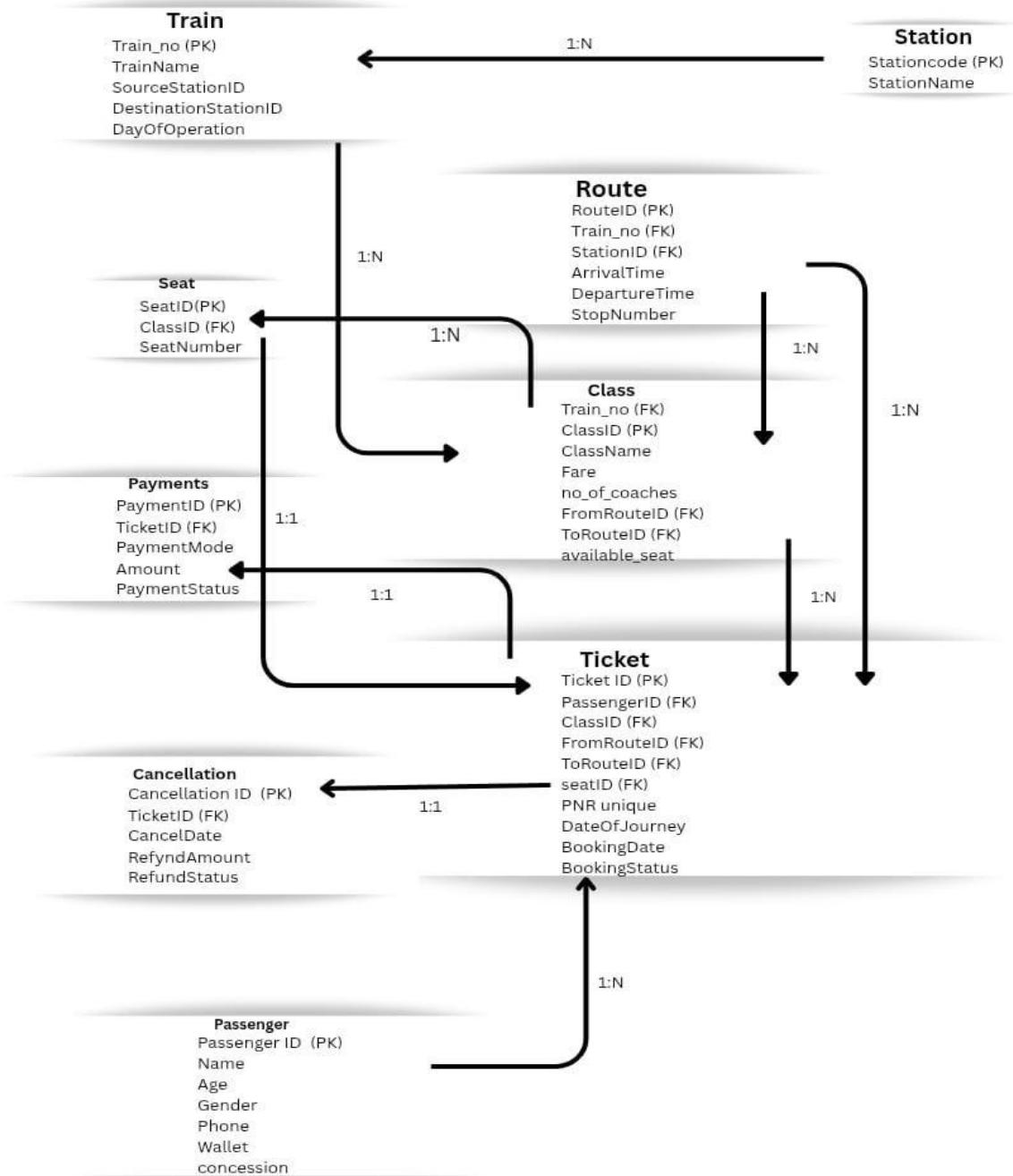


# E-R Diagram



## Relational Schema

The relational schema of the Indian Railway Ticket Reservation System is designed using a normalized structure to efficiently manage data and eliminate redundancy. It consists of multiple interrelated tables including Station, Train, Route, Class, Passenger, Ticket, Seat, Payment, and Cancellation. Each table has a defined primary key ensuring entity integrity, while foreign key constraints enforce referential integrity across relationships. The schema captures one-to-many (1:N) relationships such as a Train having multiple Classes and a Passenger having multiple Tickets, as well as one-to-one (1:1) relationships like a Ticket linked to a single Payment or Cancellation record. This structured approach ensures scalability, consistency, and optimized query performance in handling complex reservation workflows.

## Normalization Process for Each Table

### 1. Station(Stationcode, StationName)

1NF:

- All values are atomic (e.g., no multi-valued StationNames or Locations).

2NF:

- Primary Key: Stationcode
- No partial dependencies since all attributes depend on the whole primary key.

3NF:

- No transitive dependencies.
- Location and StationName are fully dependent only on Stationcode.

Station is in 3NF

### 2. Train(Train\_no, TrainName, SourceStationID, DestinationStationID, DaysOfOperation)

1NF:

- All attribute values are atomic.

2NF:

- Primary Key: Train\_no
- All attributes (TrainName, SourceStationID, DestinationStationID, DaysOfOperation) are dependent on TrainID.

3NF:

- No transitive dependencies exist.

Train is in 3NF

### 3. Route(RouteID, Train\_no, StationID, ArrivalTime, DepartureTime, StopNumber)

1NF:

- No repeating groups, all atomic fields.

2NF:

- RouteID is the primary key.
- All other attributes (StationID, StopNumber, ArrivalTime) depend on the key.

3NF:

- No transitive dependencies — e.g., StopNumber is dependent on the Route itself.

Route is in 3NF

### 4. Class(ClassID, Train\_no, ClassName, Fare, no\_of\_coaches, FromRouteID, ToRouteID, available\_seats)

1NF:

- All fields are atomic.

2NF:

- ClassID is the primary key.
- All attributes depend entirely on ClassID.

3NF:

- No transitive dependencies.
- Fare depends only on the ClassID (not indirectly via TrainID).

Class is in 3NF

### 5. Ticket(TicketID, PaymentID, ClassID, PassengerID, BookingDate, FromRouteID, ToRouteID, SeatID, BookingStatus, DateOfJourney, PNR)

1NF:

- Each row stores one ticket — no multi-valued fields.

2NF:

- Primary Key: TicketID
- All non-key fields depend entirely on TicketID.

3NF:

- No transitive dependencies.
- Everything is directly stored only in context of TicketID.

Ticket is in 3NF

## **6. Payment(PaymentID, TicketID, PaymentStatus, Amount, PaymentMode)**

1NF:

- Atomic values for each payment.

2NF:

- Primary Key: PaymentID
- All other fields are fully functionally dependent on it.

3NF:

- No transitive dependencies exist between PaymentMode or Amount and any non-key fields.

Payment is in 3NF

## **7. Cancellation(CancellationID, TicketID, CancelDate)**

1NF:

- Each cancellation is atomic per row.

2NF:

- Primary Key: CancellationID - All fields are fully dependent on it.

3NF:

- No transitive dependencies exist.

- RefundAmount depends only on the cancellation, not transitively on TicketID.

Cancellation is in 3NF

## **8. Seat(SeatID, ClassID, SeatNumber)**

1NF:

- Each seat record is atomic.

2NF:

- Primary Key: SeatID
- fields are fully dependent on it.

3NF:

- No transitive dependencies — availability status is directly tied to seat context.

Seat is in 3NF

## **9. Passenger(PassengerID, Age, Gender, Phone, Wallet, concession)**

1NF:

- Each cancellation is atomic per row.

2NF:

- Primary Key: PassengerID
- All fields are fully dependent on it.

3NF:

- No transitive dependencies exist.

