

Tables:

Station(

```
    id INT NOT NULL,  
    name VARCHAR(255) NOT NULL,  
    PRIMARY KEY(id)  
)
```

Passenger(

```
    id int NOT NULL,  
    name varchar(255) NOT NULL,  
    cnic char(13) NOT NULL UNIQUE,  
    email varchar(100) NOT NULL,  
    password varchar(100) NOT NULL,  
    phone_number char(11) NOT NULL,  
    PRIMARY KEY(id)
```

```
)
```

Staff(

```
    id int NOT NULL,  
    name varchar(255) NOT NULL,  
    role varchar(50) NOT NULL,  
    phone_number char(11) NOT NULL,  
    salary int CHECK (salary > 0),  
    PRIMARY KEY(id)
```

```
)
```

Train(

```
    id int NOT NULL,  
    name varchar(100) NOT NULL,  
    PRIMARY KEY (id)
```

```
)
```

Route(

```
    route_id INT NOT NULL,  
    route_name VARCHAR(255) NOT NULL,  
    PRIMARY KEY(route_id)
```

```
)
```

Seat(

```
    id INT NOT NULL,  
    train_id INT NOT NULL,  
    class VARCHAR(50) CHECK(class IN ('Economy','Business','AC')),
```

```
    status VARCHAR(50) CHECK(status IN ('Free','Occupied')),  
    PRIMARY KEY(id),  
    FOREIGN KEY(train_id) REFERENCES Train(id)  
)
```

Ticket(

```
    id INT NOT NULL,  
    train_id INT NOT NULL,  
    passenger_id INT NOT NULL,  
    seat_id INT NOT NULL,  
    payment_id INT,  
    travel_date DATE NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY(train_id) REFERENCES Train(id),  
    FOREIGN KEY(passenger_id) REFERENCES Passenger(id),  
    FOREIGN KEY(seat_id) REFERENCES Seat(id)
```

)

Payment(

```
    id INT NOT NULL,  
    ticket_id INT NOT NULL,  
    amount DECIMAL(10,2) NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY(ticket_id) REFERENCES Ticket(id)
```

)

Staff_Assignment(

```
    staff_id INT NOT NULL,  
    station_id INT NOT NULL,  
    shift_date DATE NOT NULL,  
    shift_type VARCHAR(20) CHECK(shift_type IN ('Morning','Evening')) NOT NULL,  
    PRIMARY KEY(staff_id, shift_date, shift_type),  
    FOREIGN KEY(staff_id) REFERENCES Staff(id),  
    FOREIGN KEY(station_id) REFERENCES Station(id)
```

)

Route_Station(

```
    id INT NOT NULL,  
    route_id INT NOT NULL,  
    station_id INT NOT NULL,  
    arrival_time TIME,  
    departure_time TIME,  
    UNIQUE(route_id, station_id),  
    PRIMARY KEY(id),
```

```
FOREIGN KEY(route_id) REFERENCES Route(route_id),  
FOREIGN KEY(station_id) REFERENCES Station(id)  
)
```

Index usage reasoning:

- On seats status since the number of seats will always be a lot considering practical railways
- On station name since in actual railways there are many routes and each route has a lot of stations for example british railway has around 2500 stations
- On staff id, the shift type(Morning/Evening) and the shift date, since a railway has thousands of employees, couple that with our rule of max 40 hours meaning 8 shifts a week, take it to the scale of years and you would get hundred of thousands of entries, indexing will make it much faster to search among them
- On passenger name and CNIC since there will be a lot of passenger entries perhaps more than hundred thousand practically

Result of Indexing:

The result of indexing for the database was that the execution time improved a small bit, the reason for small bit is that that number of entries is very little hence only a few milli second of difference is seen, analysing it we see that it searches via index while before it doesn't

Roles:

- Passenger role(can view their own tickets book tickets)
- Station Manager role(assign shifts to staff)
- Administrator role(all privileges)