AIRLINE TICKETING SYSTEM

DATABASE MANAGEMENT SYSTEM PROJECT

Nandini Maharaj – 21011A0565
Aditya Sudarsan Anand – 21011A0564
Aditya Mandal – 21011A0562
Aditya Jha – 21011A0561
BTech CSE-Regular (2-2) | JNTUH-UCESTH

SOFTWARE REQUIREMENT SPECIFICATION

1. Introduction

The airline ticketing system is an online platform designed to facilitate ticket bookings for air travel. The system will allow users to search, book, and purchase airline tickets. This document outlines the features, functions, and requirements of the system.

2. Scope

The system will be accessible to customers nationwide, and it will provide access to a wide range of airlines, flights, and travel routes. The system will also allow customers to select their preferred seats and manage their booking details.

3. Requirements

The following are the requirements for the airline ticketing system:

- User Management: The system will have a user management module that allows users to create and manage their accounts. Users will be able to log in, view their booking history, and manage their personal information.
- Flight Search: The system will allow users to search for flights based on departure and arrival locations, travel dates, and the number of passengers.
- Flight Booking: The system will allow users to select flights, book tickets, and make payments securely. The system will also generate a booking confirmation with a unique booking reference number.
- Seat Selection: The system will allow users to select their preferred seats on the plane. Users will be able to choose their seats based on availability.
- Booking Management: The system will allow users to manage their bookings, make changes to their flights, and cancel their bookings if required.
- Admin Dashboard: The system will have an admin dashboard that allows the system administrators to manage the system's users, flights, and bookings.

4. Non-Functional Requirements

- The system will be highly available and accessible at all times.

- The system will be scalable and able to handle a large number of users and bookings. - The system will be secure and protect user data from unauthorised access.

- The system will be user-friendly and easy to use for both customers and system administrators.

5. Constraints

- The system will be developed using modern web technologies.

- The system will be hosted on a reliable and scalable cloud infrastructure.

- The system will comply with all relevant regulations and standards for online payment processing and data protection.

6. Assumptions and Dependencies

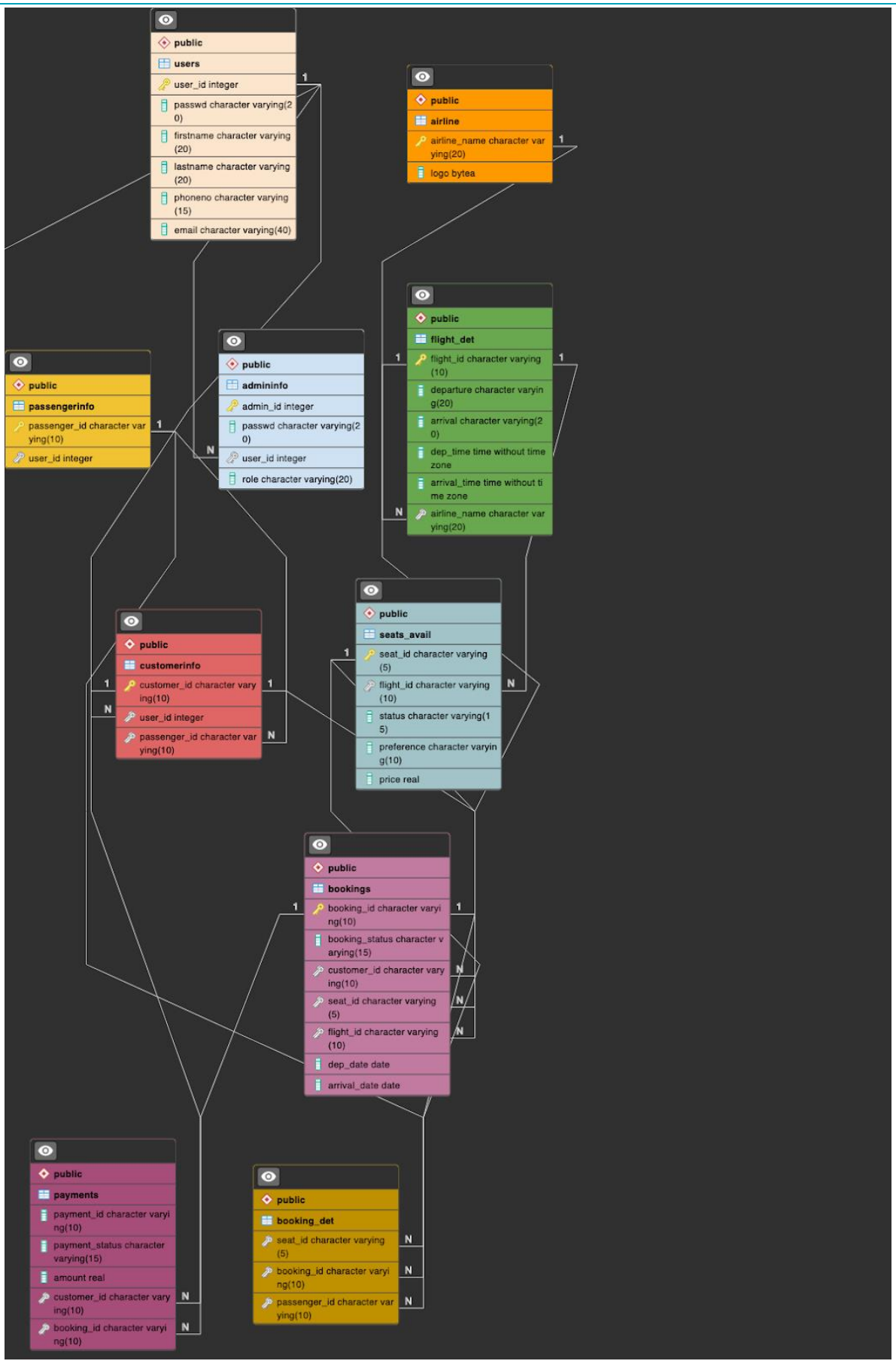
- The system assumes that users have access to a reliable internet connection and a compatible device to access the system.

