CS254
DATABASE MANAGEMENT SYSTEMS

A*

YOUR TRAVEL AGENT

PREPARED BY

CHETAN KAR

22CSB0F05

SHUBHAM PAHILWANI

22CSB0F08

B. TECH CSE-A

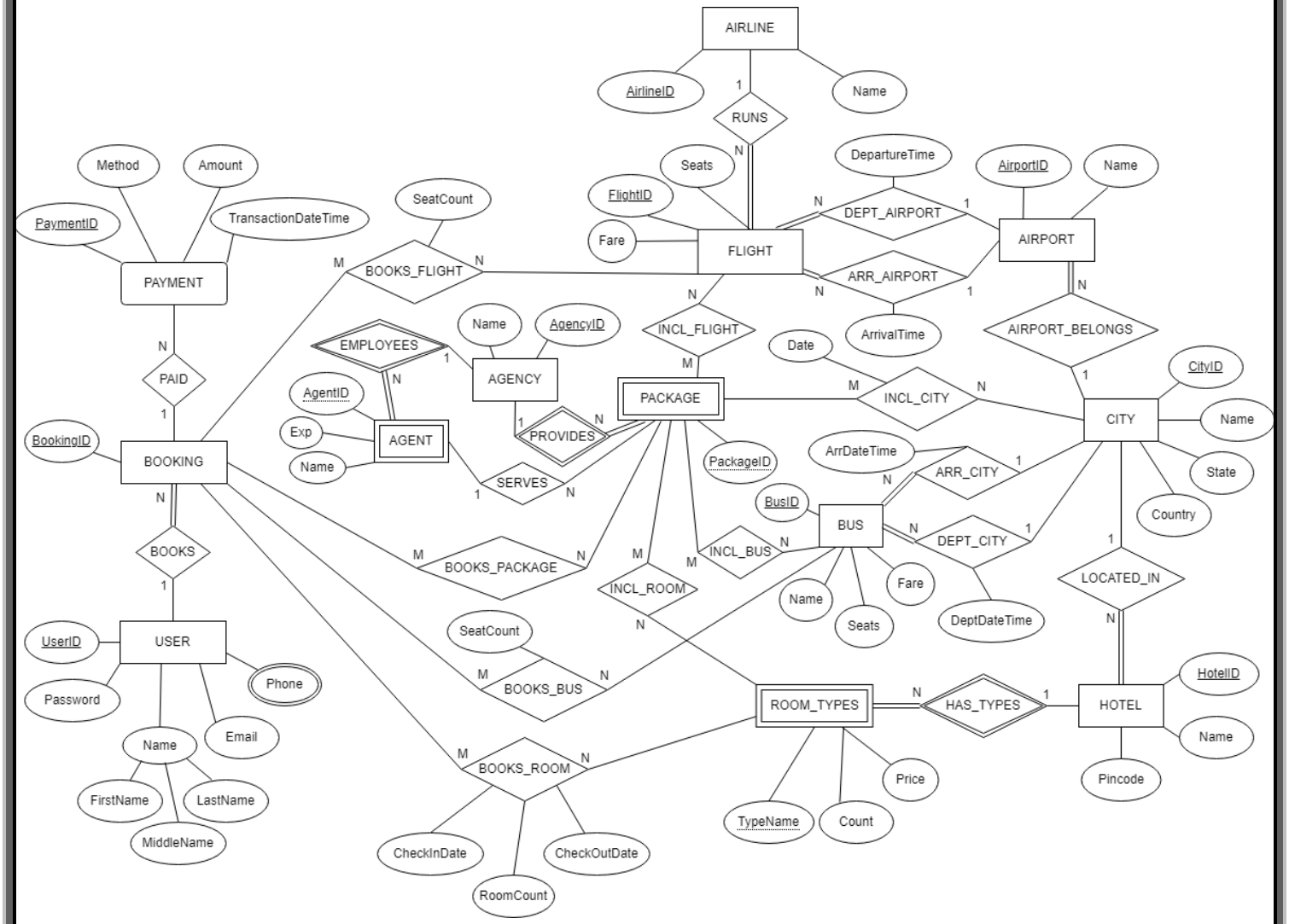
2022-26

TABLE OF CONTENTS

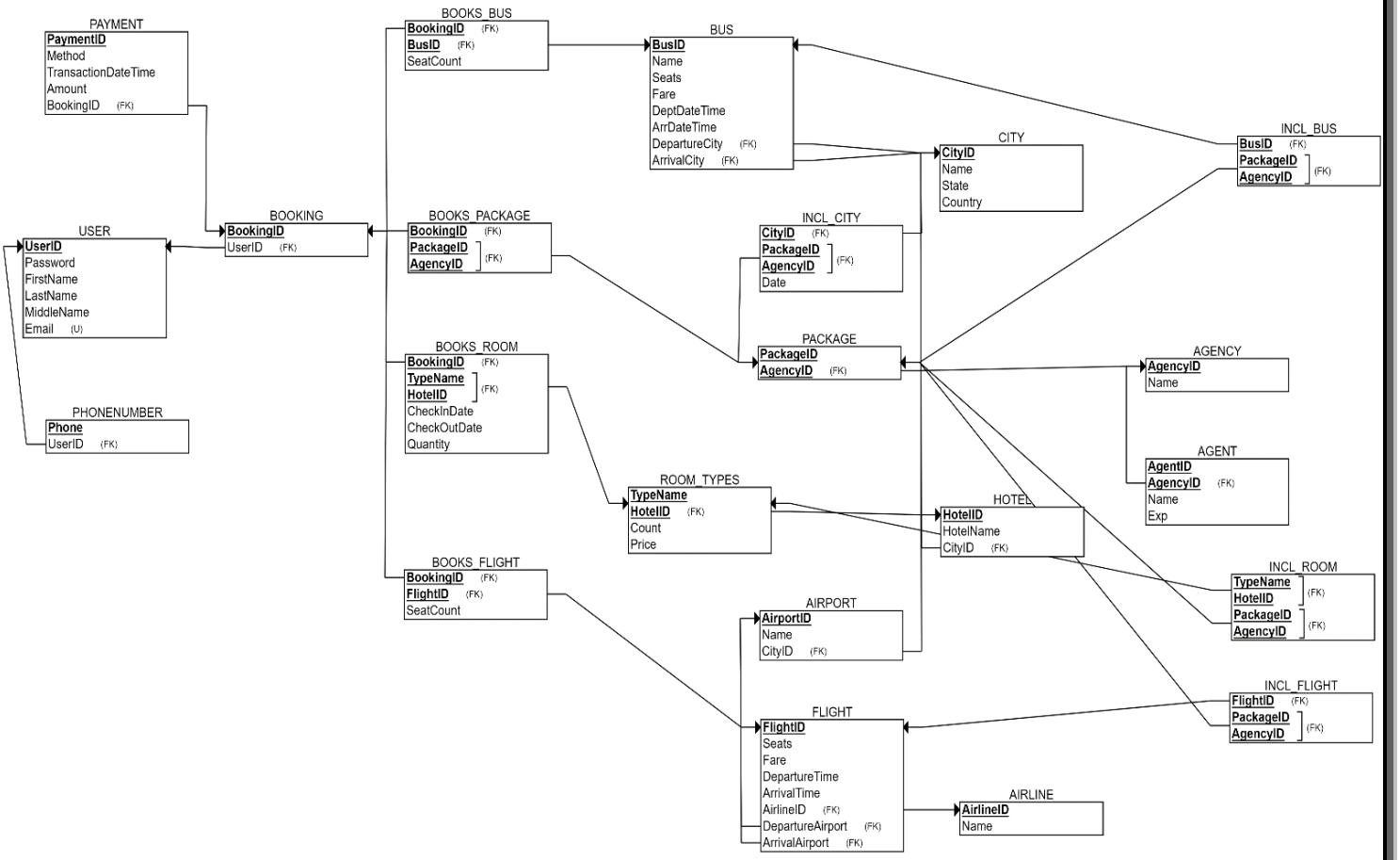
1. PROJECT OVERVIEW

- The A* travel management system is a comprehensive database designed to handle various aspects of travel planning and booking.
- It includes tables for users, flights, hotels, buses, agencies, agents, packages, airlines, cities, airports and related.
- This system facilitates seamless booking of flights, hotels, and buses either individually or as part of tour packages.
- With its extensive features, it provides users with efficient tools for planning and organizing their trips, making it indispensable for travelers and travel agents alike.

2. ENTITY RELATIONSHIP DIAGRAM (ERD)



3. RELATIONAL MODEL



4. ERD EXPLANATION

- **User**: Stores essential user details including name, user ID, email, and password. Passwords are hashed using the SHA-2 algorithm for enhanced security. Each email address is unique, enhancing user identification. This table, a strong entity, also accommodates phone numbers as a multi-valued attribute (MVA).
- **Booking**: Contains essential booking details including booking ID. Each booking is uniquely identified.
- The relationship between Booking and Users is many-to-one, signifying that a user can make multiple bookings. Additionally, Booking exhibits total participation, meaning it is necessary for a booking to be associated with a user ID
- **Payment**: Contains payment details such as payment ID, method, amount, and timestamp.
- The relationship between Payment and Booking is many-to-one, signifying that multiple payments or installments can be associated with a single booking.
- **Airline**: Contains essential details such as airline ID and airline name.
- **Flight**: Includes flight ID, fare, and available seats information.
- The relationship between Flight and Airline is many-to-one, as one airline can operate multiple flights. Additionally, Flight demonstrates total participation, signifying that each flight must be associated with an airline.
- The relationship between Flight and Booking is many-to-many because one booking can involve multiple flights, and conversely, one flight can be part of multiple bookings. This flexibility allows for complex travel itineraries where passengers may book multiple flights for a single trip or where a single flight may be shared among passengers with different bookings. Additionally, this relationship includes the seats booked attribute to specify the number of seats booked for each flight within a booking.
- **City**: Contains details such as city ID, name, state, and country.
- **Airport**: Comprises unique identifier airport ID along with corresponding name.
- The relationship between Airport and City is many-to-one, as one city can accommodate multiple airports. Additionally, Airport demonstrates total participation, ensuring that each airport must be associated with a city.
- The relationship between Flight and Airport for the departure airport is many-to-one, as multiple flights can depart from a single airport. Additionally, Flight connects with Airport again for the arrival airport, which is also many-to-one, signifying that multiple flights can arrive at a single airport. These relationships include attributes such as departure time for the departure airport and arrival time for the arrival airport. Flight exhibits total participation, ensuring that each flight must be associated with an airport.