Passenger is in 3NF

Sample Data

	SeatID	ClassID	SeatNumber
1	1	H1-01	
2	1	H1-02	
3	1	H1-03	
4	1	H1-04	
5	1	H1-05	
6	1	H1-06	
7	1	H1-07	
8	1	H1-08	
9	1	H1-09	
10	1	H1-10	
11	1	H1-11	
12	1	H1-12	
13	1	H1-13	
14	1	H1-14	
15	1	H1-15	
Seat 8			

Result Grid | Filter Rows:

PaymentID	TicketID	PaymentMode	Amount	PaymentStatus
1	3	UPI	1920.00	Refunded
2	4	UPI	2125.00	Success
3	5	UPI	2125.00	Success
4	6	UPI	2125.00	Success
5	7	UPI	3000.00	Success
6	8	UPI	3000.00	Success
7	9	UPI	1750.00	Success
8	10	UPI	2125.00	Success
9	11	UPI	2125.00	Success
10	12	UPI	3000.00	Success
11	13	UPI	3000.00	Success
12	14	UPI	2125.00	Success
13	15	UPI	3000.00	Success
14	16	UPI	2125.00	Success
15	17	UPI	3000.00	Success

Payment 7

Output

Result Grid | Filter Rows:

PassengerID	Name	Age	Gender	Phone	Wallet	concession
1	Ayaan	36	M	+916410255894	2456.36	
2	Rahul	46	M	+918594519537	4938.18	
3	Ayaan	53	M	+917109608039	5070.37	Disabled
4	Ishita	54	F	+918540114111	5151.56	Ladies
5	Vanya	51	F	+916126657693	5393.80	Ladies
6	Kabir	58	M	+916781321572	5465.33	Disabled
7	Yash	58	M	+918179162183	4599.25	
8	Dev	29	M	+918061630255	4916.65	
9	Manav	57	M	+919344363591	4587.86	
10	Ansh	47	M	+918638757879	4924.28	Student
11	Diya	47	F	+919260166217	4749.25	Ladies
12	Vanya	54	F	+917359469041	4643.93	Ladies
13	Tara	51	F	+918372144775	4704.51	Ladies
14	Navya	37	F	+916232874432	5228.88	Ladies
15	Ruhan	18	M	+919980701249	4850.66	

Passenger 6

Result Grid | Filter Rows: | Edit: | Export/Import:

	RouteID	Train_no	StationID	ArrivalTime	DepartureTime	StopNumber
1	12245	HWH	NULL	11:00:00	1	
2	12245	BZA	23:35:00	23:40:00	2	
3	12245	YPR	20:00:00	NULL	3	
4	12301	HWH	00:00:00	16:50:00	1	
5	12301	DHN	20:33:00	20:38:00	2	
6	12301	GAYA	22:43:00	22:48:00	3	
7	12301	DDU	01:40:00	01:50:00	4	
8	12301	PRYJ	03:25:00	03:27:00	5	
9	12301	CNB	06:00:00	06:05:00	6	
10	12301	NDLS	10:05:00	00:00:00	7	
11	12423	DBRG	00:00:00	20:55:00	1	
12	12423	LMG	22:10:00	22:15:00	2	
13	12423	GHY	00:55:00	01:05:00	3	
14	12423	NJP	05:40:00	05:45:00	4	
15	12423	KTR	08:55:00	09:00:00	5	

Route 5 ×

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	Train_no	ClassID	ClassName	Fare	no_of_coaches	FromRouteID	ToRouteID	available_seats
12245	1	1A	4200	1	1	3	18	
12245	2	2A	2500	3	1	3	54	
12245	3	3A	1800	6	1	3	108	
12301	4	1A	4350	1	4	10	18	
12301	5	2A	2550	3	4	10	54	
12301	6	3A	1800	6	4	10	108	
12423	7	1A	4550	1	11	18	18	
12423	8	2A	2600	3	11	18	54	
12423	9	3A	1850	6	11	18	108	
12723	10	2A	2700	2	37	54	48	
12723	11	3A	1850	5	37	54	90	
12723	12	SL	750	10	37	54	180	
12548	13	3A	1600	4	55	56	72	
12548	14	SL	600	12	55	56	144	
12615	15	1A	4100	1	57	58	18	

Class 4 ×

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	Train_no	TrainName	SourceStationID	DestinationStationID	DaysOfOperation
1920	Asarva - Agra Cantt. Special Fare Summer Special	ASV	AGC		Sun, Mon, Tue, Wed, Thu, Fri, Sat
12245	Howrah Yesvantpur Duronto	HWH	YPR		Mon,Wed,Fri
12301	Kolkata Rajdhani Express	HWH	NDLS		Mon,Tue,Wed,Thu,Fri,Sat,Sun
12423	Rajdhani Express	DBRG	NDLS		Mon,Tue,Thu,Sat
12548	Sabarmati - Agra Cantt. SF Express	SBIB	AGC		Sun, Mon, Wed, Thu
12615	Grand Trunk Express	MAS	NDLS		Mon,Tue,Wed,Thu,Fri,Sat,Sun
12627	Karnataka Express	SBC	NDLS		Mon,Tue,Wed,Thu,Fri,Sat,Sun
12723	Telangana Express	SC	NDLS		Mon,Tue,Wed,Thu,Fri,Sat,Sun
12801	Puri New Delhi Express	PURI	NDLS		Mon,Wed,Fri,Sun
12877	Ranji New Delhi Garib Rath	RNC	NDLS		Mon,Wed,Fri
12937	Gandhidham - Howrah Garba Express	GIMB	HWH		Fri
12951	Mumbai Rajdhani Express	BCT	NDLS		Mon,Tue,Wed,Thu,Fri,Sat,Sun
15046	Okha - Gorakhpur Express	OKHA	GKP		Mon
15669	Nagaland Express	GHY	DLI		Tue,Thu,Sat
		HULL	HULL		HULL

Train 3 ×

Stationcode	StationName
BBS	Bhubaneswar
BCT	Mumbai Central
BPA	Belampalli
BPL	Bhopal Junction
BPQ	Balharshah
BSB	Varanasi Junction
BVC	Bhavnagar Terminus
BVH	Ballabh Garh
BZA	Vijayawada Junction
CD	Chandrapur
CDG	Chandigarh Junction
CNB	Kanpur Central
CPL	Charlapalli
DBG	Darbhanga Junction
DRRG	Dibrugarh

Station 2 X

633  
634 • call refund\_for\_cancelling\_train(12245, '2025-04-21');  
635

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Contents:

TicketID	PassengerID	ClassID	FromRouteID	ToRouteID	SeatID	PNR	DateOfJourney	BookingDate	BookingStatus
3	10	13	55	56	195	2428916017	2025-04-21	2025-04-15	Cancelled
4	62	2	2	3	48	2893967396	2025-04-21	2025-04-15	Confirmed
5	13	2	2	3	20	2169436016	2025-04-21	2025-04-15	Confirmed
6	48	2	2	3	92	3397941077	2025-04-21	2025-04-15	Confirmed
7	19	2	2	3	45	413765897	2025-04-21	2025-04-15	Confirmed
8	24	2	1	3	39	4149256445	2025-04-21	2025-04-15	Confirmed
9	35	2	1	2	42	5694193388	2025-04-21	2025-04-15	Confirmed
10	26	2	1	2	79	8150444906	2025-04-21	2025-04-15	Confirmed
11	71	2	1	3	52	5311818793	2025-04-21	2025-04-15	Confirmed
12	2	2	1	3	27	7419006706	2025-04-21	2025-04-15	Confirmed
13	25	2	1	3	63	6503903557	2025-04-21	2025-04-15	Confirmed
14	18	2	2	3	54	3060121409	2025-04-21	2025-04-15	Confirmed
15	21	2	2	3	68	1408396118	2025-04-21	2025-04-15	Confirmed
16	60	2	2	3	87	9281584779	2025-04-21	2025-04-15	Confirmed
17	56	2	1	3	43	1097782719	2025-04-21	2025-04-15	Confirmed

Ticket 1 X

Output:  Action Output:

## **SQL queries, procedures, functions, indices, triggers and Other interesting queries beyond the required ones.**

```
CREATE DATABASE railway;
```

```
use railway; create table
```

```
Station (
```

```
    Stationcode varchar(10) PRIMARY KEY,    StationName VARCHAR(100)); insert
into Station (Stationcode,StationName)values('ADI','Ahmedabad Station'),
('NDLS','New Delhi'),
('PNBE','Patna Junction'),
('AGC','Agra Cantt'),
('RNC','Ranchi Junction'),
('PUNE','Pune Junction'),
('KOTA','Kota Junction'),
('HWH','Howrah Junction'),
('VSKP','Visakhapatnam Junction'),
('SC','Secunderabad Junction'),
('NGP','Nagpur junction'),
('BSB','Varanasi Junction'),
('CDG','Chandigarh Junction'),
('GHY','Guwahati'),
('HW','Haridwar'),
('BCT', 'Mumbai Central'),
('MAS', 'Chennai Central'),
('YPR', 'Yesvantpur Junction'),
('BBS', 'Bhubaneswar'),
('DBG', 'Darbhanga Junction'),
('SBC', 'KSR Bengaluru'),
('ET', 'Itarsi Junction'),
```

```

('ALD', 'Prayagraj Junction'),
('JAT', 'Jammu Tawi');

INSERT INTO Station (Stationcode, StationName) VALUES
('PURI', 'Puri'),
('DBRG', 'Dibrugarh'),
('DLI', 'Old Delhi Junction');

CREATE TABLE Train (
    Train_no INT(5) PRIMARY KEY,
    TrainName VARCHAR(100) NOT NULL,
    SourceStationID VARCHAR(10),
    DestinationStationID VARCHAR(10),
    DaysOfOperation VARCHAR(50), -- e.g., Mon,Tue,Wed
    FOREIGN KEY (SourceStationID) REFERENCES Station(Stationcode),
    FOREIGN KEY (DestinationStationID) REFERENCES Station(Stationcode)
);

INSERT INTO Train (Train_no, TrainName, SourceStationID, DestinationStationID,
DaysOfOperation) VALUES
(12951, 'Mumbai Rajdhani Express', 'BCT', 'NDLS', 'Mon,Tue,Wed,Thu,Fri,Sat,Sun'),
(12301, 'Kolkata Rajdhani Express', 'HWH', 'NDLS', 'Mon,Tue,Wed,Thu,Fri,Sat,Sun'),
(12627, 'Karnataka Express', 'SBC', 'NDLS', 'Mon,Tue,Wed,Thu,Fri,Sat,Sun'),
(12801, 'Puri New Delhi Express', 'PURI', 'NDLS', 'Mon,Wed,Fri,Sun'),
(12723, 'Telangana Express', 'SC', 'NDLS', 'Mon,Tue,Wed,Thu,Fri,Sat,Sun'),
(12245, 'Howrah Yesvantpur Duronto', 'HWH', 'YPR', 'Mon,Wed,Fri'),
(12615, 'Grand Trunk Express', 'MAS', 'NDLS', 'Mon,Tue,Wed,Thu,Fri,Sat,Sun'),
(12877, 'Ranchi New Delhi Garib Rath', 'RNC', 'NDLS', 'Mon,Wed,Fri'),
(12423, 'Rajdhani Express', 'DBRG', 'NDLS', 'Mon,Tue,Thu,Sat'),
(15669, 'Nagaland Express', 'GHY', 'DLI', 'Tue,Thu,Sat');

INSERT INTO Station (Stationcode, StationName) VALUES

