AIRLINE BOOKING SYSTEM

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology/Master of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Y. Navaneetha

(AP20110010411)

Under the Guidance of

Prof. Dr Rajiv Senapati

Department of Computer science and Engineering

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[December, 2022]

Certificate

Date: 08-Dec-22

This is to certify that the work present in this Project entitled "**AIRLINE BOOKING SYSTEM**" has been carried out by **Y.Navaneetha** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Dr. Rajiv Senapati,

Assistant Professor,

Department of Computer science and Engineering.

Acknowledgements

We thank **Dr.Rajiv Senapati sir**, our professor in charge, for the help and direction in finishing our project on the subject Database Management System. It was a fantastic learning opportunity.This subject has given me an opportunity to explore the field I have always been curious about. Your insightful counsel and recommendations were quite beneficial to me as I finished the assignment. I will always be grateful to you for this.

Navaneetha Yandrapragada

Table of Contents

Abstract

Data needs to be stored in an organized way so that it is easy to use and manipulate the information. This project talks about an airline booking system where it stores the user details, information about flights, tickets booked and payment information. Checking requirements, finding all entities and their properties, ER diagram, converting ER to relational model, Normalizing, and SQL code are the steps followed. The main objective is to reduce the redundancy and inconsistency of the data.

Introduction

There are many different ways to store and retrieve the data. Older versions like using files for data organizing have many issues and take more manpower and time.Data Redundancy, Data Inconsistency, Data Isolation, Atomicity, Concurrent access, Security are the few tasks which becomes an anomaly by using files. Database Management System (DBMS) helps to hold all above issues.

Airline Booking System is one of the applications where maintaining a database is required. In this report, the simple requirements are considered. This database system stores the data about the users, availability of flights and the information about the reserved tickets. Users can check for the flight details and can reserve a ticket.

2. Methodology

2.1 Entity List

1. USER
2. FLIGHT
3. AIRLINE
4. TICKET
5. PAYMENT
6. CLASS

2.1.1 Entities With Attributes

USER

ATTRIBUTE	DATATYPE	CONSTRAINT
USERID	VARCHAR (30)	PRIMARY KEY
NAME	VARCHAR (30)	NOT NULL
DOB	DATE	NOT NULL
AGE	INT	-
ADDRESS	VARCHAR (30)	NOT NULL
GENDER	VARCHAR (6)	NOT NULL
MOBILE_NO	VARCHAR (10)	NOT NULL

FLIGHT

ATTRIBUTE	DATATYPE	CONSTRAINT
FLIGHT_NO	VARCHAR (30)	PRIMARY KEY
FLIGHT_NAME	VARCHAR (30)	NOT NULL
DEP_TIME	TIME	NOT NULL
ARR_TIME	TIME	NOT NULL
SOURCE	VARCHAR (30)	NOT NULL
DESTINATION	VARCHAR (30)	NOT NULL
SEAT_AVAILABLE	INT	NOT NULL

AIRLINE

ATTRIBUTE	DATATYPE	CONSTRAINT
AIRLINE_NAME	VARCHAR (30)	PRIMARY KEY
CONTACT_NO	VARCHAR (30)	NOT NULL

TICKET

ATTRIBUTE	DATATYPE	CONSTRAINT
ETICKET	VARCHAR (30)	PRIMARY KEY
FLIGHT_NO	VARCHAR (30)	NOT NULL
DATE_TIME	DATETIME	NOTNULL
GATE	INT	NOT NULL
CLASS	VARCHAR (30)	NOT NULL
SOURCE	VARCHAR (30)	NOT NULL
DESTINATION	VARCHAR (30)	NOT NULL
SEAT_NO	INT	NOT NULL