- **Hotel**: Contains unique identifier hotel ID, pin code and corresponding name.
- The relationship between Hotel and City is many-to-one, reflecting the fact that one city can host multiple hotels.

- **Room_Types**: A dependent weak entity relying on Hotel, Room_Types features attributes such as typename (discriminator), room count, and price.
- The relationship between Room_Types and Hotel is many-to-one, as one hotel can offer multiple room types. Room_Types is a weak entity dependent on Hotel, a strong entity, as it relies on the existence of a hotel for its own existence.
- The relationship between Booking and Room_Types is many-to-many, as one booking can involve multiple room types, and conversely, one room type can be booked by multiple bookings. This relationship includes attributes such as check-in date, check-out date, and room count, to specify the details of each booking, including the number of rooms booked.
- **Bus**: Contains essential details such as bus ID, name, number of seats, and fare.
- The relationship between Bus and City for the departure city is many-to-one, as multiple buses can depart from a single city. Additionally, Bus connects with City again for the arrival city, which is also many-to-one, indicating that multiple buses can arrive at a single city. These relationships include attributes such as departure time for the departure city and arrival time for the arrival city. Bus exhibits total participation, ensuring that each bus trip must be associated with a city.
- The relationship between Bus and Booking is many-to-many, reflecting the flexibility where one booking can involve multiple buses, and conversely, one bus can be part of multiple bookings. This setup caters to scenarios where passengers may book multiple buses for a single trip or where a single bus may serve passengers from different bookings. Additionally, this relationship includes the SeatCount attribute to specify the number of seats booked for each bus within a booking.
- **Package**: A weak entity, dependent on Agency, featuring package ID, which serves as the discriminator.
- The relationship between Package and City is many-to-many, indicating that one package can encompass multiple cities, and conversely, one city can be part of multiple packages. Additionally, this relationship includes a date attribute, specifying the date when a city is visited as part of the package.
- The connection between Package and Booking is many-to-many, allowing for scenarios where one booking may include multiple packages and vice versa.
- **Agency**: Includes agency name and agency ID as its attributes, with agency ID serving as the primary key.
- The weak entity Package relies on Agency as its parent entity, as a package cannot exist without being associated with an agency. This relationship is many-to-one, indicating that multiple packages can be associated with a single agency. A package can be uniquely identified only with the agency ID along with a package ID.
- **Agent**: A weak entity featuring attributes agent ID, experience, and agent name.