```

('SBIB', 'Sabarmati BG'),  
('OKHA', 'Okha'),  
('GKP', 'Gorakhpur Junction'),  
('ASV', 'Asarva Junction'),  
('BVC', 'Bhavnagar Terminus'),  
('ASN', 'Asansol Junction'),  
('IPR', 'Islampur'),  
('HTE', 'Hatia');

INSERT INTO Train (Train\_no, TrainName, SourceStationID, DestinationStationID, DaysOfOperation) VALUES

(12548, 'Sabarmati - Agra Cantt. SF Express', 'SBIB', 'AGC', 'Sun, Mon, Wed, Thu'),  
(15046, 'Okha - Gorakhpur Express', 'OKHA', 'GKP', 'Mon'),  
(1920, 'Asarva - Agra Cantt. Special Fare Summer Special', 'ASV', 'AGC', 'Sun, Mon, Tue, Wed, Thu, Fri, Sat');

INSERT INTO Station (Stationcode, StationName) VALUES

('LT', 'Lokmanya Tilak Terminus'),  
('SMVB', 'SMVT Bengaluru'),  
('ERS', 'Ernakulam Junction'),  
('SURAT', 'Surat Junction'),  
('SAMB', 'Sambalpur Junction'),  
('KYQ', 'Kamakhya Junction'),  
('RXL', 'Raxaul Junction'),  
('CPL', 'Charlapalli'),  
('DNR', 'Danapur'),  
('GIMB', 'Gandhidham BG');

INSERT INTO Station (Stationcode, StationName) VALUES('BZA', 'Vijayawada Junction');

INSERT INTO Train (Train\_no, TrainName, SourceStationID, DestinationStationID, DaysOfOperation) VALUES

(12937, 'Gandhidham - Howrah Garba Express', 'GIMB', 'HWH', 'Fri');

```

CREATE TABLE Route (
    RouteID INT AUTO_INCREMENT PRIMARY KEY,
    Train_no INT(5),
    StationID VARCHAR(10),
    ArrivalTime TIME,
    DepartureTime TIME,
    StopNumber INT, -- order of the station in the route
    FOREIGN KEY (Train_no) REFERENCES Train(Train_no),
    FOREIGN KEY (StationID) REFERENCES Station(Stationcode)
);

INSERT INTO Route (Train_no, StationID, ArrivalTime, DepartureTime, StopNumber) VALUES
(12245, 'HWH', NULL, '11:00:00', 1),
(12245, 'BZA', '23:35:00', '23:40:00', 2),
(12245, 'YPR', '20:00:00', NULL, 3);

INSERT INTO Station (Stationcode, StationName) VALUES
('DHN', 'Dhanbad Junction'),
('GAYA', 'Gaya Junction'),
('DDU', 'Pandit Deen Dayal Upadhyay Junction'),
('PRYJ', 'Prayagraj Junction'),
('CNB', 'Kanpur Central');

INSERT INTO Route (Train_no, StationID, ArrivalTime, DepartureTime, StopNumber) VALUES
(12301, 'HWH', '00:00:00', '16:50:00', 1),
(12301, 'DHN', '20:33:00', '20:38:00', 2),
(12301, 'GAYA', '22:43:00', '22:48:00', 3),
(12301, 'DDU', '01:40:00', '01:50:00', 4),
(12301, 'PRYJ', '03:25:00', '03:27:00', 5),
(12301, 'CNB', '06:00:00', '06:05:00', 6),
(12301, 'NDLS', '10:05:00', '00:00:00', 7);

```

-- New Stations

```
INSERT INTO Station (Stationcode, StationName) VALUES  
('LMG', 'Lumding Junction'),  
('NJP', 'New Jalpaiguri Junction'),  
('KIR', 'Katihar Junction');
```

```
INSERT INTO Route (Train_no, StationID, ArrivalTime, DepartureTime, StopNumber) VALUES  
(12423, 'DBRG', '00:00:00', '20:55:00', 1),  
(12423, 'LMG', '22:10:00', '22:15:00', 2),  
(12423, 'GHY', '00:55:00', '01:05:00', 3),  
(12423, 'NJP', '05:40:00', '05:45:00', 4),  
(12423, 'KIR', '08:55:00', '09:00:00', 5),  
(12423, 'PRYJ', '16:35:00', '16:40:00', 6),  
(12423, 'CNB', '19:15:00', '19:20:00', 7),  
(12423, 'NDLS', '04:15:00', '00:00:00', 8);
```

-- New Stations

```
INSERT INTO Station (Stationcode, StationName) VALUES  
('HYB', 'Hyderabad Deccan'),  
('KZJ', 'Kazipet Junction'),  
('RDM', 'Ramgundam'),  
('MCI', 'Manchiryal'),  
('BPA', 'Belampalli'),  
('SKZR', 'Sirpur Kagaznagar'),  
('BPQ', 'Balharshah'),  
('CD', 'Chandrapur'),  
('BPL', 'Bhopal Junction'),  
('VGLJ', 'Virangana Lakshmibai Jhansi Junction'),  
('GWL', 'Gwalior Junction');
```

```
INSERT INTO Route (Train_no, StationID, ArrivalTime, DepartureTime, StopNumber) VALUES
```

(12723, 'HYB', '06:00:00', '06:05:00', 1),  
(12723, 'SC', '06:20:00', '06:25:00', 2),  
(12723, 'KZJ', '08:00:00', '08:05:00', 3),  
(12723, 'RDM', '09:30:00', '09:35:00', 4),  
(12723, 'MCI', '09:45:00', '09:50:00', 5),  
(12723, 'BPA', '10:10:00', '10:15:00', 6),  
(12723, 'SKZR', '10:40:00', '10:45:00', 7),  
(12723, 'BPQ', '12:00:00', '12:05:00', 8),  
(12723, 'CD', '12:15:00', '12:20:00', 9),  
(12723, 'NGP', '15:20:00', '15:25:00', 10),  
(12723, 'BPL', '21:45:00', '21:55:00', 11),  
(12723, 'VGLJ', '01:16:00', '01:24:00', 12),  
(12723, 'GWL', '02:45:00', '02:50:00', 13),  
(12723, 'AGC', '04:25:00', '04:30:00', 14),  
(12723, 'MTJ', '05:05:00', '05:10:00', 15),  
(12723, 'BVH', '07:06:00', '07:08:00', 16),  
(12723, 'NZM', '07:33:00', '07:35:00', 17),  
(12723, 'NDLS', '08:00:00', '00:00:00', 18);

INSERT INTO Station VALUES('MTJ','Mathura Junction'),('BVH','ballabh Garh'),('NZM','Nizamuddin');

INSERT INTO Route (Train\_no, StationID, ArrivalTime, DepartureTime, StopNumber) VALUES  
(12548, 'ADI', '00:00:00', '10:00:00', 1),  
(12548, 'AGC', '12:30:00', '12:35:00', 2);

INSERT INTO Route (Train\_no, StationID, ArrivalTime, DepartureTime, StopNumber) VALUES  
(12615, 'MAS', '00:00:00', '10:00:00', 1),  
(12615, 'NDLS', '10:00:00', '00:00:00', 2);

