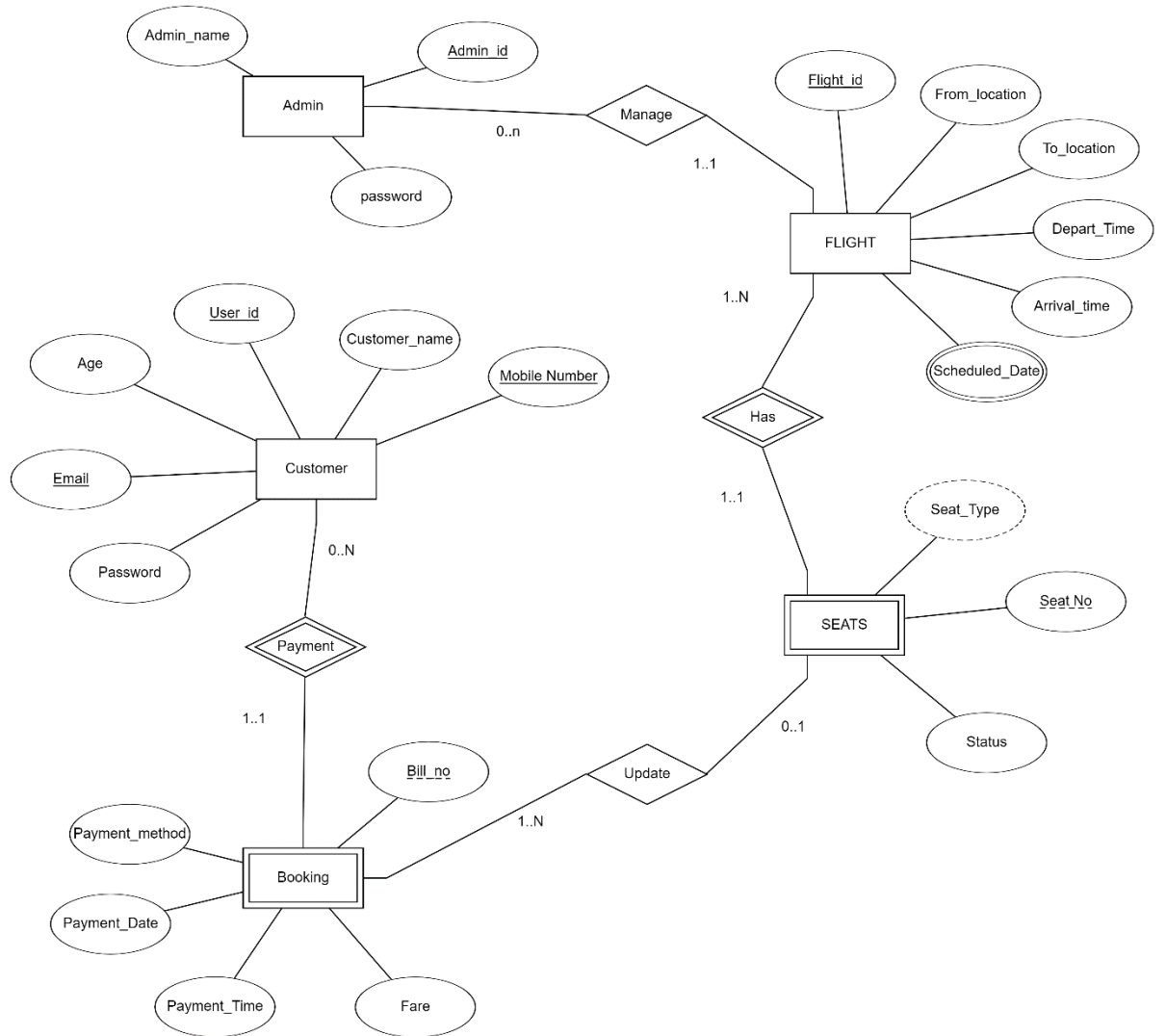


ER DIAGRAM :



TABLES

1. ADMIN (Admin_name , Admin_id , Password)

Primary key : Admin_id

Functional dependencies : Admin_id-> Admin_name , Admin_id , Password

Only one functional dependency is present and super key(Admin_id) is deriving it.