- The relationship between Agent and Agency is many-to-one, signifying that multiple agents can be associated with a single agency. Furthermore, the weak entity Agent depends on the agency ID attribute of the Agency entity.
- The relationship between Packages and Agent is one-to-many (N:1), indicating that one agent can serve many packages.
- The relationship between Package and Flight, Bus, and Room is many-to-many, allowing for scenarios where one package can include multiple flights, buses, and rooms, and vice versa.

5. TABLES, FUNCTIONAL DEPENDENCIES, NORMALIZATION

```
CREATE TABLE USER
(
  UserID INT NOT NULL,
  Password VARCHAR(100) NOT NULL,
  FirstName VARCHAR(100) NOT NULL,
  LastName VARCHAR(100),
  MiddleName VARCHAR(100),
  Email VARCHAR(100) NOT NULL,
  PRIMARY KEY (UserID),
  UNIQUE (Email)
);
```

UserID → {FirstName, MiddleName, LastName, Password, Email}
Email → {**UserID**}

Given that **UserID** is a key and **Email** is a candidate key, it can be concluded that the table is in Boyce-Codd Normal Form (BCNF). X → Y and X is a superkey

PhoneNumber → {UserID}

```
CREATE TABLE PHONE_NUMBERS (
  UserID INT NOT NULL,
  PhoneNumber VARCHAR(15) NOT NULL,
  PRIMARY KEY (PhoneNumber),
  FOREIGN KEY (UserID) REFERENCES USER(UserID)
);
```

Given that **PhoneNumber** is the primary key and all functional dependencies are of the form X → Y and X is a superkey, it is satisfying the BCNF condition.

CityID → {Name, State, Country}

```
CREATE TABLE CITY
(
  CityID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  State VARCHAR(100) NOT NULL,
  Country VARCHAR(100) NOT NULL,
  PRIMARY KEY (CityID)
);
```

Given that **CityID** is the primary key and all functional dependencies are of the form X → Y and X is a superkey, it is satisfying the BCNF condition.

AirportID → {Name, **CityID**}

```
CREATE TABLE AIRPORT
(
  AirportID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  CityID INT NOT NULL,
  PRIMARY KEY (AirportID),
  FOREIGN KEY (CityID) REFERENCES CITY(CityID)
);
```

Given that **AirportID** is the primary key and all functional dependencies are of the form X → Y and X is a superkey, it is satisfying the BCNF condition.

HotelID → {HotelName, **CityID**, Pincode}

```
CREATE TABLE HOTEL (
    HotelID INT NOT NULL,
    HotelName VARCHAR(100) NOT NULL,
```

```
    CityID INT NOT NULL,
    Pincode VARCHAR(10),
    PRIMARY KEY (HotelID),
    FOREIGN KEY (CityID) REFERENCES CITY(CityID)
);
```

Given that **HotelID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE ROOM_TYPES
(
  TypeName VARCHAR(100) NOT NULL,
  Count INT NOT NULL,
  Price FLOAT NOT NULL,
  HotelID INT NOT NULL,
  PRIMARY KEY (TypeName, HotelID),
  FOREIGN KEY (HotelID) REFERENCES HOTEL(HotelID)
);
```

$\{\textbf{TypeName, HotelID}\} \rightarrow \{\text{Count, Price}\}$

Given that **TypeName, HotelID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE AGENCY (
  AgencyID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  PRIMARY KEY (AgencyID)
);
```

$\{\textbf{AgencyID}\} \rightarrow \{\text{Name}\}$

Given that **AgencyID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE AGENT
(
  AgentID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  Exp INT NOT NULL,
  AgencyID INT NOT NULL,
  PRIMARY KEY (AgentID, AgencyID),
  FOREIGN KEY (AgencyID) REFERENCES AGENCY(AgencyID)
);
```

$\{\textbf{AgentID, AgencyID}\} \rightarrow \{\text{Name, Exp}\}$

Given that **AgentID, AgencyID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE PACKAGE
(
  PackageID INT NOT NULL,
  Title VARCHAR(100) NOT NULL,
  AgencyID INT NOT NULL,
  PRIMARY KEY (PackageID, AgencyID),
  FOREIGN KEY (AgencyID) REFERENCES AGENCY(AgencyID)
);
```

$\{\textbf{PackageID, AgencyID}\} \rightarrow \{\text{Title}\}$

Given that **PackageID, AgencyID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE AIRLINE
(
  AirlineID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  PRIMARY KEY (AirlineID)
);
```

$\{\textbf{AirlineID}\} \rightarrow \{\text{Name}\}$

Given that **AirlineID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE BUS
(
  BusID INT NOT NULL,
  Name VARCHAR(100) NOT NULL,
  Seats INT NOT NULL,
  Fare FLOAT NOT NULL,
  DeptDateTime DATE NOT NULL,
  ArrDateTime DATE NOT NULL,
  DepartureCity INT NOT NULL,
  ArrivalCity INT NOT NULL,
  PRIMARY KEY (BusID),
  FOREIGN KEY (DepartureCity) REFERENCES CITY(CityID),
  FOREIGN KEY (ArrivalCity) REFERENCES CITY(CityID)
);
```

$\{\textbf{BusID}\} \rightarrow \{\text{Name, Seats, Fare, DeptDateTime, ArrDateTime, DepartureCity, ArrivalCity}\}$

Given that **BusID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE INCL_CITY
(
  ArrDate DATE NOT NULL,
  CityID INT NOT NULL,
  PackageID INT NOT NULL,
  AgencyID INT NOT NULL,
  PRIMARY KEY (CityID, PackageID, AgencyID),
  FOREIGN KEY (CityID) REFERENCES CITY(CityID),
  FOREIGN KEY (PackageID, AgencyID) REFERENCES PACKAGE(PackageID, AgencyID)
);
```

$\{\textbf{CityID, PackageID, AgencyID}\} \rightarrow \{\text{ArrDate}\}$

Given that **CityID, PackageID, AgencyID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE INCL_BUS
(
  BusID INT NOT NULL,
  PackageID INT NOT NULL,
  AgencyID INT NOT NULL,
  PRIMARY KEY (BusID, PackageID, AgencyID),
  FOREIGN KEY (BusID) REFERENCES BUS(BusID),
  FOREIGN KEY (PackageID, AgencyID) REFERENCES PACKAGE(PackageID, AgencyID)
```