INSERT INTO Route (Train\_no, StationID, ArrivalTime, DepartureTime, StopNumber) VALUES  
(12627, 'SBC', '00:00:00', '10:00:00', 1),

```

(12627, 'NDLS', '10:00:00', '00:00:00', 2);

CREATE TABLE Class (
    Train_no int(5),
    ClassID INT AUTO_INCREMENT PRIMARY KEY,
    ClassName VARCHAR(50),
    Fare int DEFAULT 1.0,
    no_of_coaches int,
    FromRouteID INT,
    ToRouteID INT,   foreign key (Train_no)
    references Train(Train_no),
    FOREIGN KEY (FromRouteID) REFERENCES Route(RouteID),
    FOREIGN KEY (ToRouteID) REFERENCES Route(RouteID),
    available_seats int
);

-- 12245 Howrah Yesvantpur Duronto (HWH to YPR)

INSERT INTO Class (Train_no, ClassName, Fare, no_of_coaches, FromRouteID, ToRouteID,
available_seats)
VALUES
(12245, '1A', 4200, 1, 1, 3, 18),
(12245, '2A', 2500, 3, 1, 3, 54),
(12245, '3A', 1800, 6, 1, 3, 108);

-- 12301 Kolkata Rajdhani Express (HWH to NDLS)

INSERT INTO Class (Train_no, ClassName, Fare, no_of_coaches, FromRouteID, ToRouteID,
available_seats)
VALUES
(12301, '1A', 4350, 1, 4, 10, 18),
(12301, '2A', 2550, 3, 4, 10, 54),

```

(12301, '3A', 1800, 6, 4, 10, 108);

-- 12423 Rajdhani Express (DBRG to NDLS)

INSERT INTO Class (Train\_no, ClassName, Fare, no\_of\_coaches, FromRouteID, ToRouteID, available\_seats)

VALUES

(12423, '1A', 4550, 1, 11, 18, 18),

(12423, '2A', 2600, 3, 11, 18, 54),

(12423, '3A', 1850, 6, 11, 18, 108);

-- 12723 Telangana Express (HYB to NDLS)

INSERT INTO Class (Train\_no, ClassName, Fare, no\_of\_coaches, FromRouteID, ToRouteID, available\_seats)

VALUES

(12723, '2A', 2700, 2, 37, 54, 48),

(12723, '3A', 1850, 5, 37, 54, 90),

(12723, 'SL', 750, 10, 37, 54, 180);

-- 12548 Sabarmati - Agra Cantt. SF Express (ADI to AGC)

INSERT INTO Class (Train\_no, ClassName, Fare, no\_of\_coaches, FromRouteID, ToRouteID, available\_seats)

VALUES

(12548, '3A', 1600, 4, 55, 56, 72),

(12548, 'SL', 600, 12, 55, 56, 144);

-- 12615 Grand Trunk Express (MAS to NDLS)

INSERT INTO Class (Train\_no, ClassName, Fare, no\_of\_coaches, FromRouteID, ToRouteID, available\_seats)

VALUES

```
(12615, '1A', 4100, 1, 57, 58, 18),  
(12615, '2A', 2400, 2, 57, 58, 48),  
(12615, '3A', 1650, 5, 57, 58, 90),  
(12615, 'SL', 700, 10, 57, 58, 180);
```

-- 12627 Karnataka Express (SBC to NDLS)

```
INSERT INTO Class (Train_no, ClassName, Fare, no_of_coaches, FromRouteID, ToRouteID,  
available_seats)
```

VALUES

```
(12627, '2A', 2450, 3, 59, 60, 54),  
(12627, '3A', 1700, 6, 59, 60, 108),  
(12627, 'SL', 600, 12, 59, 60, 180);
```

CREATE TABLE Seat (

```
SeatID INT AUTO_INCREMENT PRIMARY KEY,  
ClassID INT,  
SeatNumber VARCHAR(10),  
FOREIGN KEY (ClassID) REFERENCES Class(ClassID)  
);
```

INSERT INTO Seat (ClassID, SeatNumber) VALUES

```
(1, 'H1-01'),(1, 'H1-02'),(1, 'H1-03'),(1, 'H1-04'),(1, 'H1-05'),(1, 'H1-06'),(1, 'H1-07'),(1,  
'H108'),(1, 'H1-09'),(1, 'H1-10'),(1, 'H1-11'),(1, 'H1-12'),(1, 'H1-13'),(1, 'H1-14'),(1, 'H1-15'),(1,  
'H1-16'),(1, 'H1-17'),(1, 'H1-18'),(2, 'A1-01'),(2, 'A1-02'),(2, 'A1-03'),(2, 'A1-04'),(2, 'A1-05'),(2,  
'A1-06'),(2, 'A1-07'),(2, 'A1-08'),(2, 'A1-09'),(2, 'A1-10'),(2, 'A1-11'),(2, 'A1-12'),(2, 'A1-13'),(2,  
'A1-14'),(2, 'A1-15'),(2, 'A1-16'),(2, 'A1-17'),  
(2, 'A1-18'),(2, 'A1-19'),(2, 'A1-20'),(2, 'A1-21'),(2, 'A1-22'),(2, 'A1-23'),(2, 'A1-24'),(2, 'A1-  
25'),(2, 'A1-26'),(2, 'A1-27'),(2, 'A1-28'),(2, 'A1-29'),(2, 'A1-30'),(2, 'A1-31'),(2, 'A1-32'),(2, 'A1-  
33'),(2, 'A1-34'),(2, 'A1-35'),(2, 'A1-36'),(2, 'A1-37'),(2, 'A1-38'),(2, 'A1-39'),(2, 'A1-40'),(2, 'A1-  
41'),(2, 'A1-42'),(2, 'A1-43'),(2, 'A1-44'),(2, 'A1-45'),(2, 'A1-46'),(2, 'A1-47'),(2, 'A1-48'),(2, 'A1-  
49'),(2, 'A1-50'),(2, 'A1-51'),(2, 'A1-52'),  
(2, 'A1-53'),(2, 'A1-54'),(2, 'A2-01'),(2, 'A2-02'),(2, 'A2-03'),(2, 'A2-04'),(2, 'A2-05'),(2, 'A2-
```

(2, 'A2-06'),(2, 'A2-07'),(2, 'A2-08'),(2, 'A2-09'),(2, 'A2-10'),(2, 'A2-11'),(2, 'A2-12'),(2, 'A2-13'),(2, 'A2-14'),(2, 'A2-15'),(2, 'A2-16'),(2, 'A2-17'),(2, 'A2-18'),(2, 'A2-19'),(2, 'A2-20'),(2, 'A2-21'),(2, 'A2-22'),(2, 'A2-23'),(2, 'A2-24'),(2, 'A2-25'),(2, 'A2-26'),(2, 'A2-27'),(2, 'A2-28'),(2, 'A2-29'),(2, 'A2-30'),(2, 'A2-31'),(2, 'A2-32'),(2, 'A2-33'),  
(2, 'A2-34'),(2, 'A2-35'),(2, 'A2-36'),(2, 'A2-37'),(2, 'A2-38'),(2, 'A2-39'),(2, 'A2-40'),(2, 'A2-41'),(2, 'A2-42'),(2, 'A2-43'),(2, 'A2-44'),(2, 'A2-45'),(2, 'A2-46'),(2, 'A2-47'),(2, 'A2-48'),(2, 'A2-49'),(2, 'A2-50'),(2, 'A2-51'),(2, 'A2-52'),(2, 'A2-53'),(2, 'A2-54'),(2, 'A3-01'),(2, 'A3-02'),(2, 'A3-03'),(2, 'A3-04'),(2, 'A3-05'),(2, 'A3-06'),(2, 'A3-07'),(2, 'A3-08'),(2, 'A3-09'),(2, 'A3-10'),(2, 'A3-11'),(2, 'A3-12'),(2, 'A3-13'),(2, 'A3-14'),  
(2, 'A3-15'),(2, 'A3-16'),(2, 'A3-17'),(2, 'A3-18'),(2, 'A3-19'),(2, 'A3-20'),(2, 'A3-21'),(2, 'A3-22'),(2, 'A3-23'),(2, 'A3-24'),(2, 'A3-25'),(2, 'A3-26'),(2, 'A3-27'),(2, 'A3-28'),(2, 'A3-29'),(2, 'A3-30'),(2, 'A3-31'),(2, 'A3-32'),(2, 'A3-33'),(2, 'A3-34'),(2, 'A3-35'),(2, 'A3-36'),(2, 'A3-37'),(2, 'A3-38'),(2, 'A3-39'),(2, 'A3-40'),(2, 'A3-41'),(2, 'A3-42'),(2, 'A3-43'),(2, 'A3-44'),(2, 'A3-45'),(2, 'A3-46'),(2, 'A3-47'),(2, 'A3-48'),(2, 'A3-49'),  
(2, 'A3-50'),(2, 'A3-51'),(2, 'A3-52'),(2, 'A3-53'),(2, 'A3-54'),(3, 'B1-01'),(3, 'B1-02'),(3, 'B103'),(3, 'B1-04'),(3, 'B1-05'),(3, 'B1-06'),(3, 'B1-07'),(3, 'B1-08'),(3, 'B1-09'),(3, 'B1-10'),(3, 'B1-11'),(3, 'B1-12'),(3, 'B1-13'),(3, 'B1-14'),(3, 'B1-15'),(3, 'B1-16'),(3, 'B1-17'),(3, 'B1-18'),(3, 'B1-19'),(3, 'B1-20'),(3, 'B1-21'),(3, 'B1-22'),(3, 'B1-23'),(3, 'B1-24'),(3, 'B1-25'),(3, 'B1-26'),(3, 'B1-27'),(3, 'B1-28'),(3, 'B1-29'),(3, 'B1-30'),  
(3, 'B1-31'),(3, 'B1-32'),(3, 'B1-33'),(3, 'B1-34'),(3, 'B1-35'),(3, 'B1-36'),(3, 'B1-37'),(3, 'B1-38'),(3, 'B1-39'),(3, 'B1-40'),(3, 'B1-41'),(3, 'B1-42'),(3, 'B1-43'),(3, 'B1-44'),(3, 'B1-45'),(3, 'B1-46'),(3, 'B1-47'),(3, 'B1-48'),(3, 'B1-49'),(3, 'B1-50'),(3, 'B1-51'),(3, 'B1-52'),(3, 'B1-53'),(3, 'B1-54'),(3, 'B1-55'),(3, 'B1-56'),(3, 'B1-57'),(3, 'B1-58'),(3, 'B1-59'),(3, 'B1-60'),(3, 'B1-61'),(3, 'B1-62'),(3, 'B1-63'),(3, 'B1-64'),(3, 'B1-65'),  
(3, 'B1-66'),(3, 'B1-67'),(3, 'B1-68'),(3, 'B1-69'),(3, 'B1-70'),(3, 'B1-71'),(3, 'B1-72'),(3, 'B1-73'),(3, 'B1-74'),(3, 'B1-75'),(3, 'B1-76'),(3, 'B1-77'),(3, 'B1-78'),(3, 'B1-79'),(3, 'B1-80'),(3, 'B1-81'),(3, 'B1-82'),(3, 'B1-83'),(3, 'B1-84'),(3, 'B1-85'),(3, 'B1-86'),(3, 'B1-87'),(3, 'B1-88'),(3, 'B1-89'),(3, 'B1-90'),(3, 'B1-91'),(3, 'B1-92'),(3, 'B1-93'),(3, 'B1-94'),(3, 'B1-95'),(3, 'B1-96'),(3, 'B1-97'),(3, 'B1-98'),(3, 'B1-99'),(3, 'B1-100'),  
(3, 'B1-101'),(3, 'B1-102'),(3, 'B1-103'),(3, 'B1-104'),(3, 'B1-105'),(3, 'B1-106'),(3, 'B1-107'),(3, 'B1-108'),(3, 'B2-01'),(3, 'B2-02'),(3, 'B2-03'),(3, 'B2-04'),(3, 'B2-05'),(3, 'B2-06'),(3, 'B2-07'),(3, 'B2-08'),(3, 'B2-09'),(3, 'B2-10'),(3, 'B2-11'),(3, 'B2-12'),(3, 'B2-13'),(3, 'B2-14'),(3, 'B2-15'),(3, 'B2-16'),(3, 'B2-17'),(3, 'B2-18'),(3, 'B2-19'),(3, 'B2-20'),(3, 'B2-21'),(3, 'B2-22'),(3, 'B2-23'),(3, 'B2-24'),(3, 'B2-25'),(3, 'B2-26'),  
(3, 'B2-27'),(3, 'B2-28'),(3, 'B2-29'),(3, 'B2-30'),(3, 'B2-31'),(3, 'B2-32'),(3, 'B2-33'),(3, 'B2-