- The system depends on reliable third-party services for payment processing and other functionalities.

7. Conclusion

The airline ticketing system will provide a comprehensive platform for users to book and manage their air travel. The system will be secure, scalable, and user-friendly, making it an ideal solution for customers worldwide.

ENTITY-RELATIONSHIP DIAGRAM



NORMALISING RELATIONS TILL 3NF

1) USERS

Query | Query History

```
1  CREATE TABLE USERS(
2  user_id integer primary key,
3  passwd varchar(20),
4  firstname varchar(20),
5  lastname varchar(20),
6  phoneno varchar(15),
7  email varchar(40));
```

Query | Query History

```
1  SELECT * FROM public.users
2  ORDER BY user_id ASC
```

	user_id [PK] integer	passwd character varying (20)	firstname character varying (20)	lastname character varying (20)	phoneno character varying (15)	email character varying (40)
1	100001	123	Nandini	Maharaj	9963732286	nandini.maharaj@gmail.com
2	100002	123	Aaditya	Maharaj	9334455666	aaditya.maharaj@gmail.com
3	100003	123	Vinod	Kumar	9778899006	vinod11@gmail.com
4	100004	123	Sharada	Gundimeda	8877665544	sharrockz@gmail.com
5	100005	123	Adilakshmi	Ghali	7788996655	laxmii234@gmail.com
6	100006	123	Mahathi	Suryadevara	4466337744	mahatthi@gmail.com
7	100007	123	Basmah	Fatima	3344556677	bfatima@hotmail.com
8	100008	123	Reya	Stalin	2244335566	josephh@hotmail.com
9	100009	123	Afreen	Shaik	2267896789	shakeeaff@hotmail.com
10	100010	123	Rohith	Mynampati	9879876565	mynamrohith@hotmail.com
11	100011	123	Rithish	Mynampati	9678678990	mynamrishi@yahoo.com
12	100012	123	Rohith	Kodakandla	8769876334	kondaa@yahoo.com
13	100013	123	Bharghav	Kodakandla	8769876334	bhargkod223@yahoo.com
14	100014	123	Rishikesh	Sauderpally	5566449900	rishi@hotmail.com
15	100015	123	Aditya	Anand	8798346654	adibroo@gmail.com
16	100016	123	Aditya	Mandal	7733442255	adimandal34@gmail.com
17	100017	123	Aditya	Jha	7836511772	adi88jha@gmail.com
18	100018	123	Praneeth		9900887799	prannuy@gmail.com
19	100019	123	Harsha	Mallemkondu	9876004429	harshhrockz@gmail.com
20	100020	123	Raju	K	9123345567	rajuu@gmail.com
21	100021	123	Ram		8900987633	ramsiyaa@gmail.com

This table is in 1NF, 2NF & 3NF.