$\{\textbf{BusID, PackageID, AgencyID}\} \rightarrow \{\textbf{BusID, PackageID, AgencyID}\}$

As the only functional dependency is a trivial functional dependency, the table is in Boyce-Codd Normal Form (BCNF).

```
);
```

$\{\textbf{TypeName, HotelID, PackageID, AgencyID}\} \rightarrow \{\textbf{TypeName, HotelID, PackageID, AgencyID}\}$

As the only functional dependency is a trivial functional dependency, the table is in Boyce-Codd Normal Form (BCNF).

```
CREATE TABLE INCL_ROOM
(
  TypeName VARCHAR(100) NOT NULL,
  HotelID INT NOT NULL,
  PackageID INT NOT NULL,
  AgencyID INT NOT NULL,
  PRIMARY KEY (TypeName, HotelID, PackageID, AgencyID),
  FOREIGN KEY (TypeName, HotelID) REFERENCES ROOM_TYPES(TypeName, HotelID),
  FOREIGN KEY (PackageID, AgencyID) REFERENCES PACKAGE(PackageID, AgencyID)
);
```

$\{\textbf{BookingID}\} \rightarrow \{\textbf{UserID}\}$

Given that **BookingID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE BOOKING
(
  BookingID INT NOT NULL,
  UserID INT NOT NULL,
  PRIMARY KEY (BookingID),
  FOREIGN KEY (UserID) REFERENCES USER(UserID)
);
```

```
CREATE TABLE PAYMENT
(
  PaymentID INT NOT NULL,
  Method VARCHAR(100) NOT NULL,
  TransactionDateTime DATE NOT NULL,
  Amount FLOAT NOT NULL,
  BookingID INT NOT NULL,
  PRIMARY KEY (PaymentID),
  FOREIGN KEY (BookingID) REFERENCES BOOKING(BookingID)
);
```

$\{\textbf{PaymentID}\} \rightarrow \{\textbf{BookingID}, \text{Method, Amount, TransactionDateTime}\}$

Given that **PaymentID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE BOOKS_BUS
(
  BookingID INT NOT NULL,
  BusID INT NOT NULL,
  SeatCount INT NOT NULL,
  PRIMARY KEY (BookingID, BusID),
  FOREIGN KEY (BookingID) REFERENCES BOOKING(BookingID),
  FOREIGN KEY (BusID) REFERENCES BUS(BusID)
);
```

$\{\textbf{BookingID, BusID}\} \rightarrow \{\text{SeatCount}\}$

Given that **BookingID, BusID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE BOOKS_ROOM
(
  CheckInDate DATE NOT NULL,
  CheckOutDate DATE NOT NULL,
  Quantity INT NOT NULL,
  BookingID INT NOT NULL,
  TypeName VARCHAR(100) NOT NULL,
  HotelID INT NOT NULL,
  PRIMARY KEY (BookingID, TypeName, HotelID),
  FOREIGN KEY (BookingID) REFERENCES BOOKING(BookingID),
  FOREIGN KEY (TypeName, HotelID) REFERENCES ROOM_TYPES(TypeName, HotelID)
);
```

$\{\textbf{BookingID, TypeName, HotelID}\} \rightarrow \{\text{Quantity, CheckInDate, CheckOutDate}\}$

Given that **BookingID, TypeName, HotelID** is the primary key and all functional dependencies are of the form X → Y and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE BOOKS_PACKAGE
(
  BookingID INT NOT NULL,
  PackageID INT NOT NULL,
  AgencyID INT NOT NULL,
  PRIMARY KEY (BookingID, PackageID, AgencyID),
  FOREIGN KEY (BookingID) REFERENCES BOOKING(BookingID),
  FOREIGN KEY (PackageID, AgencyID) REFERENCES PACKAGE(PackageID, AgencyID)
);
```

$\{\textbf{BookingID, PackageID, AgencyID}\} \rightarrow \{\textbf{BookingID, PackageID, AgencyID}\}$

As the only functional dependency is a trivial functional dependency, the table is in Boyce-Codd Normal Form (BCNF).

```
CREATE TABLE FLIGHT
(
  FlightID INT NOT NULL,
  Seats INT NOT NULL,
  Fare FLOAT NOT NULL,
  DepartureTime DATE NOT NULL,
  ArrivalTime DATE NOT NULL,
  AirlineID INT NOT NULL,
  DepartureAirport INT NOT NULL,
  ArrivalAirport INT NOT NULL,
  PRIMARY KEY (FlightID),
  FOREIGN KEY (AirlineID) REFERENCES AIRLINE(AirlineID),
  FOREIGN KEY (DepartureAirport) REFERENCES AIRPORT(AirportID),
  FOREIGN KEY (ArrivalAirport) REFERENCES AIRPORT(AirportID)
);
```

$\{\textbf{FlightID}\} \rightarrow \{\textbf{AirlineID}, \text{Seats, Fare, DepartureTime, ArrivalTime, DepartureAirport, ArrivalAirport}\}$

Given that **FlightID** is the primary key and all functional dependencies are of the form X → Y and X is a superkey, it is satisfying the BCNF condition.

```
CREATE TABLE INCL_FLIGHT
(
  FlightID INT NOT NULL,
  PackageID INT NOT NULL,
  AgencyID INT NOT NULL,
  PRIMARY KEY (FlightID, PackageID, AgencyID),
  FOREIGN KEY (FlightID) REFERENCES FLIGHT(FlightID),
  FOREIGN KEY (PackageID, AgencyID) REFERENCES PACKAGE(PackageID, AgencyID)
);
```

$\{\textbf{PackageID, AgencyID, FlightID}\} \rightarrow \{\textbf{PackageID, AgencyID, FlightID}\}$

As the only functional dependency is a trivial functional dependency, the table is in Boyce-Codd Normal Form (BCNF).

```
CREATE TABLE BOOKS_FLIGHT
(
  BookingID INT NOT NULL,
  FlightID INT NOT NULL,
  SeatCount INT NOT NULL,
  PRIMARY KEY (BookingID, FlightID),
  FOREIGN KEY (BookingID) REFERENCES BOOKING(BookingID),
  FOREIGN KEY (FlightID) REFERENCES FLIGHT(FlightID)
);
```

$\{\textbf{BookingID, FlightID}\} \rightarrow \{\text{SeatCount}\}$

Given that **BookingID, FlightID** is the primary key and all functional dependencies are of the form $X \rightarrow Y$ and X is a superkey, it is satisfying the BCNF condition.