Hence in BCNF.

2. Customer (User_id , Customer_name , Mobile_Number , Age , Email , Password)

Candidate Keys : User_id or Mobile_Number or Email .

Primary key : User_id

Functional dependencies : User_id -> Customer_name , Mobile_Number , Age , Email , Password

Customer_name -> User_id , Mobile_Number , Age , Email , Password

Mobile_Number-> User_id , Customer_name, Age , Email , Password

All the three : User_id , Customer_name , Mobile_Number are super keys.

As super keys are deriving other attributes the table is in BCNF.

3. **FLIGHT** (Flight_id , From_location , To_loaction , Depart_Time , Arrival_time ,Admin_id)

Primary key : Flight_id

Functional dependencies : Flight_id -> Flight_id , From_location , To_loaction , Depart_Time , Arrival_time, Admin_id

Foreign key : Admin_id connecting to Admin Table(Primary table).

We have Scheduled_Date which is multi value attribute. Only one functional dependency is present and super key(Flight_id) is deriving it. Hence in BCNF.

4. **FLIGHT_DATE** (Flight_id , Scheduled_date)

Primary key : Flight_id , Scheduled_date
No functional dependencies.
Hence the table is in BCNF.

5. **SEATS** (Seat_No , Status , Flight_id , Scheduled_Date)

Partial key : Seat_No.
Primary key : {Seat_No, Flight_id , Scheduled_Date}.
Functional Dependencies : {Seat_No, Flight_id ,
Scheduled_Date }-> Status

Only one functional dependency is present and super key(Seat_No, Flight_id , Scheduled_Date) is deriving it.
Hence in BCNF.

Since Seat_Type is a derived entity from Seat_no no need to mention in the table.

6. **BOOKING_SEAT** (Seat No , Flight id , Schedule Date , Customer_id , Bill_no)

Primary Key : {Seat_No , Flight_id , Schedule_Date}
Foreign Key :

{Seat_No , Flight_id , Schedule_Date} pointing to the Seats Table.

{Customer_id , Bill_no} pointing to Booking table.

It's in BCNF.

7. BOOKING (Bill_no , Payment_method ,
Payment_Date , Payment_Time , Fare,
User_id)

Partial key : Bill_no

Primary key : {Bill_no , User_id }

Functional dependencies : {Bill_no , User_id } ->
Payment_method , Payment_Date , Payment_Time
, Fare

Only one functional dependency is present and super key
{Bill_no , User_id } is deriving it.
Hence in BCNF.

RELATIONS

1. Between Admin and Flight : MANAGES

It's one to many relationship where there's a total participation from Flight side.

We need not create a separate table for this relation we can use Admin_id as foreign key in FLIGHT Table. (Admin_id is the primary key of ADMIN table).

2. Between Flight and Seats : HAS

It's an identifying relation where Flight is the strong entity and Seats is the weak entity.

We need not create a separate table for identifying relation.

Seat_number is the partial key for the SEATS table.

Seat_number , Scheduled_Date , Flight_id all three would be the primary key of SEATS table.

3. Between Seats and Booking: UPDATE

It's a one to many relation where the total participation is on the many side. We will create a new table as with foreign key concept we can't manage the relation.

Booking_Seats table is created with attributes : {Seat_no , Flight_id , Schedule_Date } as primary key .

4. Between Customer and Booking :

PAYMENT

It's a identifying relation where Customer is the strong entity and Booking is the weak entity. We need not create a separate table for identifying relation. Here Bill_no is the partial key for Booking table. Hence we need to include the PK of Customer in the table of Booking.

Hence the primary key of Booking table would be : Bill_no , User_id .

RELATIONAL MODEL :