PAYMENT

ATTRIBUTE	DATATYPE	CONSTRAINT
TRANSACTION_ID	VARCHAR (30)	PRIMARY KEY
ACCOUNT_ID	LONG	NOT NULL
AMOUNT	LONG	NOT NUL

CLASS

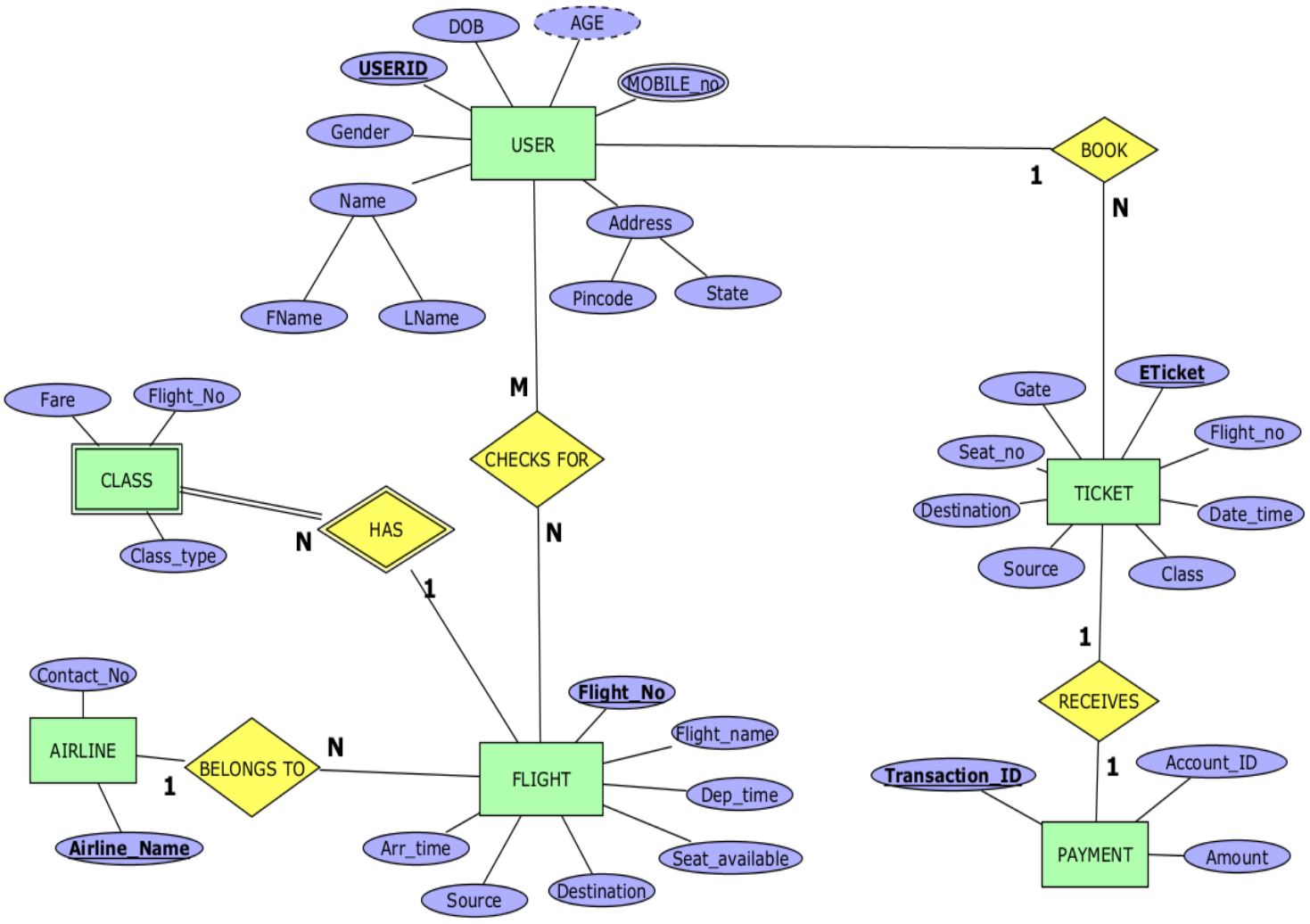
ATTRIBUTE	DATATYPE	CONSTRAINT
FLIGHT_NO	VARCHAR (30)	NOT NULL
CLASS_TYPE	VARCHAR (30)	NOT NULL
FARE	LONG	NOT NULL