34'),(3, 'B2-35'),(3, 'B2-36'),(3, 'B2-37'),(3, 'B2-38'),(3, 'B2-39'),(3, 'B2-40'),(3, 'B2-41'),(3, 'B2-42'),(3, 'B2-43'),(3, 'B2-44'),(3, 'B2-45'),(3, 'B2-46'),(3, 'B2-47'),(3, 'B2-48'),(3, 'B2-49'),(3, 'B2-50'),(3, 'B2-51'),(3, 'B2-52'),(3, 'B2-53'),(3, 'B2-54'),(3, 'B2-55'),(3, 'B2-56'),(3, 'B2-57'),(3, 'B2-58'),(3, 'B2-59'),(3, 'B2-60'),(3, 'B2-61'),  
 (3, 'B2-62'),(3, 'B2-63'),(3, 'B2-64'),(3, 'B2-65'),(3, 'B2-66'),(3, 'B2-67'),(3, 'B2-68'),(3, 'B2-69'),(3, 'B2-70'),(3, 'B2-71'),(3, 'B2-72'),(3, 'B2-73'),(3, 'B2-74'),(3, 'B2-75'),(3, 'B2-76'),(3, 'B2-77'),(3, 'B2-78'),(3, 'B2-79'),(3, 'B2-80'),(3, 'B2-81'),(3, 'B2-82'),(3, 'B2-83'),(3, 'B2-84'),(3, 'B2-85'),(3, 'B2-86'),(3, 'B2-87'),(3, 'B2-88'),(3, 'B2-89'),(3, 'B2-90'),(3, 'B2-91'),(3, 'B2-92'),(3, 'B2-93'),(3, 'B2-94'),(3, 'B2-95'),(3, 'B2-96'),  
 (3, 'B2-97'),(3, 'B2-98'),(3, 'B2-99'),(3, 'B2-100'),(3, 'B2-101'),(3, 'B2-102'),(3, 'B2-103'),(3, 'B2-104'),(3, 'B2-105'),(3, 'B2-106'),(3, 'B2-107'),(3, 'B2-108'),(3, 'B3-01'),(3, 'B3-02'),(3, 'B303'),(3, 'B3-04'),(3, 'B3-05'),(3, 'B3-06'),(3, 'B3-07'),(3, 'B3-08'),(3, 'B3-09'),(3, 'B3-10'),(3, 'B3-11'),(3, 'B3-12'),(3, 'B3-13'),(3, 'B3-14'),(3, 'B3-15'),(3, 'B3-16'),(3, 'B3-17'),(3, 'B3-18'),(3, 'B319'),(3, 'B3-20'),(3, 'B3-21'),(3, 'B3-22'),  
 (3, 'B3-23'),(3, 'B3-24'),(3, 'B3-25'),(3, 'B3-26'),(3, 'B3-27'),(3, 'B3-28'),(3, 'B3-29'),(3, 'B3-30'),(3, 'B3-31'),(3, 'B3-32'),(3, 'B3-33'),(3, 'B3-34'),(3, 'B3-35'),(3, 'B3-36'),(3, 'B3-37'),(3, 'B3-38'),(3, 'B3-39'),(3, 'B3-40'),(3, 'B3-41'),(3, 'B3-42'),(3, 'B3-43'),(3, 'B3-44'),(3, 'B3-45'),(3, 'B3-46'),(3, 'B3-47'),(3, 'B3-48'),(3, 'B3-49'),(3, 'B3-50'),(3, 'B3-51'),(3, 'B3-52'),(3, 'B3-53'),(3, 'B354'),(3, 'B3-55'),(3, 'B3-56'),(3, 'B3-57'),  
 (3, 'B3-58'),(3, 'B3-59'),(3, 'B3-60'),(3, 'B3-61'),(3, 'B3-62'),(3, 'B3-63'),(3, 'B3-64'),(3, 'B3-65'),(3, 'B3-66'),(3, 'B3-67'),(3, 'B3-68'),(3, 'B3-69'),(3, 'B3-70'),(3, 'B3-71'),(3, 'B3-72'),(3, 'B3-73'),(3, 'B3-74'),(3, 'B3-75'),(3, 'B3-76'),(3, 'B3-77'),(3, 'B3-78'),(3, 'B3-79'),(3, 'B3-80'),(3, 'B3-81'),(3, 'B3-82'),(3, 'B3-83'),(3, 'B3-84'),(3, 'B3-85'),(3, 'B3-86'),(3, 'B3-87'),(3, 'B3-88'),(3, 'B389'),(3, 'B3-90'),(3, 'B3-91'),(3, 'B3-92'),  
 (3, 'B3-93'),(3, 'B3-94'),(3, 'B3-95'),(3, 'B3-96'),(3, 'B3-97'),(3, 'B3-98'),(3, 'B3-99'),(3, 'B3100'),(3, 'B3-101'),(3, 'B3-102'),(3, 'B3-103'),(3, 'B3-104'),(3, 'B3-105'),(3, 'B3-106'),(3, 'B3-107'),(3, 'B3-108');

CREATE TABLE Ticket (

- TicketID INT AUTO\_INCREMENT PRIMARY KEY,
- PassengerID INT,
- ClassID INT,
- FromRouteID INT,
- ToRouteID INT,
- SeatID INT, -- Can be NULL for waitlisted tickets

```
    PNR int(10) UNIQUE,  
    DateOfJourney DATE,  
    BookingDate DATE,  
    BookingStatus VARCHAR(20), -- Confirmed, RAC, WL  
    FOREIGN KEY (PassengerID) REFERENCES Passenger(PassengerID),  
    FOREIGN KEY (ClassID) REFERENCES Class(ClassID),   foreign key  
(FromRouteID) REFERENCES Route(RouteID),   foreign key  
(ToRouteID) REFERENCES Route(RouteID),  
    FOREIGN KEY (SeatID) REFERENCES Seat(SeatID)  
);
```

```
ALTER TABLE Ticket MODIFY PNR BIGINT UNIQUE;
```

```
CREATE TABLE Passenger (  
    PassengerID INT PRIMARY KEY,  
    Name VARCHAR(100) NOT NULL,  
    Age INT,  
    Gender CHAR(1),  
    Phone VARCHAR(15),  
    Wallet DECIMAL(10,2) DEFAULT 0.00,
```

```
concession VARCHAR(50)
```

```
);
```

```
select*from Passenger;
```

```
CREATE TABLE Payment (  
    PaymentID INT AUTO_INCREMENT PRIMARY KEY,  
    TicketID INT,  
    PaymentMode VARCHAR(50), -- UPI, Credit Card, Cash, etc.  
    Amount DECIMAL(10,2),  
    PaymentStatus VARCHAR(10), -- Success, Failed
```

```
FOREIGN KEY (TicketID) REFERENCES Ticket(TicketID)
);