2) PASSENGERINFO

Query | Query History

```
1  CREATE TABLE PASSENGERINFO(
2  passenger_id varchar(10) primary key,
3  user_id integer,
4  FOREIGN KEY(user_id)
5        REFERENCES users(user_id)
6  );
```

Query | Query History

```
1  SELECT * FROM public.passengerinfo
2  ORDER BY passenger_id ASC
```

	passenger_id [PK] character varying (10)	user_id integer
1	1E202301	100001
2	1E202302	100002
3	1E202303	100003
4	1E202304	100004
5	1E202305	100005
6	1E202306	100006
7	1E202307	100007
8	1E202308	100008
9	1E202309	100009
10	1E202310	100010
11	1E202311	100011
12	1E202312	100012
13	1E202313	100013
14	1E202314	100014
15	1E202315	100015
16	1E202316	100016
17	1E202317	100017

This table is in 1NF, 2NF & 3NF.

3) CUSTOMERINFO

Query | Query History

```
1  CREATE TABLE CUSTOMERINFO(
2  customer_id varchar(10) primary key,
3  user_id integer,
4  passenger_id varchar(10),
5  FOREIGN KEY(user_id)
6        REFERENCES users(user_id),
7  FOREIGN KEY(passenger_id)
8        REFERENCES passengerinfo(passenger_id)
9
10 );
```

Query | Query History

```
1  SELECT * FROM public.customerinfo
2  ORDER BY customer_id ASC
```

Data Output | Messages | Notifications

	customer_id [PK] character varying (10)	user_id integer	passenger_id character varying (10)
1	IND01	100001	1E202301
2	IND02	100005	1E202305
3	IND03	100006	1E202306
4	IND04	100009	1E202309
5	IND05	100011	1E202311
6	IND06	100013	1E202313
7	IND07	100014	1E202314
8	IND08	100015	1E202315

This table is in 1NF, 2NF but not 3NF. This is because of transitive dependency:

{user_id}->{customer_id}

{passenger_id}->{user_id}

So decompose the table as: CUSTOMERINFO

(Since relation between passenger_id and user_id already exists)

Query Query History

```
1  CREATE TABLE public.customerinfo
2  (
3      customer_id character varying(10) NOT NULL,
4      user_id integer NOT NULL,
5      PRIMARY KEY (customer_id),
6      CONSTRAINT user_id FOREIGN KEY (user_id)
7          REFERENCES public.users (user_id)
8  );
```

```
1  SELECT * FROM public.customerinfo
2  ORDER BY customer_id ASC
```

Data Output Messages Notifications ...

	customer_id [PK] character varying (10)	user_id integer
1	IND01	100001
2	IND02	100005
3	IND03	100006
4	IND04	100009
5	IND05	100011
6	IND06	100013
7	IND07	100014
8	IND08	100015

This table is in 1NF, 2NF & 3NF.

4) ADMININFO

Query | Query History

```
1  CREATE TABLE public.admininfo
2  (
3      admin_id integer NOT NULL,
4      passwd character varying(20),
5      role character varying(20),
6      user_id integer NOT NULL,
7      PRIMARY KEY (admin_id),
8      CONSTRAINT user_id FOREIGN KEY (user_id)
9          REFERENCES public.users (user_id)
10 );
```

Query | Query History

```
1  SELECT * FROM public.admininfo
2  ORDER BY admin_id ASC
```

Data Output | Messages | Notifications

	admin_id [PK] integer	passwd character varying (20)	role character varying (20)	user_id integer
1	282301	123	HEAD ADMIN	100018
2	282302	123	HEAD ADMIN	100019
3	282303	123	SUB ADMIN	100020
4	282304	123	SUB ADMIN	100021

This table is in 1NF, 2NF & 3NF.