2.2 Cardinality & ER Diagram

Entity-1	Entity-2	Relation	Cardinality
USER	TICKET	BOOK	1: N
USER	FLIGHT	CHECKS FOR	M: N
FLIGHT	CLASS	HAS	1: N
FLIGHT	AIRLINE	BELONG TO	N: 1
TICKET	PAYMENT	RECEIVES	1: 1

ER MODEL: It is a pictorial representation of entities and relationships among them.

ER DIAGRAM



2.3 ER Model to Relational Model

RELATIONAL MODEL: Data is represented as relation (or table) which consists of rows and columns.

Below are the rules used for conversion of ER model to Relational model

1. An entity in the ER model is represented by a relational table in the relational model.

2. All the attributes of the ER model are represented in different columns as an attribute.

3. Primary key attribute of the ER model is represented as the primary key in the relational model.

4. Composite key of the ER model is split in different columns in the relational model.

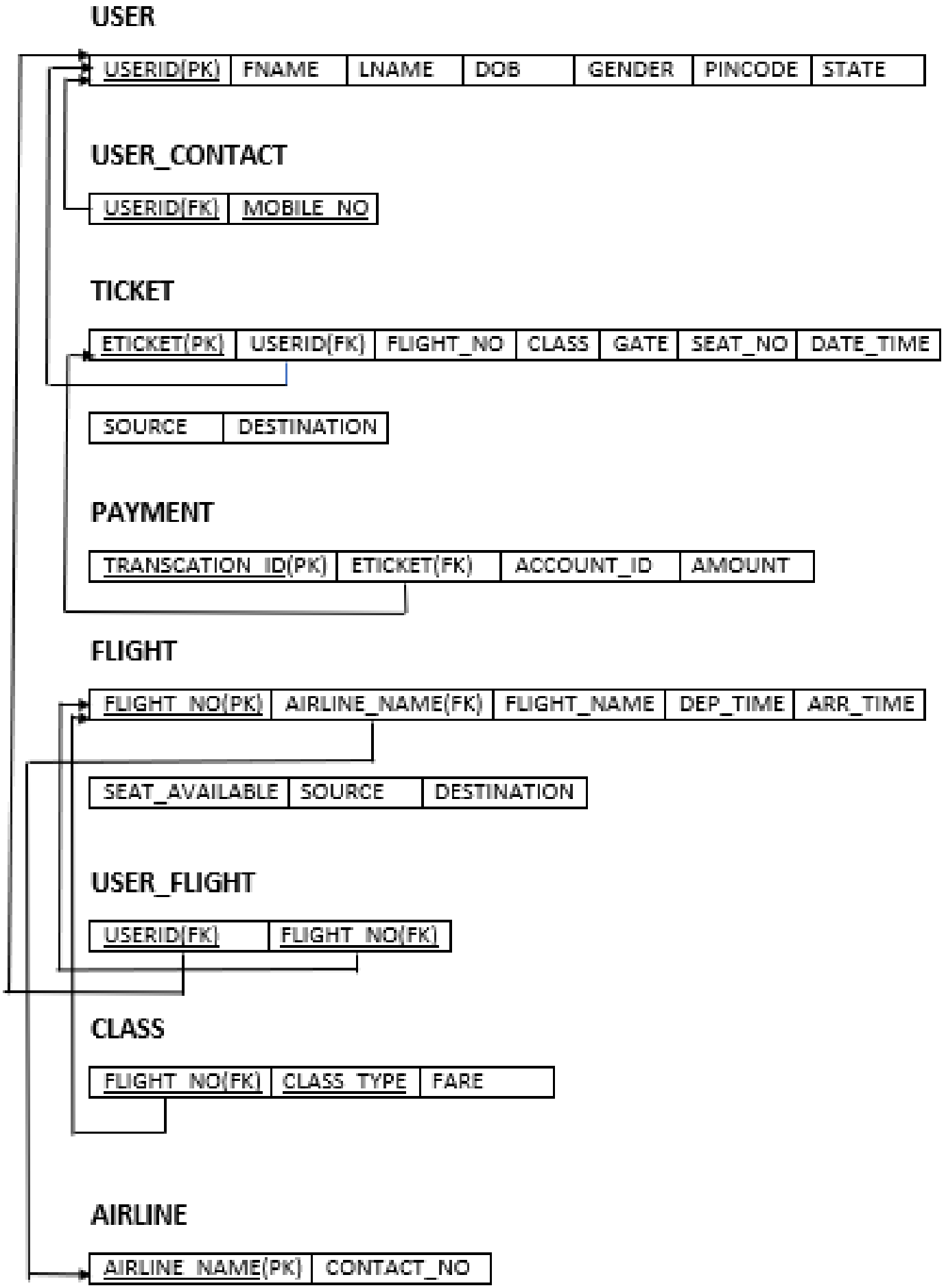
5. Derived attributes must be dropped in relational models.