```

```
CREATE TABLE Cancellation (
    CancellationID INT AUTO_INCREMENT PRIMARY KEY,
    TicketID INT UNIQUE,
    CancelDate DATE,
    FOREIGN KEY (TicketID) REFERENCES Ticket(TicketID)
);
```

## Triggers, Procedures, Functions and Queries

```
DELIMITER $$
```

```
CREATE FUNCTION billing_desk (pid INT,train_no INT,from_route_id INT,to_route_id  
INT,class_name VARCHAR(50),journey_date DATE,payment_mode VARCHAR(50))
```

```
RETURNS VARCHAR(255)
```

```
DETERMINISTIC
```

```
MODIFIES SQL DATA
```

```
BEGIN
```

```
    DECLARE date_day VARCHAR(50);
```

```
    DECLARE class_id INT;
```

```
    DECLARE seat_count INT DEFAULT 0;
```

```
    DECLARE rac_limit INT DEFAULT 0;
```

```
    DECLARE rac_current INT DEFAULT 0;
```

```
    DECLARE wl_current INT DEFAULT 0;
```

```
    DECLARE confirmed_current INT DEFAULT 0;
```

```
    DECLARE status VARCHAR(20);
```

```
    DECLARE seat_id INT;
```

```
    DECLARE pnr_generated INT;
```

```
    DECLARE base_fare DECIMAL(10,2);
```

```

DECLARE total_fare DECIMAL(10,2);
DECLARE concession_type VARCHAR(50);
DECLARE concession_amt DECIMAL(10,2) DEFAULT 0.00;
DECLARE convenience_fee DECIMAL(10,2) DEFAULT 50.00;
DECLARE wallet_bal DECIMAL(10,2) DEFAULT 0.00;
DECLARE new_payment_id INT;

SELECT ClassID, available_seats*no_of_coaches, Fare INTO class_id, seat_count, base_fare
FROM Class
WHERE Train_no = train_no AND ClassName = class_name LIMIT 1;

IF class_id IS NULL THEN
    RETURN 'Class does not exist for this Train or ClassName.';
END IF;

SELECT Wallet, concession INTO wallet_bal, concession_type
FROM Passenger WHERE PassengerID = pid LIMIT 1;

IF wallet_bal IS NULL THEN
    RETURN 'Passenger not found.';
END IF;

SELECT DaysOfOperation INTO date_day FROM Train where Train_no = train_no LIMIT 1;

IF LOCATE(DATE_FORMAT(journey_date, '%a'), date_day) = 0 THEN
    RETURN (' Train does not run on this day.');
END IF;

IF concession_type = 'Senior Citizen' THEN
    SET concession_amt = base_fare * 0.40;
ELSEIF concession_type = 'Disabled' THEN
    SET concession_amt = base_fare * 0.50;
ELSEIF concession_type = 'Ladies' THEN
    SET concession_amt = base_fare * 0.35;
ELSE

```

```

SET concession_amt = 0.00;

END IF;

SET convenience_fee = 0.20 * base_fare;

SET total_fare = base_fare - concession_amt + convenience_fee;

IF journey_date < curdate() THEN

    return CONCAT('Train Departed. No booking can be done. Amount has not been
deducted:',total_fare);

END IF;

SELECT COUNT(*) INTO confirmed_current

FROM Ticket WHERE FromRouteID >= from_route_id AND ToRouteID <= to_route_id AND
ClassID=class_id AND DateOfJourney = journey_date AND BookingStatus = 'Confirmed';

SELECT SUM(CASE WHEN BookingStatus = 'RAC1' THEN 1 WHEN BookingStatus = 'RAC'
THEN 2 ELSE 0 END) INTO rac_current

FROM Ticket WHERE FromRouteID >= from_route_id AND ToRouteID <= to_route_id AND
ClassID=class_id AND DateOfJourney = journey_date AND (BookingStatus = 'RAC1' OR
BookingStatus = 'RAC');

SELECT COUNT(*) INTO wl_current FROM Ticket WHERE FromRouteID >= from_route_id
AND ToRouteID <= to_route_id AND ClassID=class_id AND DateOfJourney = journey_date
AND BookingStatus = 'WL';

IF class_name = 'SL' THEN

    SET rac_limit = 14;

ELSEIF class_name = '3A' THEN

    SET rac_limit = 8;

ELSE

    SET rac_limit = 0;

END IF;

IF wallet_bal < total_fare THEN

    INSERT INTO Payment (PaymentMode, Amount, PaymentStatus) VALUES
(payment_mode, total_fare, 'Failed');

    RETURN 'Transaction Failed: Insufficient Wallet Balance';

```

```

ELSE
    -- Deduct and record payment
    UPDATE Passenger SET Wallet = Wallet - total_fare WHERE PassengerID = pid;
    INSERT INTO Payment (PaymentMode, Amount, PaymentStatus) VALUES
    (payment_mode, total_fare, 'Success');

    SET new_payment_id = LAST_INSERT_ID();

    -- Determine status and assign seat
    IF confirmed_current < (seat_count - (rac_limit/2)) THEN
        SET status = 'Confirmed';
        SELECT SeatID INTO seat_id FROM Seat WHERE ClassID = class_id AND SeatID NOT IN
        (
            SELECT SeatID FROM Ticket WHERE FromRouteID >= from_route_id AND ToRouteID
            <= to_route_id
            AND ClassID = class_id AND DateOfJourney = journey_date AND BookingStatus =
            'Confirmed'
        ) LIMIT 1;

        ELSEIF rac_current < (rac_limit/2) THEN
            SET status = 'RAC1';
            SELECT SeatID INTO seat_id FROM Seat WHERE ClassID = class_id AND SeatID NOT IN
            (
                SELECT SeatID FROM Ticket WHERE FromRouteID >= from_route_id AND ToRouteID
                <= to_route_id
                AND ClassID = class_id AND DateOfJourney = journey_date AND (BookingStatus =
                'Confirmed' OR BookingStatus = 'RAC')
            ) LIMIT 1;

            ELSEIF rac_current < rac_limit THEN
                SET status = 'RAC';
                SELECT SeatID INTO seat_id FROM Ticket WHERE FromRouteID >=
                from_route_id AND ToRouteID <= to_route_id

```

```

        AND ClassID = class_id AND DateOfJourney = journey_date AND
BookingStatus = 'RAC1'

        LIMIT 1;

    ELSE

        SET status = 'WL';

        SET seat_id = NULL;

    END IF;

    SET pnr_generated = FLOOR(1000000000 + RAND() * 899999);

    INSERT INTO Ticket (PassengerID, ClassID, FromRouteID, ToRouteID, SeatID, PNR,
DateOfJourney, BookingDate, BookingStatus)

        VALUES (pid, class_id, from_route_id, to_route_id, seat_id, pnr_generated,
journey_date, CURDATE(), status);

    UPDATE Payment SET TicketID = LAST_INSERT_ID() WHERE PaymentID =
new_payment_id;

    RETURN CONCAT('Transaction Successful. Rs ', total_fare, ' deducted from wallet.',
'\t','Ticket booked: ', status, '. PNR: ', pnr_generated);

END IF;

END$$

DELIMITER ;

```

```
select billing_desk(10,12245,1,3,'3A','2025-04-16','Credit Card');
```

```

DELIMITER $$

CREATE TRIGGER after_cancellation_update_waitlist
AFTER INSERT ON Cancellation
FOR EACH ROW
BEGIN

    DECLARE v_class_id INT;
    DECLARE v_from_route INT;

```

```

DECLARE v_to_route INT;
DECLARE v_journey_date DATE;
DECLARE v_cancelled_seat INT;
DECLARE v_wait_ticket_id INT;

SELECT ClassID, FromRouteID, ToRouteID, DateOfJourney, SeatID
INTO v_class_id, v_from_route, v_to_route, v_journey_date, v_cancelled_seat
FROM Ticket
WHERE TicketID = NEW.TicketID;

SELECT TicketID
INTO v_wait_ticket_id
FROM Ticket
WHERE BookingStatus = 'WL'
AND ClassID = v_class_id
AND FromRouteID = v_from_route
AND ToRouteID = v_to_route
AND DateOfJourney = v_journey_date
ORDER BY TicketID
LIMIT 1;

IF v_wait_ticket_id IS NOT NULL THEN
    UPDATE Ticket
    SET BookingStatus = 'Confirmed',
        SeatID = v_cancelled_seat
    WHERE TicketID = v_wait_ticket_id;
END IF;

END$$

DELIMITER ;