5) FLIGHTINFO

```
Query  Query History
1  CREATE TABLE FLIGHT_DET(
2  flight_id varchar(10) PRIMARY KEY,
3  departure varchar(20),
4  arrival varchar(20),
5  dep_time TIME,
6  arrival_time TIME,
7  airline_name varchar(20),
8      FOREIGN KEY(airline_name)
9      REFERENCES airline(airline_name)
10 )
```

```
Query  Query History
1  SELECT * FROM public.flight_det
2  ORDER BY flight_id ASC
```

Data Output | Messages | Notifications

	flight_id [PK] character varying (10)	departure character varying (20)	arrival character varying (20)	dep_time time without time zone	arrival_time time without time zone	airline_name character varying (20)
1	ASTR21	HYD	GOI	23:45:00	17:30:00	CONTINENTAL
2	BRCH20	MAA	SAG	06:00:00	09:20:00	DELTA
3	BRIM20	BOM	IXC	02:20:00	05:00:00	INDIGO
4	CBR21	BLR	CJB	10:32:00	15:50:00	AIR INDIA
5	CYP20	DEL	CCU	16:30:00	18:22:00	CONTINENTAL
6	FDE22	CCU	DEL	19:10:00	21:20:00	EMIRATES
7	HBR22	CJB	BLR	23:30:00	16:00:00	CONTINENTAL
8	JETT20	IXC	MAA	02:50:00	05:15:00	AIR INDIA
9	KJ20	SAG	BOM	14:30:00	18:00:00	CONTINENTAL
10	KYO21	GOI	HYD	02:10:00	17:10:00	INDIGO
11	NEO22	HYD	MAA	17:00:00	23:25:00	DELTA
12	OMEN20	MAA	BLR	05:45:00	08:20:00	AIR INDIA
13	PHNX20	BOM	GOI	00:05:00	03:30:00	INDIGO
14	RAZ20	BLR	SAG	13:05:00	14:20:00	EMIRATES
15	REY20	DEL	CJB	10:15:00	12:20:00	EMIRATES
16	SAG20	CCU	DEL	19:25:00	12:50:00	INDIGO
17	SKY20	CJB	IXC	16:45:00	01:00:00	DELTA
18	SOVA20	IXC	MAA	21:00:00	23:00:00	AIR INDIA
19	VPR20	SAG	CCU	08:00:00	10:00:00	DELTA
20	YOR21	GOI	BOM	22:10:00	02:50:00	EMIRATES

This table is in 1NF, 2NF but not 3NF. This is because of transitive dependency:

{airline_name}->{flight_id}

{flight_id}->{departure, arrival,dep_time, arrival_time}

So decompose the tables as: FLIGHTINFO AND AIRLINEINFO

Query | Query History

```
1  CREATE TABLE public.flightinfo
2  (
3      flight_id character varying(10) NOT NULL,
4      dep_location character varying(20),
5      arrival_location character varying(20),
6      dep_time time without time zone,
7      arr_time time without time zone,
8      PRIMARY KEY (flight_id)
9  );
```

Query | Query History

```
1  SELECT * FROM public.flightinfo
2  ORDER BY flight_id ASC
```

Data Output | Messages | Notifications

	flight_id [PK] character varying (10)	dep_location character varying (20)	arrival_location character varying (20)	dep_time time without time zone	arr_time time without time zone
1	ASTR21	HYD	GOI	23:45:00	17:30:00
2	BRCH20	MAA	SAAG	06:00:00	09:20:00
3	BRIM20	BOM	IXC	02:20:00	05:00:00
4	CBR21	BLR	CJB	10:32:00	15:50:00
5	CYP20	DEL	CCU	16:30:00	18:22:00
6	FDE22	CCU	DEL	19:10:00	21:20:00
7	HBR22	CJB	BLR	23:30:00	16:00:00
8	JETT20	IXC	MAA	02:50:00	05:15:00
9	KJ20	SAG	BOM	14:30:00	18:00:00
10	KYO21	GOI	HYD	02:10:00	17:10:00
11	NEO22	HYD	MAA	17:00:00	23:25:00
12	OMEN20	MAA	BLR	05:45:00	08:20:00
13	PHNX20	BOM	GOI	00:05:00	03:30:00
14	RAZ20	BLR	SAG	13:05:00	14:20:00
15	REY20	DEL	CJB	10:15:00	12:20:00
16	SAG20	CCU	DEL	19:25:00	12:50:00
17	SKY20	CJB	IXC	16:45:00	01:00:00
18	SOVA20	IXC	MAA	21:00:00	23:00:00
19	VPR20	SAG	CCU	08:00:00	10:00:00
20	YOR21	GOI	BOM	22:10:00	02:15:00