6. Multi valued attributes need to be kept separate in a new table.

7. If the relationship between two entities is 1:1 then the primary key of one relation becomes foreign key in another relation.

8. If the relationship between entities is 1:M or M:1 then the primary key attribute of one-sided relations becomes a foreign key in many-sided relations.

9. If the relationship between entities is M:M then a new table needs to be created to represent that relation, where the new relation will have the primary key of both relations as the foreign keys.


USER
USERID(PK)
FNAME
LNAME
DOB
GENDER
PINCODE
STATE
USER_CONTACT
USERID(FK)
MOBILE_NO
TICKET
ETICKET(PK)
USERID(FK)
FLIGHT_NO
CLASS
GATE
SEAT_NO
DATE_TIME
SOURCE
DESTINATION
PAYMENT
TRANSCATION_ID(PK)
ETICKET(FK)
ACCOUNT_ID
AMOUNT
FLIGHT
FLIGHT_NO(PK)
AIRLINE_NAME(FK)
FLIGHT_NAME
DEP_TIME
ARR_TIME
SEAT_AVAILABLE
SOURCE
DESTINATION
USER_FLIGHT
USERID(FK)
FLIGHT_NO(FK)
CLASS
FLIGHT_NO(FK)
CLASS_TYPE
FARE
AIRLINE
AIRLINE_NAME(PK)
CONTACT_NO

2.4 Normalization

Normalization is a process of analyzing and decomposing the complex relation which satisfies some constraint to form a simple relation.

There are 5 Normal forms:

1. First Normal Form (1NF)

2. Second Normal Form (2NF)

3. Third Normal Form (3NF) or Boyes Cord Normal Form (BCNF)

4. Fourth Normal Form (4NF)

5. Fifth Normal Form (5F) or Project Join Normal Form (PJNF)

FIRST NORMAL FORM (1NF)

- A relation is said to be in 1NF, if it has got no non-atomic attributes i.e., which cannot be subdivided.

➢ Our relational model is already in the 1NF because it has only non-atomic attributes in every relation.

SECOND NORMAL FORM (2NF)

- A relation in 1NF is said to be in 2NF, if it satisfies any one of the following conditions. They are,

1. The primary key consists of only one attribute.
2. There exists no non key attribute.
3. Every non key attribute present in relation should functionally depend upon a full set of primary keys.

➢ All the relations in our model have only one attribute in the primary key except USER_CONTACT, USER_FLIGHT and CLASS.
➢ USER_CONTACT and USER_FLIGHT have no non key attributes.
➢ CLASS non key attribute functionally depends on its set of primary keys.
➢ Each relation satisfies at least one of the conditions. So, the model is in 2NF.