6. DATA INSERTION

```
INSERT INTO USER (UserId, Password, FirstName, LastName, MiddleName, Email)
VALUES
(1, SHA2('password1', 256), 'Chetan', 'Kar', NULL, 'chetankar65@gmail.com'),
(2, SHA2('password2', 256), 'Aarav', 'Patel', 'Kumar',
'aarav.patel@example.com'),
(3, SHA2('password3', 256), 'Ananya', 'Sharma', 'Singh',
'ananya.sharma@example.com'),
(4, SHA2('password4', 256), 'Aryan', 'Das', 'Gupta', 'aryan.das@example.com'),
(5, SHA2('password5', 256), 'Neha', NULL, NULL, 'neha.joshi@example.com'),
(6, SHA2('password6', 256), 'Rahul', 'Gupta', 'Sharma',
'rahul.gupta@example.com'),
(7, SHA2('password7', 256), 'Priya', NULL, 'Kumari',
'priya.verma@example.com'),
(8, SHA2('password8', 256), 'Ravi', 'Malhotra', 'Singh',
'ravi.malhotra@example.com'),
(9, SHA2('password9', 256), 'Sneha', 'Choudhary', 'Yadav',
'sneha.choudhary@example.com'),
(10, SHA2('password10', 256), 'Shubham', 'Pahilwani', NULL,
'shubhampahilwani1@gmail.com');

INSERT INTO PHONE_NUMBERS (UserID, PhoneNumber)
VALUES
    (1, '1234567890'), -- Chetan Kar
    (1, '9876543210'), -- Chetan Kar (multiple phone numbers)
    (2, '2345678901'), -- Aarav Patel
    (3, '3456789012'), -- Ananya Sharma
    (4, '4567890123'), -- Aryan Das
    (5, '5678901234'), -- Neha Joshi
    (6, '6789012345'), -- Rahul Gupta
    (7, '7890123456'), -- Priya Verma
    (8, '8901234567'), -- Ravi Malhotra
    (9, '9012345678'), -- Sneha Choudhary
    (10, '0123456789'); -- Shubham Pahilwani

INSERT INTO CITY (CityID, Name, State, Country)
VALUES
(1, 'Bengaluru', 'Karnataka', 'India'),
(2, 'Mumbai', 'Maharashtra', 'India'),
(3, 'Hyderabad', 'Telangana', 'India'),
(4, 'Dubai', 'Emirate of Dubai', 'United Arab Emirates'),
```

```
(5, 'New York City', 'New York', 'United States of America'),
(6, 'London', 'London', 'United Kingdom'),
(7, 'San Francisco', 'California', 'United States of America'),
(8, 'Jaipur', 'Rajasthan', 'India'),
(9, 'Kolkata', 'West Bengal', 'India'),
(10, 'Chennai', 'Tamil Nadu', 'India');

INSERT INTO Airport
(AirportID, Name, CityID) VALUES
(1, 'Kempegowda International Airport', 1),
(2, 'Chatrapati Shivaji International Airport', 2),
(3, 'Rajiv Gandhi International Airport', 3),
(4, 'Dubai International Airport', 4),
(5, 'Heathrow International Airport', 5),
(6, 'JFK International Airport', 6),
(7, 'Sanfrancisco International Airport', 7),
(8, 'Jaipur International Airport', 8),
(9, 'DV Patil International Airport', 2),
(10, 'Dumdum Airport', 9);

INSERT INTO HOTEL (HotelID, HotelName, CityID, Pincode)
VALUES
(1, 'Taj Mahal Palace', 2, '400001'),
(2, 'ITC Gardenia', 1, '560001'),
(3, 'Taj Falaknuma Palace', 3, '500001'),
(4, 'JW Marriott Marquis Hotel Dubai', 4, '123456'),
(5, 'The Plaza Hotel', 5, '10001'),
(6, 'The Ritz London', 6, 'W1J 9BR'),
(7, 'The St. Regis San Francisco', 7, '94103'),
(8, 'Fairmont Jaipur', 8, '302002'),
(9, 'The Oberoi Bengaluru', 1, '560001'),
(10, 'Taj Lands End, Mumbai', 2, '400050'),
(11, 'ITC Kakatiya, Hyderabad', 3, '500082'),
(12, 'Burj Al Arab Jumeirah', 4, '123456'),
(13, 'The Peninsula New York', 5, '10019'),
(14, 'The Langham London', 6, 'SE1 1UN'),
(15, 'Hotel Nikko San Francisco', 7, '94108'),
(16, 'Rambagh Palace, Jaipur', 8, '302005'),
(17, 'JW Marriot Chennai', 10, '600008');
```

```sql
INSERT INTO ROOM_TYPES (TypeName, Count, Price, HotelID)
VALUES ('Luxury Suite', 10, 500.00, 1),
('Deluxe Room', 20, 250.00, 1),
('Executive Suite', 5, 800.00, 1),
('Garden View Room', 15, 300.00, 2),
('Executive Club Room', 10, 400.00, 2),
('Presidential Suite', 3, 1200.00, 2),
('Grand Royal Suite', 5, 1500.00, 3),
('Heritage Room', 20, 600.00, 3),
('Nizam Suite', 3, 2000.00, 3),
('Deluxe Room', 30, 400.00, 4),
('Executive Suite', 15, 800.00, 4),
('Royal Suite', 5, 1500.00, 4),
('Plaza Suite', 10, 1000.00, 5),
('Grand Suite', 20, 700.00, 5),
('Deluxe Room', 50, 400.00, 5),
('Junior Suite', 8, 900.00, 6),
('Executive Room', 15, 600.00, 6),
('Penthouse Suite', 3, 2500.00, 6),
('Superior Room', 25, 500.00, 7),
('Executive Suite', 10, 1000.00, 7),
('St. Regis Suite', 5, 1500.00, 7),
('Royal Suite', 5, 1200.00, 8),
('Deluxe Room', 30, 400.00, 8),
('Luxury Tent', 10, 800.00, 8),
('Deluxe Room', 45, 900.00, 17);

INSERT INTO AGENCY (AgencyID, Name)
VALUES
(1, 'Thomas Cook'),
(2, 'MakeMyTrip'),
(3, 'Cox & Kings'),
(4, 'Expedia'),
(5, 'Travelocity'),
(6, 'Goibibo'),
(7, 'Yatra.com'),
(8, 'Cleartrip');

INSERT INTO AGENT (AgentID, Name, Exp, AgencyID)
VALUES
(1, 'Aarav Mehta', 5, 1),     -- Thomas Cook
```

```
(2, 'Anjali Singhania', 7, 2), -- MakeMyTrip
(3, 'Rohan Khanna', 6, 3),    -- Cox & Kings
(4, 'Shreya Patel', 8, 4),     -- Expedia
(5, 'Amit Kumar', 4, 5),       -- Travelocity
(6, 'Kavita Sharma', 9, 6),    -- Goibibo
(7, 'Rajeev Desai', 3, 7),     -- Yatra.com
(8, 'Sneha Gupta', 6, 8);      -- Cleartrip

INSERT INTO PACKAGE (PackageID, Title, AgencyID)
VALUES
(1, 'Golden Triangle Tour', 1),  -- Thomas Cook
(2, 'Andaman Adventure', 2),      -- MakeMyTrip
(3, 'Goa Beach Getaway', 3),      -- Cox & Kings
(4, 'Dubai Desert Safari', 4),    -- Expedia
(5, 'New York City Explorer', 5), -- Travelocity
(6, 'London Theater Experience', 6), -- Goibibo
(7, 'San Francisco Bay Cruise', 7), -- Yatra.com