```

DELIMITER \$\$

```

CREATE FUNCTION cancel_ticket(pnr BIGINT)
RETURNS VARCHAR(255)
DETERMINISTIC
MODIFIES SQL DATA
BEGIN

    DECLARE t_passenger_id INT;
    DECLARE t_fare DECIMAL(10,2);
    DECLARE t_payment_id INT;
    DECLARE t_wallet DECIMAL(10,2);
    DECLARE t_status VARCHAR(20);

    DECLARE ticket_ID INT;

    -- Get passenger ID and fare from the ticket and payment
    SELECT t.PassengerID, p.Amount, p.PaymentID, t.BookingStatus, t.TicketID
    INTO t_passenger_id, t_fare, t_payment_id, t_status, ticket_ID
    FROM Ticket t
    JOIN Payment p ON t.TicketID = p.TicketID
    WHERE t.PNR = pnr;

    -- Check if ticket is already cancelled
    IF t_status = 'Cancelled' THEN
        RETURN 'Ticket is already cancelled.';
    END IF;

    UPDATE Passenger SET Wallet = Wallet + t_fare WHERE PassengerID = t_passenger_id;
    UPDATE Ticket SET BookingStatus = 'Cancelled' WHERE PNR = pnr;
    UPDATE Payment SET PaymentStatus = 'Refunded' WHERE PaymentID = t_payment_id;
    INSERT INTO Cancellation(TicketID,CancelDate)VALUES(ticket_ID,CURDATE());

    RETURN CONCAT('Ticket with PNR ', pnr, ' has been cancelled. Rs ', t_fare, ' refunded to wallet.');

```

```
END$$
```

```
DELIMITER ;
```

```
select cancel_ticket(2428916017);
```

```
DELIMITER $$
```

```
CREATE TRIGGER after_ticket_insert_add_payment
```

```
AFTER INSERT ON Ticket
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE base_fare DECIMAL(10,2);
```

```
    DECLARE convenience_fee DECIMAL(10,2);
```

```
    DECLARE total_amount DECIMAL(10,2);
```

```
    DECLARE concession_type VARCHAR(50);
```

```
    DECLARE concession_amt DECIMAL(10,2);
```

```
-- Get base fare from Class
```

```
    SELECT Fare INTO base_fare
```

```
    FROM Class
```

```
    WHERE ClassID = NEW.ClassID;
```

```
-- Get concession type from Passenger
```

```
    SELECT p.concession INTO concession_type
```

```
    FROM Passenger p
```

```
    WHERE p.PassengerID = NEW.PassengerID;
```

```
-- Calculate concession amount
```

```
    IF concession_type = 'Senior Citizen' THEN
```

```
        SET concession_amt = base_fare * 0.40;
```

```
    ELSEIF concession_type = 'Disabled' THEN
```

```

SET concession_amt = base_fare * 0.50;

ELSEIF concession_type = 'Ladies' THEN

    SET concession_amt = base_fare * 0.35;

ELSE

    SET concession_amt = 0.00;

END IF;

-- Calculate convenience fee

SET convenience_fee = base_fare * 0.20;

-- Total payable

SET total_amount = base_fare - concession_amt + convenience_fee;

-- Insert into Payment table

INSERT INTO Payment (TicketID, PaymentMode, Amount, PaymentStatus)

VALUES (NEW.TicketID, 'UPI', total_amount, 'Success');

END$$

DELIMITER ;

```

```

insert into
Ticket(PassengerID,ClassID,FromRouteID,ToRouteID,SeatID,PNR,DateOfJourney,BookingDate,
BookingStatus) values(10,13,55,56,195,2428916017,'2025-04-21','2025-04-15','Confirmed');

drop trigger after_ticket_insert_add_payment;

-- 1st query

SELECT t.PNR, p.Name AS PassengerName, p.Age, p.Gender, tr.Train_no, tr.TrainName,
r1.StationId AS FromStation,
r2.StationId AS ToStation, t.DateOfJourney, t.BookingStatus, s.SeatNumber, c.ClassName,
pay.PaymentStatus, pay.PaymentMode, pay.Amount

FROM Ticket t

JOIN Passenger p ON t.PassengerID = p.PassengerID

JOIN Class c ON t.ClassID = c.ClassID

JOIN Seat s ON t.SeatID = s.SeatID

```

```
JOIN Route r1 ON t.FromRouteID = r1.RouteID  
JOIN Route r2 ON t.ToRouteID = r2.RouteID  
JOIN Train tr ON r1.Train_no = tr.Train_no AND r1.Train_no = r2.Train_no  
LEFT JOIN Payment pay ON t.TicketID = pay.TicketID  
WHERE t.PNR = 3397941077; -- replace with your PNR
```

-- 2nd query

```
DELIMITER $$  
  
CREATE PROCEDURE train_schedule(IN input_train_no INT)  
  
BEGIN  
  
    DECLARE train_exists INT DEFAULT 0;  
  
    SELECT COUNT(*) INTO train_exists FROM Class WHERE Train_no = input_train_no;  
  
    IF train_exists = 0 THEN  
  
        SIGNAL SQLSTATE '45000'  
  
        SET MESSAGE_TEXT = 'Train does not exist. Please enter a valid Train number.';  
  
    END IF;  
  
    SELECT t.Train_no, t.TrainName, ss.StationName AS SourceStation, ds.StationName AS  
DestinationStation,  
  
        t.DaysOfOperation, r.StopNumber, s.StationName AS RouteStation, r.ArrivalTime,  
r.DepartureTime  
  
    FROM Train t  
  
    JOIN Station ss ON t.SourceStationID = ss.Stationcode  
  
    JOIN Station ds ON t.DestinationStationID = ds.Stationcode  
  
    JOIN Route r ON t.Train_no = r.Train_no  
  
    JOIN Station s ON r.StationID = s.Stationcode  
  
    WHERE t.Train_no = input_train_no  
  
    ORDER BY r.StopNumber;  
  
END $$
```

```
DELIMITER ;
```

```
call train_schedule(12245);
```

```
-- 3rd query
```

```
DELIMITER $$
```

```
CREATE PROCEDURE available_seats (IN train_no INT, IN class_name VARCHAR(50), IN journey_date DATE)
```

```
BEGIN
```

```
    DECLARE class_id INT;
```

```
    DECLARE total_seats INT DEFAULT 0;
```

```
    DECLARE rac_limit INT DEFAULT 0;
```

```
    DECLARE confirmed_count INT DEFAULT 0;
```

```
    DECLARE rac_count INT DEFAULT 0;
```

```
    DECLARE wl_count INT DEFAULT 0;
```

```
    DECLARE train_exists INT DEFAULT 0;
```

```
    DECLARE Date_Day VARCHAR(50);
```

```
        SELECT COUNT(*) INTO train_exists FROM Class WHERE Train_no = train_no;
```

```
        IF train_exists = 0 THEN
```

```
            SIGNAL SQLSTATE '45000'
```

```
            SET MESSAGE_TEXT = 'Train does not exist. Please enter a valid Train number.';
```

```
        END IF;    select DaysOfOperation INTO Date_Day FROM Train where Train_no = train_no
```

```
        LIMIT 1;
```

```
        IF LOCATE(DATE_FORMAT(journey_date, '%a'), Date_Day) = 0 THEN
```

```
            SIGNAL SQLSTATE '45000'
```

```
            SET MESSAGE_TEXT = 'Train does not run on particular date. Please enter a valid Date.';
```

```
        END IF;
```

```

SELECT ClassID, available_seats*no_of_coaches INTO class_id, total_seats FROM Class
WHERE Train_no = train_no AND ClassName = class_name
LIMIT 1;

IF class_id IS NULL THEN
    SIGNAL SQLSTATE '45000'

    SET MESSAGE_TEXT = 'Train does not have particular class. Please enter a valid Class.';

END IF;

IF class_name = 'SL' THEN
    SET rac_limit = 14;
ELSEIF class_name = '3A' THEN
    SET rac_limit = 8;
ELSE
    SET rac_limit = 0;
END IF;

SELECT IFNULL(COUNT(*), 0) INTO confirmed_count FROM Ticket
WHERE ClassID = class_id AND DateOfJourney = journey_date AND BookingStatus =
'Confirmed';

SELECT IFNULL(SUM(CASE
    WHEN BookingStatus = 'RAC1' THEN 1
    WHEN BookingStatus = 'RAC' THEN 2
    ELSE 0
END), 0)
INTO rac_count FROM Ticket
WHERE ClassID = class_id AND DateOfJourney = journey_date AND (BookingStatus =
'RAC1' OR BookingStatus = 'RAC');

SELECT IFNULL(COUNT(*), 0) INTO wl_count FROM Ticket
WHERE ClassID = class_id AND DateOfJourney = journey_date AND BookingStatus = 'WL';