THIRD NORMAL FORM (3NF)

- A relation which is in 2NF is said to be in 3NF, if there exists no transitive functional dependency of any non-key attribute on the set of primary keys.
- Transitivity says that, if X->Y and Y->Z then X->Z.

➢ In USER relation, USERID -> PINCODE and PINCODE -> STATE

So, the relation should be decomposed.

USER

USERID(PK)	FNAME	LNAME	DOB	GENDER	PINCODE

USER_PINCODE

PINCODE(PK)	STATE

➢ In TICKET relation, ETICKET -> FLIGHT_NO and FLIGHT_NO -> SOURCE, DESTINATION, DATE_TIME.

TICKET

ETICKET(PK)	USERID(FK)	FLIGHT_NO	CLASS	GATE	SEAT_NO

TICKET_FLIGHT_NO

FLIGHT_NO (PK)	SOURCE	DESTINATION	DATE_TIME

RELATIONS

USER

USERID(PK)	FNAME	LNAME	DOB	GENDER	PINCODE

USER_PINCODE

PINCODE(PK)	STATE

USER_CONTACT

USERID(FK)	MOBILE_NO

TICKET

ETICKET(PK)	USERID(FK)	FLIGHT_NO	CLASS	GATE	SEAT_NO

TICKET_FLIGHT_NO

FLIGHT_NO (PK)	SOURCE	DESTINATION	DATE_TIME

PAYMENT

TRANSCATION_ID(PK)	ETICKET(FK)	ACCOUNT_ID	AMOUNT

FLIGHT

FLIGHT_NO(PK)	AIRLINE_NAME(FK)	FLIGHT_NAME	DEP_TIME	ARR_TIME

SEAT_AVAILABLE	SOURCE	DESTINATION

USER_FLIGHT

USERID(FK)	FLIGHT_NO(FK)

CLASS

FLIGHT_NO(FK)	CLASS_TYPE	FARE

AIRLINE

AIRLINE_NAME(PK)	CONTACT_NO

2.5 SQL Code

/* Creating Database*/

```
create database airline_booking_system;
```

/* Using created database*/

```
use airline_booking_system;
```

/* Creating Table for user */

```
create table user(
 userid varchar(30) primary key,
 fname varchar(30) not null,
 lname varchar(30) not null,
 dob date not null,
 gender varchar(6) not null,
 pincode varchar(10) not null
 );
```

Field	Type	Null	Key	Default	Extra
userid	varchar(30)	NO	PRI	NULL	
fname	varchar(30)	NO		NULL	
lname	varchar(30)	NO		NULL	
dob	date	NO		NULL	
gender	varchar(6)	NO		NULL	
pincode	varchar(10)	NO		NULL	

/* Inserting values into user*/

```
insert into user values("20110010411","Ram","Potheneni",'2002-05-12',"male",522438);
 insert into user values("20110010318","Sahi","Pendyala",'2002-11-23',"female",524789);
 insert into user
values("20110011741","Seetha","Simhadri",'2003-12-11',"female",522417);
```

userid	fname	lname	dob	gender	pincode
20110010318	Sahi	Pendyala	2002-11-23	female	524789
20110010411	Ram	Potheneni	2002-05-12	male	522438
20110011741	Seetha	Simhadri	2003-12-11	female	522417
NULL	NULL	NULL	NULL	NULL	NULL

/* Creating Table for user_contact */

```
create table user_contact(
 userid varchar(30) not null,
 mobile_no varchar(10) not null,
 primary key(userid,mobile_no),
 foreign key(userid) references user(userid)
 );
```

Field	Type	Null	Key	Default	Extra
userid	varchar(30)	NO	PRI	NULL	
mobile_no	varchar(10)	NO	PRI	NULL	