(8, 'Rajasthan Heritage Tour', 8), -- Cleartrip
(9, 'Goa Family Pack', 1),    -- Thomas Cook
(10, 'South India special package', 2);    -- MakeMyTrip

INSERT INTO AIRLINE (AirlineID, Name)
VALUES
(1, 'Emirates'),
(2, 'Singapore Airlines'),
(3, 'Qatar Airways'),
(4, 'Cathay Pacific'),
(5, 'British Airways'),
(6, 'Lufthansa'),
(7, 'Air France'),
(8, 'Delta Air Lines'),
(9, 'IndiGo'),
(10, 'Vistara'),
(11, 'Spicejet'),
(12, 'Air India');

INSERT INTO BUS (BusID, Name, Seats, Fare, DeptDateTime, ArrDateTime,
DepartureCity, ArrivalCity)
VALUES
(7273, 'Morning Star Travels', 40, 2050, '2024-04-10 07:00:00', '2024-04-10
15:00:00', 3, 1), -- Hyd to BLR
```

```
(7689, 'Morning Star Travels', 40, 2090, '2024-04-11 08:00:00', '2024-04-11
16:00:00', 1, 3), -- BLR to Hyd
(8383, 'Vinayak Travels', 47, 1990, '2024-04-12 09:00:00', '2024-04-12
17:00:00', 1, 2), -- Blr to Mum
(6111, 'TSRTC', 53, 3500, '2024-04-13 10:00:00', '2024-04-13 18:00:00', 3, 2),
-- Hyd to Mumbai
(4555, 'MSRTC', 53, 3900, '2024-04-14 11:00:00', '2024-04-14 19:00:00', 2, 3),
-- Mumbai to Hyd
(3001, 'TSRTC', 60, 1900, '2024-04-14 11:00:00', '2024-04-14 19:00:00', 1, 3),
-- blr to hyd
(3393, 'KSRTC', 65, 1500, '2024-04-16 12:00:00', '2024-04-14 16:00:00', 1, 10);
-- blr to chennai

INSERT INTO INCL_CITY
VALUES
('2024-04-10', 2, 1, 1),
('2024-04-12', 8, 1, 1),
('2024-04-14', 9, 1, 1),
('2024-04-16', 10, 1, 1),
('2024-04-11', 5, 5, 5),
('2024-04-13', 10, 10, 2),
('2024-04-15', 1, 10, 2),
('2024-04-17', 3, 10, 2),
('2024-04-19', 2, 10, 2),
('2024-05-20', 6, 6, 6);

-- South India package from Hyderabad to Bangalore
INSERT INTO INCL_BUS
VALUES
(7689, 10, 2),
(3393, 10, 2);

-- Bengaluru (under packageID 10, agencyID 2)
INSERT INTO INCL_ROOM (TypeName, HotelID, PackageID, AgencyID)
VALUES ('Presidential Suite', 2, 10, 2);

-- Mumbai (under packageIDs 1 and 10, agencyIDs 1 and 2)
INSERT INTO INCL_ROOM (TypeName, HotelID, PackageID, AgencyID)
VALUES ('Luxury Suite', 1, 1, 1),
       ('Luxury Suite', 1, 10, 2);
```

```sql
-- Jaipur (under packageID 8, agencyID 8)
INSERT INTO INCL_ROOM (TypeName, HotelID, PackageID, AgencyID)
VALUES ('Luxury Tent', 8, 8, 8);

-- London (under packageID 6, agencyID 6)
INSERT INTO INCL_ROOM (TypeName, HotelID, PackageID, AgencyID)
VALUES ('Junior Suite', 6, 6, 6);

-- San Francisco (Under packageID 7, AgencyID 7)
INSERT INTO INCL_ROOM (TypeName, HotelID, PackageID, AgencyID)
VALUES ('Executive Suite', 7, 7, 7);

-- Chennai (Under packageID 10, agencyID 2)
INSERT INTO INCL_ROOM (TypeName, HotelID, PackageID, AgencyID)
VALUES ('Deluxe Room', 17, 10, 2);

INSERT INTO BOOKING (BookingID, UserID)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);

INSERT INTO PAYMENT (PaymentID, Method, TransactionDateTime, Amount, BookingID)
VALUES
(1, 'UPI', '2024-4-5 15:00', 7000, 1),
(2, 'Debit', '2024-4-5 15:00', 9000, 2),
(3, 'Credit', '2024-4-5 15:00', 4000, 3),
(4, 'Netbanking', '2024-4-5 15:00', 6000, 4),
(5, 'UPI', '2024-4-5 15:00', 7000, 5);

INSERT INTO BOOKS_BUS (BookingID, BusID, SeatCount)
VALUES
(1, 3001, 2),
(2, 3393, 1);

INSERT INTO BOOKS_ROOM (CheckInDate, CheckOutDate, Quantity, BookingId,
TypeName, HotelId)
VALUES
('2024-4-13', '2024-4-15', 1, 1, 'Nizam Suite', 3),
```

```
('2024-4-13', '2024-4-15', 2, 2, 'Executive Club Room', 2);

INSERT INTO BOOKS_PACKAGE (BookingID, PackageID, AgencyID)
VALUES
(3, 1, 1),
(4, 2, 2);

-- Blr to Mumbai (Emirates)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(1, 150, 5000.00, '2024-04-10', '2024-04-10', 1, 1, 2);

-- Blr to London (British Airways)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(2, 200, 12000.00, '2024-04-11', '2024-04-11', 5, 1, 5);

-- Blr to San Francisco (Lufthansa)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(3, 180, 15000.00, '2024-04-12', '2024-04-12', 6, 1, 7);

-- Blr to Hyderabad (IndiGo)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(4, 160, 4000.00, '2024-04-13', '2024-04-13', 9, 1, 3);

-- Blr to Chennai (Spicejet)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(5, 150, 3500.00, '2024-04-14', '2024-04-14', 11, 1, 10);

-- Hyd to San Francisco (Air India)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
```

```
(6, 180, 18000.00, '2024-04-15', '2024-04-15', 12, 3, 7);

-- Jaipur to Bengaluru (Vistara)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(7, 140, 6000.00, '2024-04-16', '2024-04-16', 10, 8, 1);

-- Jaipur to Dubai (Emirates)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(8, 200, 18000.00, '2024-04-17', '2024-04-17', 1, 8, 4);

-- Mumbai to Bangalore (IndiGo)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(9, 180, 4500.00, '2024-04-18', '2024-04-18', 9, 2, 1);

-- Mumbai to Kolkata (Air India)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(11, 180, 6000.00, '2024-04-15', '2024-04-15', 12, 2, 3);

-- Chennai to Kolkata (SpiceJet)
INSERT INTO FLIGHT (FlightID, Seats, Fare, DepartureTime, ArrivalTime,
AirlineID, DepartureAirport, ArrivalAirport)
VALUES
(10, 160, 5500.00, '2024-04-19', '2024-04-19', 11, 10, 3);

INSERT INTO INCL_FLIGHT (FlightId, PackageId, AgencyId)
VALUES
(2, 6, 6),
(3, 7, 7),
(9, 10, 2),
(8, 4, 4),
(10, 1, 1),
(11, 1, 1);
```

```sql
INSERT INTO BOOKS_FLIGHT
VALUES
(4, 1, 1),
(5, 1, 2);
```