This table is in 1NF, 2NF & 3NF.

6) AIRLINEINFO

Query | Query History

```
1  CREATE TABLE public.airlineinfo
2  (
3      airline_name character varying(20) NOT NULL,
4      flight_id character varying(10) NOT NULL,
5      PRIMARY KEY (airline_name, flight_id),
6      CONSTRAINT flight_id FOREIGN KEY (flight_id)
7          REFERENCES public.flightinfo (flight_id)
8  );
```

Query | Query History

```
1  SELECT * FROM public.airlineinfo
2  ORDER BY airline_name ASC, flight_id ASC
```

Data Output | Messages | Notifications

	airline_name [PK] character varying (20)	flight_id [PK] character varying (10)
1	AIR INDIA	OMEN20
2	AIR INDIA	SOVA20
3	CONTINENTAL	ASTR21
4	DELTA	SKY20
5	DELTA	VPR20
6	EMIRATES	RAZ20
7	EMIRATES	REY20
8	EMIRATES	YOR21
9	INDIGO	PHNX20
10	INDIGO	SAG20

This table is in 1NF, 2NF & 3NF.

7) SEATINFO

```
Query  Query History

1  CREATE TABLE SEATS_AVAIL(
2  seat_id varchar(5) PRIMARY KEY,
3  flight_id varchar(10),
4  status varchar(15),
5  preference varchar(10),
6  price real,
7      FOREIGN KEY(flight_id)
8      REFERENCES flight_det(flight_id)
```

```
1  SELECT * FROM public.seats_avail
2  ORDER BY seat_id ASC
```

Data Output | Messages | Notifications

	seat_id [PK] character varying (5)	flight_id character varying (10)	status character varying (15)	preference character varying (10)	price real
70	153D	ASTR21	AVAILABLE	AISLE	19000
71	161A	KYO21	AVAILABLE	WINDOW	41000
72	161B	KYO21	AVAILABLE	WINDOW	41000
73	162A	KYO21	AVAILABLE	WINDOW	27000
74	162B	KYO21	BOOKED	WINDOW	27000
75	162C	KYO21	AVAILABLE	AISLE	27000
76	162D	KYO21	AVAILABLE	AISLE	27000
77	163A	KYO21	AVAILABLE	AISLE	19000
78	163B	KYO21	AVAILABLE	AISLE	19000
79	163C	KYO21	AVAILABLE	AISLE	19000
80	163D	KYO21	AVAILABLE	AISLE	19000
81	171A	CBR21	AVAILABLE	WINDOW	41000
82	171B	CBR21	BOOKED	WINDOW	41000
83	172A	CBR21	AVAILABLE	WINDOW	27000
84	172B	CBR21	AVAILABLE	WINDOW	27000
85	172C	CBR21	AVAILABLE	AISLE	27000
86	172D	CBR21	AVAILABLE	AISLE	27000
87	173A	CBR21	AVAILABLE	AISLE	19000
88	173B	CBR21	AVAILABLE	AISLE	19000
89	173C	CBR21	AVAILABLE	AISLE	19000
90	173D	CBR21	BOOKED	AISLE	19000
91	181A	NEO22	AVAILABLE	WINDOW	41000
92	181B	NEO22	AVAILABLE	WINDOW	41000

This table is in 1NF, 2NF but not 3NF. This is because of transitive dependency:

{flight_id}->{seat_id}

{seat_id}->{status,preference,price}

So decompose the tables as: SEATFLIGHT AND SEATINFO

Query Query History

```
1  CREATE TABLE public.seatinfo
2  (
3      seat_id character varying(5) NOT NULL,
4      preference character varying(10),
5      price real,
6      PRIMARY KEY (seat_id)
7  );
```

Query Query History

```
1  SELECT * FROM public.seatinfo
2  ORDER BY seat_id ASC
```

Data Output Messages Notifications

	seat_id [PK] character varying (5)	preference character varying (10)	price real
1	1A	WINDOW	41000
2	1B	WINDOW	41000
3	2A	WINDOW	27000
4	2B	WINDOW	27000
5	2C	AISLE	27000
6	2D	AISLE	27000
7	3A	AISLE	19000
8	3B	AISLE	19000
9	3C	AISLE	19000
10	3D	AISLE	19000

This table is in 1NF, 2NF & 3NF.

8) SEATFLIGHT

Query | Query History

```
1  CREATE TABLE public.seatflight
2  (
3      seat_id character varying(5) NOT NULL,
4      flight_id character varying(10) NOT NULL,
5      PRIMARY KEY (seat_id, flight_id),
6      CONSTRAINT seat_id FOREIGN KEY (seat_id)
7          REFERENCES public.seatinfo (seat_id),
8      CONSTRAINT flight_id FOREIGN KEY (flight_id)
9          REFERENCES public.flightinfo (flight_id)
10 );
```

Query | Query History

```
1  SELECT * FROM public.seatflight
2  ORDER BY flight_id ASC
```

Data Output | Messages | Notifications

	seat_id [PK] character varying (5)	flight_id [PK] character varying (10)
1	1A	ASTR21
2	1B	ASTR21
3	2A	ASTR21
4	2B	ASTR21
5	2C	ASTR21
6	2D	ASTR21
7	3A	ASTR21
8	3B	ASTR21
9	3C	ASTR21
10	3D	ASTR21
11	1A	BRCH20
12	3D	BRCH20
13	3C	BRCH20
14	3B	BRCH20
15	3A	BRCH20
16	2D	BRCH20
17	2C	BRCH20

This table is in 1NF, 2NF & 3NF.

9) BOOKINGSINFO

```
Query  Query History
1  CREATE TABLE BOOKINGS(
2  booking_id varchar(10) PRIMARY KEY,
3  booking_status varchar(15),
4  customer_id varchar(10),
5  seat_id varchar(5),
6  flight_id varchar(10),
7  dep_date DATE,
8  arrival_date DATE,
9      FOREIGN KEY(customer_id)
10     REFERENCES customerinfo(customer_id),
11
12     FOREIGN KEY(seat_id)
13     REFERENCES seats_avail(seat_id),
14
15     FOREIGN KEY(flight_id)
16     REFERENCES flight_det(flight_id)
17
18 )
```

```
Query  Query History
1  SELECT * FROM public.bookings
2  ORDER BY booking_id ASC
```