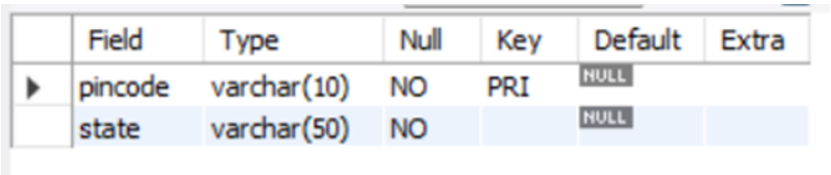
/* Inserting values into user_contact*/

```
insert into user_contact values("20110010411","7418529630");
 insert into user_contact values("20110010411","9632587014");
 insert into user_contact values("20110010318","8521479633");
 insert into user_contact values("20110011741","8462157934");
 insert into user_contact values("20110011741","9876543210");
```

userid	mobile_no
20110010318	8521479633
20110010411	7418529630
20110010411	9632587014
20110011741	8462157934
20110011741	9876543210
NULL	NULL

/* Creating table for user_pincode */

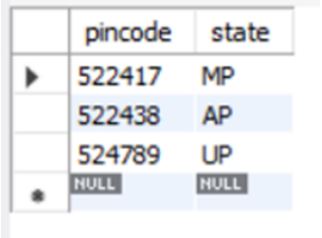
```
create table user_pincode(
 pincode varchar(10) primary key,
 state varchar(50) not null
 );
```



Field	Type	Null	Key	Default	Extra
pincode	varchar(10)	NO	PRI	NULL	
state	varchar(50)	NO		NULL	

/* Inserting values into user_pincode*/

```
insert into user_pincode values("522438","AP");
 insert into user_pincode values("524789","UP");
 insert into user_pincode values("522417","MP");
```



pincode	state
522417	MP
522438	AP
524789	UP
NULL	NULL

/* Creating table for ticket */

```
create table ticket(
eticket varchar(30) primary key,
userid varchar(30) not null,
flight_no varchar(30) not null,
class varchar(30) not null,
gate int not null,
seat_no int not null,
foreign key(userid) references user(userid)
);
```

Field	Type	Null	Key	Default	Extra
eticket	varchar(30)	NO	PRI	NULL	
userid	varchar(30)	NO	MUL	NULL	
flight_no	varchar(30)	NO		NULL	
class	varchar(30)	NO		NULL	
gate	int	NO		NULL	
seat_no	int	NO		NULL	

/* Inserting values into ticket */

insert into ticket values("123456789","20110010411","F123","First Class",6,10);

insert into ticket values("147852366","20110010411","F234","Economy",5,21);

insert into ticket values("254789136","20110010318","F222","First Class",5,10);

insert into ticket values("314785299","20110011741","F123","First Class",6,12);

eticket	userid	flight_no	class	gate	seat_no
123456789	20110010411	F123	F123 lass	6	10
147852366	20110010411	F234	Economy	5	21
254789136	20110010318	F222	First Class	5	10
314785299	20110011741	F123	First Class	6	12
NULL	NULL	NULL	NULL	NULL	NULL

/* Creating table for ticket_flight_no */

create table ticket_flight_no(

flight_no varchar(30) primary key,

source varchar(30) not null,

destination varchar(30) not null,

date_time datetime not null

);

Field	Type	Null	Key	Default	Extra
flight_no	varchar(30)	NO	PRI	NULL	
source	varchar(30)	NO		NULL	
destination	varchar(30)	NO		NULL	
date_time	datetime	NO		NULL	

/* Inserting values into ticket_flight_no */

insert into ticket_flight_no values("F123","Hyderbad","Delhi",'2022-12-05 10:00:00');

insert into ticket_flight_no values("F222","Hyderbad","Mumbai",'2022-12-05 11:30:00');

insert into ticket_flight_no values("F234","Mumbai","Delhi",'2022-12-06 10:00:00');

flight_no	source	destination	date_time
F123	Hyderbad	Delhi	2022-12-05 10:00:00
F222	Hyderbad	Mumbai	2022-12-05 11:30:00
F234	Mumbai	Delhi	2022-12-06 10:00:00
NULL	NULL	NULL	NULL