7. SAMPLE QUERIES

```
-- What packages are available?
SELECT * FROM Package;
```

PackageID	Title	AgencyID
1	Golden Triangle Tour	1
2	Andaman Adventure	2
3	Goa Beach Getaway	3
4	Dubai Desert Safari	4
5	New York City Explorer	5
6	London Theater Experience	6
7	San Francisco Bay Cruise	7
8	Rajasthan Heritage Tour	8
9	Goa Family Pack	1
10	South India special package	2
NULL	NULL	NULL

```
-- What cities are covered in individial packages?
SELECT Agency.Name as Agency_Name, Package.Title as Package_title, City.Name as
Cities_covered FROM Package
INNER JOIN Agency on Package.AgencyID = Agency.AgencyID
INNER JOIN INCL_CITY on Package.PackageID = INCL_CITY.PackageID
INNER JOIN CITY on INCL_CITY.CityID = City.CityID
ORDER BY (Package.Title);
```

Agency_Name	Package_title	Cities_covered
Thomas Cook	Golden Triangle Tour	Mumbai
Thomas Cook	Golden Triangle Tour	Jaipur
Thomas Cook	Golden Triangle Tour	Kolkata
Thomas Cook	Golden Triangle Tour	Chennai
Goibibo	London Theater Experience	London
Travelocity	New York City Explorer	New York City
MakeMyTrip	South India special package	Bengaluru
MakeMyTrip	South India special package	Mumbai
MakeMyTrip	South India special package	Hyderabad
MakeMyTrip	South India special package	Chennai

```
-- What hotels are covered in individual packages?
SELECT Agency.Name as Agency_Name, Package.Title as Package_title,
Hotel.HotelName AS Hotel_name, City.Name as City_name FROM Package
INNER JOIN Agency on Package.AgencyID = Agency.AgencyID
INNER JOIN INCL_ROOM on Package.PackageID = INCL_ROOM.PackageID
```

```
INNER JOIN HOTEL on INCL_ROOM.HotelID = HOTEL.CityID
INNER JOIN CITY on Hotel.CityID = City.CityID
ORDER BY (Package.Title);
```

Agency_Name	Package_title	Hotel_name	City_name
Thomas Cook	Golden Triangle Tour	ITC Gardenia	Bengaluru
Thomas Cook	Golden Triangle Tour	The Oberoi Bengaluru	Bengaluru
Goibibo	London Theater Experience	The Ritz London	London
Goibibo	London Theater Experience	The Langham London	London
Cleartrip	Rajasthan Heritage Tour	Fairmont Jaipur	Jaipur
Cleartrip	Rajasthan Heritage Tour	Rambagh Palace, Jaipur	Jaipur
Yatra.com	San Francisco Bay Cruise	The St. Regis San Francisco	San Francisco
Yatra.com	San Francisco Bay Cruise	Hotel Nikko San Francisco	San Francisco
MakeMyTrip	South India special package	ITC Gardenia	Bengaluru
MakeMyTrip	South India special package	The Oberoi Bengaluru	Bengaluru
MakeMyTrip	South India special package	Taj Mahal Palace	Mumbai
MakeMyTrip	South India special package	Taj Lands End, Mumbai	Mumbai

```
-- What flights are covered in packages?
SELECT Agency.Name as Agency_Name, Package.Title as Package_title, Airline.Name
as Airline FROM Package
INNER JOIN Agency on Package.AgencyID = Agency.AgencyID
INNER JOIN INCL_FLIGHT on Package.PackageID = INCL_FLIGHT.PackageID
INNER JOIN FLIGHT on INCL_FLIGHT.FlightID = Flight.FlightID
INNER JOIN AIRLINE on FLIGHT.AirlineID = Airline.AirlineID
ORDER BY (Package.Title);
```