	booking_id [PK] character varying (10)	booking_status character varying (15)	customer_id character varying (10)	seat_id character varying (5)	flight_id character varying (10)	dep_date date	arrival_date date
1	N6501	BOOKED	IND01	11A	BRIM20	2023-06-01	2023-06-01
2	N6502	BOOKED	IND02	13D	BRIM20	2023-06-01	2023-06-01
3	N6503	BOOKED	IND03	23B	OMEN20	2023-06-01	2023-06-01
4	N6504	BOOKED	IND04	32D	VPR20	2023-06-01	2023-06-01
5	N6505	BOOKED	IND05	42B	RAZ20	2023-06-01	2023-06-01
6	N6506	BOOKED	IND06	51B	CYP20	2023-06-01	2023-06-02
7	N6507	BOOKED	IND07	53D	CYP20	2023-06-01	2023-06-02
8	N6508	BOOKED	IND08	63B	SAG20	2023-06-01	2023-06-02
9	N6509	BOOKED	IND01	72D	SOVA20	2023-06-01	2023-06-02
10	N6510	BOOKED	IND02	82B	PHNX20	2023-06-02	2023-06-02
11	N6511	BOOKED	IND03	91B	JETT20	2023-06-02	2023-06-02
12	N6512	BOOKED	IND04	93D	JETT20	2023-06-02	2023-06-02
13	N6513	BOOKED	IND05	103B	BRCH20	2023-06-02	2023-06-02
14	N6514	BOOKED	IND06	112D	REY20	2023-06-02	2023-06-02
15	N6515	BOOKED	IND07	122B	KJ20	2023-06-02	2023-06-02
16	N6516	BOOKED	IND08	131B	SKY20	2023-06-02	2023-06-03
17	N6517	BOOKED	IND01	133D	SKY20	2023-06-02	2023-06-03
18	N6518	BOOKED	IND02	143B	YOR21	2023-06-02	2023-06-03
19	N6519	BOOKED	IND03	152D	ASTR21	2023-06-02	2023-06-03
20	N6520	BOOKED	IND04	162B	KYO21	2023-06-03	2023-06-03
21	N6521	BOOKED	IND05	171B	CBR21	2023-06-03	2023-06-03
22	N6522	BOOKED	IND06	173D	CBR21	2023-06-03	2023-06-03
23	N6523	BOOKED	IND07	183D	NEO22	2023-06-03	2023-06-03
24	N6524	BOOKED	IND08	193B	FDE22	2023-06-03	2023-06-04
25	N6525	BOOKED	IND01	202D	HBR22	2023-06-03	2023-06-04

This table is in 1NF, 2NF but not 3NF. This is because of transitive dependency:

{booking_id}->{customer_id,dep_date,arrival_date,booking_status}

{seat_id,flight_id,passenger_id}->{booking_id}

So decompose the tables as: BOOKINGINFO AND SEATSBOOKED

Query Query History

```
1  CREATE TABLE public.bookinginfo
2  (
3      booking_id character varying(10) NOT NULL,
4      customer_id character varying(10) NOT NULL,
5      dep_date date,
6      arrival_time date,
7      PRIMARY KEY (booking_id),
8      CONSTRAINT customer_id FOREIGN KEY (customer_id)
9          REFERENCES public.customerinfo (customer_id)
10 );
```

Query Query History

```
1  SELECT * FROM public.bookinginfo
2  ORDER BY booking_id ASC
```

Data Output Messages Notifications

	booking_id [PK] character varying (10)	customer_id character varying (10)	dep_date date	arrival_time date
1	N6501	IND01	2023-06-01	2023-06-01
2	N6502	IND02	2023-06-01	2023-06-01
3	N6503	IND03	2023-06-02	2023-06-02
4	N6504	IND04	2023-06-02	2023-06-02
5	N6505	IND05	2023-06-02	2023-06-02
6	N6506	IND06	2023-06-03	2023-06-03
7	N6507	IND07	2023-06-03	2023-06-03
8	N6508	IND08	2023-06-03	2023-06-03

This table is in 1NF, 2NF & 3NF.

10) SEATSBOOKED

```
Query  Query History
1  CREATE TABLE public.seatsbooked
2  (
3      booking_id character varying(10) NOT NULL,
4      seat_id character varying(5) NOT NULL,
5      flight_id character varying(10) NOT NULL,
6      passenger_id character varying(10),
7      PRIMARY KEY (booking_id, seat_id, flight_id),
8      CONSTRAINT booking_id FOREIGN KEY (booking_id)
9          REFERENCES public.bookinginfo (booking_id),
10     CONSTRAINT seat_id FOREIGN KEY (seat_id)
11         REFERENCES public.seatinfo (seat_id),
12     CONSTRAINT flight_id FOREIGN KEY (flight_id)
13         REFERENCES public.flightinfo (flight_id)
14 );
```

```
Query  Query History
1  SELECT * FROM public.seatsbooked
2  ORDER BY booking_id ASC, seat_id ASC, flight_id ASC
```

