SEN2104
DATABASE MANAGEMENT SYSTEMS
Project Report

Group No: 14
Project Title: BAU Plane Ticket System

Lab Section No: 901

Student Full Name: Berk Avcı 1804755
Begüm Doğru 1803672

1. Introduction

1.1Purpose/Project Proposal

The purpose of the project is supplying a airplane ticket system to customers.

1.2 Project Environment

We are going to use SQL.

1.3 Work Partitioning

A list showing all business events to which the work. (The following one is an example!)

Name	*Role*	*Description*
Berk Avcı 1804755	*Team member*	*I implemented the c,d, the half of the e and f statements in the given instruction*
Begüm Doğru 1803672	*Team member*	*I implemented the a,b, the half of the e and f statements in the given instruction*

2. BCNF

Flight:

flight_id,flight_date,destination,flight_time,plane_id

flight_id -> plane_id
flight_id -> flight_date,flight_time,destination
flight_date -> flight_time

Since all attributes are in atomic form it is 1NF.
Data does not require more than one column to qualify as a unique identifier. Thus, it is in 2NF

Canonical Form of the table:
Step 1: Obtain singleton RHS
flight_id -> plane_id
flight_id -> flight_date
flight_id -> flight_time
flight_id -> destination
flight_date -> flight_time
Step2: Remove extraneous attributes
Since left side is singleton,it can't contain any extraneous attribute.
Step3: Remove reduntans FDs
flight_id -> plane_id
flight_id -> flight_date

flight_id -> destination
flight_date -> flight_time

Step4: Final merged minimal cover:

flight -> plane_id, flight_date, destination
flight_date -> flight_time

BCNF:

(flight)+ = (flight_id, plane_id, flight_date, destination)

(flight_date)+ =(flight_date, flight_time) -> it violates the BCNF. because flight_date is neither a super key nor a superset thereof.

identifier = {flight_date}

Decomposition:

R1= (flight_date, flight_time)
R2=(flight_id,flight_date,plane_id,destination)

Airport:

airport_id,airport_city,airport_name

airport_id -> airport_city,airport_name

This table is already in BCNF form.

Airplane:

plane_id,airport_id

plane_id -> airport_id

This table is already in BCNF form.

Payment:

flight_id,customer_id,card_type,card_no,book_date

flight_id,customer_id -> card_type,card_no,book_date
card_type -> card_no

Since all attributes are in atomic form it is 1NF.
Data does not require more than one column to qualify as a unique identifier. Thus, it is in 2NF

Canonical Form of the table:

Step 1: Obtain singleton RHS

flight_id,customer_id -> card_type
flight_id,customer_id -> card_no
flight_id,customer_id -> book_date
card_type -> card_no

Step2: Remove extraneous attributes

(flight_id)+ = (flight_id,card_type,card_no,book_date)
(customer_id)+ = (customer_id,card_type,card_no,book_date)

flight_id,customer_id -> card_type
flight_id,customer_id -> card_no
flight_id,customer_id -> book_date
card_type -> card_no
(no extraneous attributes)

Step3: Remove reduntans FDs
flight_id,customer_id -> card_type
flight_id,customer_id -> book_date
card_type -> card_no

Step4: Final merged minimal cover:
flight_id,customer_id -> card_type,book_date
card_type -> card_no

BCNF:
(flight_id,customer_id)+ = (card_type,card_no,book_date)
(card_type)+ = (card_type,card_no)-> it violates the BCNF. because card_type is neither a super key nor a superset thereof.

Decomposition:
R1= (card_type,card_no)
R2=(flight_id,customer_id,card_type,book_date)

Customer:
customer_id,fname,lname,email,customer_address,telephone_no

customer_id,telephone_no,email -> fname,lname,customer_address
customer_id -> telephone_no,email
fname,lname -> customer_address

Since all attributes are in atomic form it is 1NF.
Data does not require more than one column to qualify as a unique identifier.
Thus, it is in 2NF

Canonical Form of the table:
Step 1: Obtain singleton RHS
customer_id,telephone_no,email ->fname
customer_id,telephone_no,email ->lname
customer_id,telephone_no,email ->customer_address
customer_id ->telephone_no
customer_id ->email
fname,lname ->customer_address

Step2: Remove extraneous attributes
(customer_id)+ = (customer_id,telephone_no,email,fname,lname,customer_address)
(telephone_no)+ = (telephone_no, fname, lname,customer_address)

(email)+ = (email,fname,lname,customer_address)

telephone_no and email can be removed since customer_id covers both of them.