Agency_Name	Package_title	Airline
Expedia	Dubai Desert Safari	Emirates
Thomas Cook	Golden Triangle Tour	Spicejet
Thomas Cook	Golden Triangle Tour	Air India
Goibibo	London Theater Experience	British Airways
Yatra.com	San Francisco Bay Cruise	Lufthansa
MakeMyTrip	South India special package	IndiGo

```
-- Display flight details of flights booked by Aryan Gupta.
SELECT Flight.FlightID, Airline.Name, A1.Name, A2.Name FROM Flight
INNER JOIN BOOKS_FLIGHT on Flight.FlightID = BOOKS_FLIGHT.FlightID
INNER JOIN Airline on Flight.AirlineID = Airline.AirlineID
INNER JOIN BOOKING on BOOKS_FLIGHT.BookingID = BOOKING.BookingID
INNER JOIN Airport A1 on A1.AirportID = Flight.DepartureAirport
INNER JOIN Airport A2 on A2.AirportID = Flight.ArrivalAirport
```

```
WHERE BOOKING.UserID = (SELECT UserID FROM USER WHERE email =
'aryan.das@example.com');
```

FlightID	Name	Name	Name
1	Emirates	Kempegowda International Airport	Chatrapati Shivaji International Airport

```
-- Display hotels along with roomtypes present in Bengaluru, Mumbai and
Chennai.
SELECT Hotel.HotelName AS Hotel_name, City.Name AS City, ROOM_TYPES.TypeName as
Room_type, ROOM_TYPES.Count AS Rooms_available,
ROOM_TYPES.Price as Price_per_night FROM ROOM_TYPES
JOIN Hotel on Hotel.HotelId = ROOM_TYPES.HotelID
JOIN City on City.CityID = Hotel.CityID
WHERE City.CityID in (Select CityID from CITY where City.Name = 'Bengaluru' or
City.Name = 'Mumbai'
or City.Name = 'Chennai');
```

Hotel_name	City	Room_type	Rooms_available	Price_per_night
ITC Gardenia	Bengaluru	Executive Club Room	10	400
ITC Gardenia	Bengaluru	Garden View Room	15	300
ITC Gardenia	Bengaluru	Presidential Suite	3	1200
Taj Mahal Palace	Mumbai	Deluxe Room	20	250
Taj Mahal Palace	Mumbai	Executive Suite	5	800
Taj Mahal Palace	Mumbai	Luxury Suite	10	500
JW Marriot Chennai	Chennai	Deluxe Room	45	900

```
-- Display average price of hotel rooms present in hotels in Bengaluru.
Select City.Name, ROUND(AVG(ROOM_TYPES.Price), 2) AS Avg_cost_per_night FROM
ROOM_TYPES
INNER JOIN HOTEL ON Hotel.HotelID = ROOM_TYPES.HotelID
INNER JOIN CITY ON City.CityID = Hotel.CityID
WHERE City.CityID = (Select CityId from City where City.Name = 'Bengaluru')
GROUP BY City.CityID;
```

Name	Avg_cost_per_night
Bengaluru	633.33

```
-- Display Hotel rooms in bengaluru that have below average price.
Select Hotel.HotelName, ROOM_TYPES.TypeName, ROOM_TYPES.Price FROM ROOM_TYPES
INNER JOIN HOTEL ON Hotel.HotelID = ROOM_TYPES.HotelID
INNER JOIN CITY ON City.CityID = Hotel.CityID
WHERE City.CityID = (Select CityId from City where City.Name = 'Bengaluru')
```

```
AND ROOM_TYPES.Price < (SELECT ROUND(AVG(ROOM_TYPES.Price), 2) AS
Avg_cost_per_night FROM ROOM_TYPES
INNER JOIN HOTEL ON Hotel.HotelID = ROOM_TYPES.HotelID
INNER JOIN CITY ON City.CityID = Hotel.CityID
WHERE City.CityID = (Select CityId from City where City.Name = 'Bengaluru')
GROUP BY City.CityID);
```

	HotelName	TypeName	Price
▶	ITC Gardenia	Executive Club Room	400
	ITC Gardenia	Garden View Room	300

8. LEARNING AND OUTCOMES

- **Database design and data organization:** This project displays how to design an efficient database and how to structure and organize the data in a travel planning system. It highlights the importance of properly structuring tables and establishing meaningful relationships between them.
- **Efficient data retrieval:** The database design supports efficient data retrieval through appropriate indexing and query optimization. It provides fast and reliable information and improves system performance as a whole.
- **Normalization:** The use of normalized tables helps reduce data redundancy and maintains data consistency.
- **Entity-relationship modelling:** The database design shows the various relationships between entities and their associated
attributes. These relationships result in smooth flow of data within the system.
- **Data integrity and constraints:** The use of keys (primary and foreign keys) helps maintain data integrity. It also enforces a high level of data consistency throughout the database.
- **Data analysis:** This database stores various type of data such as users, hotels, airports, buses, bookings, packages which can be used for a wide range of data analysis. Various types of information such as which city is visited most, which airport is the busiest, which hotels are popular in a particular city etc can be performed. The database has been designed and fine tuned for such forms of data analysis.
- **Scalability**: The modular design of the database allows the possibility of scalability in the future. New features can be added by making additional entities and establishing proper relationships.
- **Integration:** The database can easily be integrated with a GUI/frontend, as well as external systems providing real-time updates like abrupt flight cancellations, fluctuations in hotel room prices etc.

Overall, this database builds a foundation for a robust, efficient and scalable travel planning and itinerary system.

9. CONCLUSION

In conclusion, the A* database project demonstrates effective organization, management and analysis of data. This project implements design and normalization principles, entity-relationship modelling and data integrity constraints to ensure a functional and secure database. By implementing a relational database schema, the project ensures efficient data retrieval and analysis. Overall, this project highlights the importance of planning and design to build a system that is efficient, robust, secure and scalable, and provides insights into the complexities of handling data.