Data Output Messages Notifications

	booking_id [PK] character varying (10)	seat_id [PK] character varying (5)	flight_id [PK] character varying (10)	passenger_id character varying (10)
1	N6501	1A	BRIM20	1E202301
2	N6501	2D	VPR20	1E202309
3	N6501	3B	OMEN20	1E202306
4	N6501	3D	BRIM20	1E202305
5	N6502	1B	CYP20	1E202313
6	N6502	2B	RAZ20	1E202311
7	N6502	3D	CYP20	1E202314
8	N6503	2B	PHNX20	1E202305
9	N6503	2D	SOVA20	1E202301
10	N6503	3B	SAG20	1E202315
11	N6504	1B	JETT20	1E202306
12	N6504	3B	BRCH20	1E202311
13	N6504	3D	JETT20	1E202309
14	N6505	1B	SKY20	1E202315
15	N6505	2B	KJ20	1E202314
16	N6505	2D	REY20	1E202313
17	N6506	2D	ASTR21	1E202306
18	N6506	3B	YOR21	1E202305
19	N6506	3D	SKY20	1E202301
20	N6507	1B	CBR21	1E202311
21	N6507	2B	KYO21	1E202309
22	N6507	3D	CBR21	1E202313
23	N6508	2D	HBR22	1E202301
24	N6508	3B	FDE22	1E202315
25	N6508	3D	NEO22	1E202314

This table is in 1NF, 2NF & 3NF.

11) PAYMENTS

```
Query  Query History
1  CREATE TABLE PAYMENTS(
2  payment_id varchar(10),
3  payment_status varchar(15),
4  amount real,
5  customer_id varchar(10),
6  booking_id varchar(10),
7      FOREIGN KEY(booking_id)
8      REFERENCES bookings(booking_id),
9
10     FOREIGN KEY(customer_id)
11     REFERENCES customerinfo(customer_id)
```

```
Query  Query History
1  SELECT * FROM public.payments
```

Data Output | Messages | Notifications

	payment_id character varying (10)	payment_status character varying (15)	amount real	customer_id character varying (10)	booking_id character varying (10)
1	A64G01	PAID	41000	IND01	N6501
2	A64G02	PAID	27000	IND01	N6509
3	A64G03	PAID	19000	IND01	N6517
4	A64G04	PAID	27000	IND01	N6525
5	A64G05	PAID	19000	IND02	N6502
6	A64G06	PAID	27000	IND02	N6510
7	A64G07	PAID	19000	IND02	N6518
8	A64G08	PAID	19000	IND03	N6503
9	A64G09	PAID	41000	IND03	N6511
10	A64G10	PAID	27000	IND03	N6519
11	A64G11	PAID	27000	IND04	N6504
12	A64G12	PAID	19000	IND04	N6512
13	A64G13	PAID	27000	IND04	N6520
14	A64G14	PAID	27000	IND05	N6505
15	A64G15	PAID	19000	IND05	N6513
16	A64G16	PAID	41000	IND05	N6521
17	A64G17	PAID	41000	IND06	N6506
18	A64G18	PAID	27000	IND06	N6514
19	A64G19	PAID	19000	IND06	N6522
20	A64G20	PAID	19000	IND07	N6507
21	A64G21	PAID	27000	IND07	N6515

This table is in 1NF, 2NF but not 3NF. This is because of transitive dependency:

{booking_id}->{customer_id}

{payment_id}->{payment_status,amount,booking_id}

So decompose the table as: PAYMENTS

(Since relation between booking_id and customer_id already exists)

Query Query History

```
1  CREATE TABLE public.payments
2  (
3      payment_id character varying(10) NOT NULL,
4      booking_id character varying(10) NOT NULL,
5      amount real,
6      PRIMARY KEY (payment_id),
7      CONSTRAINT booking_id FOREIGN KEY (booking_id)
8          REFERENCES public.bookinginfo (booking_id)
9  );
```

Query Query History

```
1  SELECT * FROM public.payments
2  ORDER BY payment_id ASC
```

Data Output Messages Notifications

	payment_id [PK] character varying (10)	booking_id character varying (10)	amount real
1	A64G01	N6501	114000
2	AG4G02	N6502	65000
3	AG4G03	N6503	73000
4	AG4G04	N6504	73000
5	AG4G05	N6505	87000
6	AG4G06	N6506	87000
7	AG4G07	N6507	65000
8	AG4G08	N6508	79000