\_booking  
TO 'Admin'@'localhost';

GRANT SELECT, UPDATE  
ON Bus  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, ALTER, DROP  
ON Bus  
TO 'Admin'@'localhost';

GRANT SELECT  
ON Bus\_stop  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Bus\_stop  
TO 'Admin'@'localhost';

GRANT SELECT  
ON Bus\_stops\_at  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Bus\_stops\_at  
TO 'Admin'@'localhost';

GRANT SELECT  
ON Bus\_distance\_from  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Bus\_distance\_from  
TO 'Admin'@'localhost';

GRANT SELECT, DELETE  
ON Bus\_booking

TO 'Admin'@'localhost';

REVOKE INSERT, UPDATE, ALTER, DROP  
ON Bus\_booking  
TO 'Admin'@'localhost';

GRANT SELECT  
ON Airport  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Airport  
TO 'Admin'@'localhost';

GRANT SELECT, UPDATE  
ON Flight  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, ALTER, DROP  
ON Flight  
TO 'Admin'@'localhost';

GRANT SELECT, UPDATE  
ON Flight\_classes  
TO 'Admin'@'localhost';

REVOKE DELETE, INSERT, ALTER, DROP  
ON Flight\_classes  
TO 'Admin'@'localhost';

GRANT SELECT, DELETE  
ON Flight\_booking  
TO 'Admin'@'localhost';

REVOKE INSERT, UPDATE, ALTER, DROP  
ON Flight\_booking  
TO 'Admin'@'localhost';

GRANT SELECT  
ON Train  
TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Train  
TO 'Client'@'localhost';

GRANT SELECT  
ON Train\_stops  
TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Train\_stops  
TO 'Client'@'localhost';

GRANT SELECT  
ON Station  
TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Station  
TO 'Client'@'localhost';

GRANT SELECT  
ON Train\_distance\_from  
TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Train\_distance\_from  
TO 'Client'@'localhost';

GRANT SELECT  
ON Train\_classes  
TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP  
ON Train\_classes  
TO 'Client'@'localhost';

GRANT SELECT, INSERT, DELETE  
ON Train\_booking  
TO 'Client'@'localhost';

REVOKE UPDATE, ALTER, DROP  
ON Train\_booking  
TO 'Client'@'localhost';

GRANT SELECT  
ON Bus

TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP

ON Bus

TO 'Client'@'localhost';

GRANT SELECT

ON Bus\_stop

TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP

ON Bus\_stop

TO 'Client'@'localhost';

GRANT SELECT

ON Bus\_stops\_at

TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP

ON Bus\_stops\_at

TO 'Client'@'localhost';

GRANT SELECT

ON Bus\_distance\_from

TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP

ON Bus\_distance\_from

TO 'Client'@'localhost';

GRANT SELECT, INSERT, DELETE

ON Bus\_booking

TO 'Client'@'localhost';

REVOKE UPDATE, ALTER, DROP

ON Bus\_booking

TO 'Client'@'localhost';

GRANT SELECT

ON Airport

TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP

ON Airport

TO 'Client'@'localhost';

GRANT SELECT

ON Flight

TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP

ON Flight

TO 'Client'@'localhost';

GRANT SELECT

ON Flight\_classes

TO 'Client'@'localhost';

REVOKE DELETE, INSERT, UPDATE, ALTER,  
DROP

ON Flight\_classes

TO 'Client'@'localhost';

GRANT SELECT, INSERT, DELETE

ON Flight\_booking

TO 'Client'@'localhost';

REVOKE UPDATE, ALTER, DROP

ON Flight\_booking

TO 'Client'@'localhost';