Step3: Remove reduntans FDs

customer_id -> fname

customer_id -> lname

customer_id -> customer_address

customer_id -> telephone_no

customer_id -> email

fname,lname -> customer_address

Step4: Final merged minimal cover:

customer_id -> fname,lname,customer_address,telephone_no,email

fname,lname -> customer_address

BCNF:

(customer_id)+ = (customer_id,fname,lname,customer_address,telephone_no,email)

(fname,lname)+ =(fname,lname,customer_address) -> it violates the BCNF. not cover relation entirely

Decomposition:

R1= (fname,lname,customer_address)

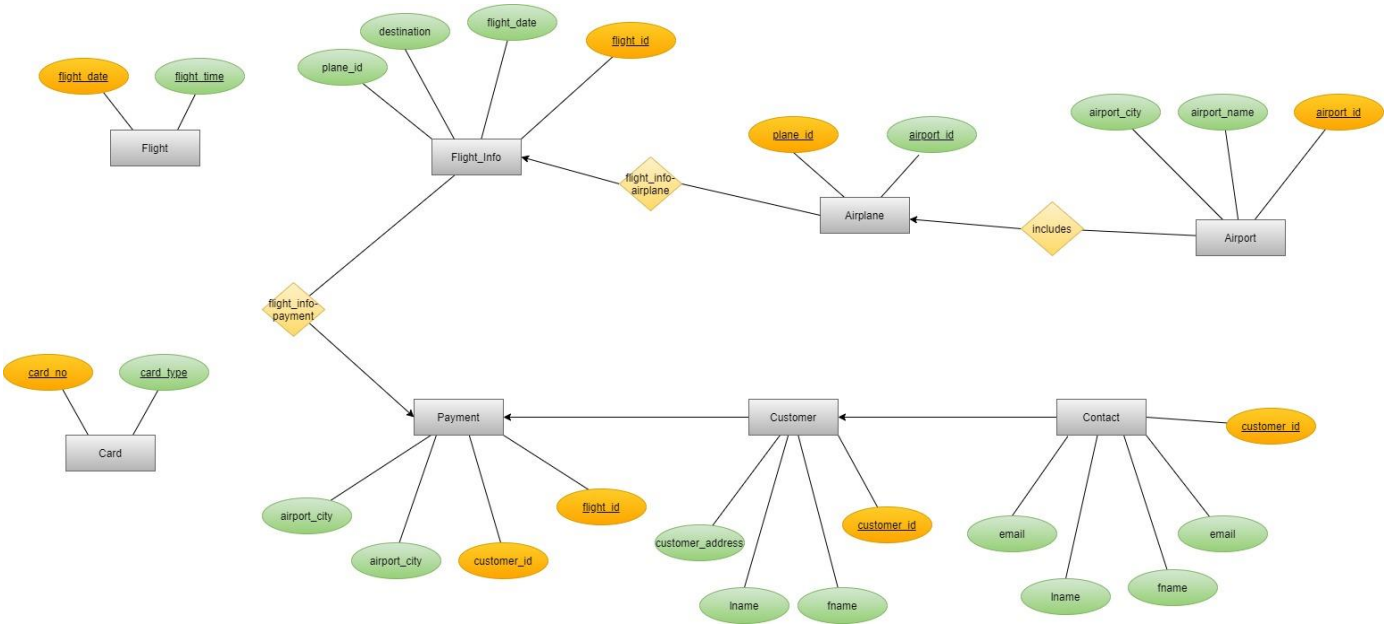
R2=(customer_id,telephone_no, email)

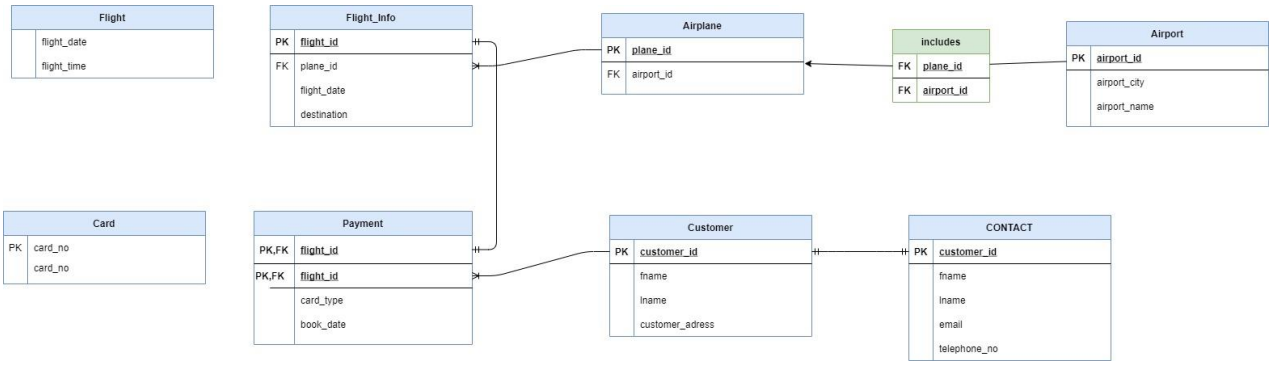
R2A=(customer_id,fname,lname,telephone_no,email)