```

```
-- Show final availability  SELECT      total_seats -  
confirmed_count - FLOOR(rac_limit/2) AS Available,      rac_limit -  
rac_count AS RAC_Available,      wl_count AS Waitlist_Count;  
END $$  
DELIMITER ;
```

```
call available_seats(12245,'2A','2025-04-21');
```

```
-- 4th query  
SELECT p.PassengerID, p.Name, p.Age, p.Gender, t.PNR, t.BookingStatus  
FROM Ticket t  
JOIN Passenger p ON t.PassengerID = p.PassengerID  
JOIN Class c ON t.ClassID = c.ClassID  
WHERE c.Train_no = 12245    -- Replace with your desired train number  
AND t.DateOfJourney = '2025-04-21'; -- Replace with your journey date
```

```
-- 5th query  
DELIMITER $$  
CREATE PROCEDURE waitlisted_passengers (  
    IN input_train_no INT,  
    IN input_date DATE  
)  
BEGIN  
    DECLARE train_exists INT DEFAULT 0;  
    DECLARE Date_Day VARCHAR(50);  
    SELECT COUNT(*) INTO train_exists FROM Class WHERE Train_no = input_train_no;  
    IF train_exists = 0 THEN  
        SIGNAL SQLSTATE '45000'
```

```

SET MESSAGE_TEXT = 'Train does not exist. Please enter a valid Train number.';

END IF; select DaysOfOperation INTO Date_Day FROM Train where Train_no =
input_train_no;

IF LOCATE(DATE_FORMAT(input_date, '%a'), Date_Day) = 0 THEN

    SIGNAL SQLSTATE '45000'

SET MESSAGE_TEXT = 'Train does not run on particular date. Please enter a valid Date.';

END IF;

SELECT p.PassengerID, p.Name, p.Age, p.Gender, t.PNR, t.BookingStatus, t.DateOfJourney,
c.Train_no

FROM Ticket t

JOIN Passenger p ON t.PassengerID = p.PassengerID

JOIN Class c ON t.ClassID = c.ClassID

WHERE t.BookingStatus = 'WL'

AND c.Train_no = input_train_no

AND t.DateOfJourney = input_date;

END$$

DELIMITER ; select*from Ticket; call
waitlisted_passengers(12245,'2025-04-21');

-- 6th query

DELIMITER $$

CREATE PROCEDURE refund_for_cancelling_train (
    IN input_train_no INT,
    IN input_date DATE -- assumed train can be cancelled on given date
)

BEGIN

    DECLARE train_exists INT DEFAULT 0;
    DECLARE Date_Day VARCHAR(50);

```

```

SELECT COUNT(*) INTO train_exists FROM Class WHERE Train_no = input_train_no;

IF train_exists = 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Train does not exist. Please enter a valid Train number.';

END IF; select DaysOfOperation INTO Date_Day FROM Train where Train_no =
input_train_no;

IF LOCATE(DATE_FORMAT(input_date, '%a'), Date_Day) = 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Train does not run on particular date. Please enter a valid Date.';

END IF;

SELECT c.Train_no, SUM(p.Amount) AS TotalRefundAmount
FROM Ticket t
JOIN Class c ON t.ClassID = c.ClassID
JOIN Payment p ON t.TicketID = p.TicketID
WHERE c.Train_no = input_train_no
AND t.DateOfJourney = input_date
AND p.PaymentStatus = 'Success'
AND t.BookingStatus <> 'Cancelled'
GROUP BY c.Train_no;

END$$

DELIMITER ;

```

```
call refund_for_cancelling_train(12245,'2025-04-21');
```

```
-- 7th query
```

```
DELIMITER $$
```

```
CREATE PROCEDURE total_convenience_fee_revenue (
    IN start_date DATE,
```

```
    IN end_date DATE
)
BEGIN
SELECT
    IFNULL(ROUND(SUM(c.Fare * 0.20), 2),0) AS TotalConvenienceFeeRevenue
FROM Ticket t
JOIN Payment p ON t.TicketID = p.TicketID
JOIN Class c ON t.ClassID = c.ClassID
WHERE
    p.PaymentStatus = 'Success'
    AND t.DateOfJourney BETWEEN start_date AND end_date;
END $$
```

DELIMITER ;

```
CALL total_convenience_fee_revenue('2025-04-01', '2025-04-30');
```

-- 8th query

```
SELECT t.TicketID, t.PNR, psg.Name AS PassengerName, t.DateOfJourney, c.CancelDate,
pay.Amount AS RefundAmount, pay.PaymentStatus AS RefundStatus
FROM Cancellation c
JOIN Ticket t ON c.TicketID = t.TicketID
JOIN Payment pay ON pay.TicketID = t.TicketID
JOIN Passenger psg ON t.PassengerID = psg.PassengerID
WHERE c.CancelDate IS NOT NULL;
```

-- 9th query

```
SELECT
```

```
t.FromRouteID, t.ToRouteID,  
r1.StationId AS FromStation,  
r2.StationId AS ToStation,  
COUNT(*) AS PassengerCount  
  
FROM Ticket t  
  
JOIN Route r1 ON t.FromRouteID = r1.RouteID  
  
JOIN Route r2 ON t.ToRouteID = r2.RouteID  
  
GROUP BY t.FromRouteID, t.ToRouteID  
  
ORDER BY PassengerCount DESC  
  
LIMIT 1;
```

```
-- 10th query  
  
DELIMITER $$  
  
CREATE PROCEDURE generate_itemized_bill(IN tid INT)  
  
BEGIN  
  
DECLARE pid INT;  
  
DECLARE class_id INT;  
  
DECLARE base_fare DECIMAL(10,2);  
  
DECLARE concession_type VARCHAR(50);  
  
DECLARE concession_amt DECIMAL(10,2) DEFAULT 0.00;  
  
DECLARE convenience_fee DECIMAL(10,2);  
  
DECLARE total_fare DECIMAL(10,2);  
  
DECLARE payment_mode VARCHAR(50);  
  
DECLARE payment_status VARCHAR(10);  
  
DECLARE train_no INT;  
  
DECLARE from_station VARCHAR(10);  
  
DECLARE to_station VARCHAR(10);  
  
DECLARE class_name VARCHAR(50);
```

```

DECLARE journey_date DATE;
DECLARE from_route INT;
DECLARE to_route INT;
DECLARE ticket_exists INT DEFAULT 0;

SELECT COUNT(*) INTO ticket_exists FROM Ticket WHERE TicketID = tid;
IF ticket_exists = 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Ticket ID does not exist. Please enter a valid TicketID.';
END IF;

-- Get ticket details
SELECT PassengerID, ClassID, DateOfJourney, FromRouteID, ToRouteID
INTO pid, class_id, journey_date, from_route, to_route
FROM Ticket
WHERE TicketID = tid;

-- Get base fare, class name and train number
SELECT c.Fare, c.ClassName, c.Train_no
INTO base_fare, class_name, train_no FROM Class c
WHERE c.ClassID = class_id;

-- Get station codes
SELECT StationId INTO from_station FROM Route WHERE RouteID = from_route;
SELECT StationId INTO to_station FROM Route WHERE RouteID = to_route;

-- Get passenger concession
SELECT concession INTO concession_type FROM Passenger WHERE PassengerID = pid;

```

```
-- Calculate concession

IF concession_type = 'Senior Citizen' THEN
    SET concession_amt = base_fare * 0.40;
ELSEIF concession_type = 'Disabled' THEN
    SET concession_amt = base_fare * 0.50;
ELSEIF concession_type = 'Ladies' THEN
    SET concession_amt = base_fare * 0.35;
ELSE
    SET concession_amt = 0.00;
END IF;
```

```
-- Convenience fee is 20% of base fare
SET convenience_fee = 0.20 * base_fare;
SET total_fare = base_fare - concession_amt + convenience_fee;
```

```
-- Get payment details
SELECT PaymentMode, PaymentStatus INTO payment_mode, payment_status FROM
Payment
WHERE TicketID = tid;
-- Final itemized bill   SELECT      tid AS
TicketID,      pid AS PassengerID,
train_no AS TrainNo,      class_name AS
Class,      journey_date AS JourneyDate,
from_station AS FromStation,
to_station AS ToStation,      base_fare AS
BaseFare,      concession_type AS
ConcessionType,      concession_amt AS
```

```
ConcessionAmount,      convenience_fee
AS ConvenienceFee,    total_fare AS
TotalPayable,         payment_mode AS
PaymentMode,          payment_status AS
PaymentStatus;
END$$
```

```
DELIMITER ;
```

```
call generate_itemized_bill(4);
```

```
-- Find the Busiest Station (Most Routes Passing Through)
```

```
SELECT s.StationName, COUNT(*) AS NumberOfStops
FROM Route r
JOIN Station s ON r.StationID = s.Stationcode
GROUP BY s.StationName
ORDER BY NumberOfStops DESC
LIMIT 1;
```

```
SELECT Name, Phone, Wallet
FROM Passenger
ORDER BY Wallet DESC
LIMIT 5;
```

```
SELECT Train_no, TrainName, DaysOfOperation
FROM Train
WHERE SourceStationID = 'HWH' AND DestinationStationID = 'NDLS'
AND DaysOfOperation LIKE '%Mon%';
```