/* Creating table for payment */

create table payment(

transaction_id varchar(30) primary key,

eticket varchar(30) not null,

account_id long not null,

amount long not null,

foreign key(eticket) references ticket(eticket)

);

Field	Type	Null	Key	Default	Extra
transaction_id	varchar(30)	NO	PRI	NULL	
eticket	varchar(30)	NO	MUL	NULL	
account_id	mediumtext	NO		NULL	
amount	mediumtext	NO		NULL	

/* Inserting values into payment */

insert into payment values("T147258","123456789",287456985,30000);

insert into payment values("T258147","147852366",287456985,45000);

insert into payment values("T857424","254789136",211400524,25000);

insert into payment values("T365666","314785299",285963777,30000);

transaction_id	eticket	account_id	amount
T147258	123456789	287456985	30000
T258147	147852366	287456985	45000
T365666	314785299	285963777	30000
T857424	254789136	211400524	25000
NULL	NULL	NULL	NULL

/* Creating table for airline */

```
create table airline(
airline_name varchar(30) primary key,
contact_no varchar(30) not null
);
```

Field	Type	Null	Key	Default	Extra
airline_name	varchar(30)	NO	PRI	NULL	
contact_no	mediumtext	NO		NULL	

/* Inserting values into airline */

```
insert into airline values("DENMARK","7894561230");
insert into airline values("BAJAJ","7778889990");
insert into airline values("AMAZON","8528528521");
```

airline_name	contact_no
AMAZON	8528528521
BAJAJ	7778889990
DENMARK	7894561230
NULL	NULL

/* Creating table for flight */

```
create table flight(
flight_no varchar(30) primary key,
airline_name varchar(30) not null,
```

flight_name varchar(30) not null,

dep_time time not null,

arr_time time not null,

seat_available int not null,

source varchar(30) not null,

destination varchar(30) not null,

foreign key(airline_name) references airline(airline_name)

);

Field	Type	Null	Key	Default	Extra
flight_no	varchar(30)	NO	PRI	NULL	
airline_name	varchar(30)	NO	MUL	NULL	
flight_name	varchar(30)	NO		NULL	
dep_time	time	NO		NULL	
arr_time	time	NO		NULL	
seat_available	int	NO		NULL	
source	varchar(30)	NO		NULL	
destination	varchar(30)	NO		NULL	

/* Inserting values into flight */

insert into flight
values("F123","DENMARK","DENMARK_F1",'10:00:00','14:00:00',20,"Hyderbad","Delhi");

insert into flight
values("F222","AMAZON","AMAZON_F1",'11:30:00','13:00:00',22,"Hyderbad","Mumbai");

insert into flight
values("F234","BAJAJ","BAJAJ_F1",'10:00:00','14:00:00',41,"Mumbai","Delhi");

flight_no	airline_name	flight_name	dep_time	arr_time	seat_available	source	destination
F123	DENMARK	DENMARK_F1	10:00:00	14:00:00	20	Hyderbad	Delhi
F222	AMAZON	AMAZON_F1	11:30:00	13:00:00	22	Hyderbad	Mumbai
F234	BAJAJ	BAJAJ_F1	10:00:00	14:00:00	41	Mumbai	Delhi
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

/* Creating table for user_flight */

create table user_flight(

userid varchar(30) not null,

flight_no varchar(30) not null,

foreign key(userid) references user(userid),

foreign key(flight_no) references flight(flight_no),

primary key(userid,flight_no)

);

	Field	Type	Null	Key	Default	Extra
▶	userid	varchar(30)	NO	PRI	NULL	
	flight_no	varchar(30)	NO	PRI	NULL	

/* Inserting values into user_flight*/

insert into user_flight values("20110010411","F123");

insert into user_flight values("20110010411","F234");

insert into user_flight values("20110010318","F222");

insert into user_flight values("20110011741","F234");

	userid	flight_no
▶	20110010411	F123
	20110010318	F222
	20110010411	F234
	20110011741	F234
*	NULL	NULL

/* Creating table for class */

create table class(

flight_no varchar(30) not null,

class_type varchar(30) not null,

fare long not null,

foreign key(flight_no) references flight(flight_no),

primary key(flight_no,class_type)

);

	Field	Type	Null	Key	Default	Extra
▶	flight_no	varchar(30)	NO	PRI	NULL	
	class_type	varchar(30)	NO	PRI	NULL	
	fare	mediumtext	NO		NULL	

/* Inserting values into class */

insert into class values("F123","First Class",30000);

insert into class values("F123","Economy",24000);

insert into class values("F234","First Class",50000);

insert into class values("F234","Economy",45000);

insert into class values("F222","First Class",25000);

	flight_no	class_type	fare
▶	F123	Economy	24000
	F123	First Class	30000
	F222	First Class	25000
	F234	Economy	45000
	F234	First Class	50000
*	NULL	NULL	NULL

3. SQL Queries

SQL:

select userid,eticket,amount from user natural join ticket natural join payment;

userid	eticket	amount
20110010318	254789136	25000
20110010411	123456789	30000
20110010411	147852366	45000
20110011741	314785299	30000

SQL:

select * from user natural join user_contact natural join user_pincode where userid="20110010411";

pincode	userid	fname	lname	dob	gender	mobile_no	state
522438	20110010411	Ram	Potheneni	2002-05-12	male	7418529630	AP
522438	20110010411	Ram	Potheneni	2002-05-12	male	9632587014	AP

SQL:

delete from payment where eticket="254789136";

delete from ticket where eticket="254789136";

transaction_id	eticket	account_id	amount
T147258	123456789	287456985	30000
T258147	147852366	287456985	45000
T365666	314785299	285963777	30000
NULL	NULL	NULL	NULL

eticket	userid	flight_no	class	gate	seat_no
123456789	20110010411	F123	First Class	6	10
147852366	20110010411	F234	Economy	5	21
314785299	20110011741	F123	First Class	6	12
NULL	NULL	NULL	NULL	NULL	NULL

SQL:

select * from flight natural join class where source='Hyderbad';

flight_no	airline_name	flight_name	dep_time	arr_time	seat_available	source	destination	class_type	fare
F123	DENMARK	DENMARK_F1	10:00:00	14:00:00	20	Hyderbad	Delhi	Economy	24000
F123	DENMARK	DENMARK_F1	10:00:00	14:00:00	20	Hyderbad	Delhi	First Class	30000
F222	AMAZON	AMAZON_F1	11:30:00	13:00:00	22	Hyderbad	Mumbai	First Class	25000

SQL:

select * from class natural join flight where class.class_type="Economy";

flight_no	class_type	fare	airline_name	flight_name	dep_time	arr_time	seat_available	source	destination
F123	Economy	24000	DENMARK	DENMARK_F1	10:00:00	14:00:00	20	Hyderbad	Delhi
F234	Economy	45000	BAJAJ	BAJAJ_F1	10:00:00	14:00:00	41	Mumbai	Delhi

Conclusion

Airline Booking System helps to store the user details, users can check for the flight availability. It stores the reserved ticket details, and their corresponding payment. Identifying the requirements, converting them into relational models and normalizing must be done carefully to avoid the data redundancy and inconsistency.

References

1. https://www.academia.edu/36303556/Database_Management_System_Project_on_Airlines_Reservation_Database_Design
2. https://www.youtube.com/watch?v=FcC8zhtOaSg
3. https://www.slideshare.net/shekhardesigner/airlines-database-design
4. https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description
5. https://www.javatpoint.com/dbms-normalization