Final Decomposition:

R1 and R2A

3. Architectural Representation





4.CREATE TABLE and INSERT INTO STATEMENTS

```
CREATE TABLE Customer(
customer_id int primary key,
fname VARCHAR(20),
lname VARCHAR(20),
customer_address VARCHAR(40)
);

CREATE TABLE CONTACT(
customer_id int primary key,
fname VARCHAR(20),
lname VARCHAR(20),
email VARCHAR(30),
telephone_no VARCHAR(15)
CONSTRAINT FK_CONTACT_Customer FOREIGN KEY(customer_id) REFERENCES Customer(customer_id)
);

CREATE TABLE Airport(
airport_id VARCHAR(3) primary key,
airport_city VARCHAR(20),
airport_name VARCHAR(30)
);

CREATE TABLE Airplane(
plane_id VARCHAR(15) primary key,
airport_id VARCHAR(3),
CONSTRAINT Airplane_fk FOREIGN KEY (airport_id) REFERENCES Airport(airport_id)
);

CREATE TABLE Flight_Info (
flight_id int primary key,
plane_id VARCHAR(15),
flight_date date,
destination VARCHAR(10),
CONSTRAINT Flight_Info_fk FOREIGN KEY (plane_id) REFERENCES Airplane(plane_id)
);
```

```
CREATE TABLE Flight(
flight_date date,
flight_time timestamp
);

CREATE TABLE Payment(
flight_id int PRIMARY KEY,
customer_id int PRIMARY KEY,
card_type VARCHAR(10),
book_date date,
CONSTRAINT Payment_fk FOREIGN KEY(customer_id) REFERENCES Customer(customer_id),
CONSTRAINT FK_Payment_Flight_Info FOREIGN KEY (flight_id) REFERENCES Flight_Info(flight_id)

);

CREATE TABLE Card(
card_no VARCHAR(9),
card_type VARCHAR(10),
CONSTRAINT Card_fk PRIMARY KEY(card_no)
);
CREATE TABLE includes(
airport_id VARCHAR(3),
plane_id VARCHAR(15),
CONSTRAINT airport_id_fk FOREIGN KEY (airport_id) REFERENCES Airport(airport_id),
CONSTRAINT plane_id_fk FOREIGN KEY (plane_id) REFERENCES Plane(plane_id)

);


INSERT INTO Airport VALUES ('IST','Istanbul','Istanbul Airport');
INSERT INTO Airport VALUES ('ADB','Izmir','İzmir Adnan Menderes Airport');
INSERT INTO Airport VALUES ('ESB','Ankara','Ankara Esenboğa Airport');

INSERT INTO Airplane VALUES ('BOEING 727','IST');
INSERT INTO Airplane VALUES ('DC9','ADB');
INSERT INTO Airplane VALUES ('BOEING 747','ESB');

INSERT INTO Customer VALUES (1,'Berk','Avcı','Kadıköy/İstanbul');
INSERT INTO Customer VALUES (2,'Begüm','Doğru','Ataşehir/İstanbul');
INSERT INTO Customer VALUES (3,'Mesut','Özil','Sarıyer/İstanbul');
INSERT INTO Customer VALUES (4,’LeBron James',’LosAngeles/USA’);
INSERT INTO Customer VALUES (5,’Tom Brady’,’Kansas/USA’);
INSERT INTO Customer VALUES (6,’Nazım Sangare’,’Kadıköy/İstanbul’);


INSERT INTO CONTACT VALUES (1,'Berk','Avcı','berkavci@windowslive.com','05075080498');
INSERT INTO CONTACT VALUES (2,'Begüm','Doğru','begumdogru@hotmail.com','05389857714');
INSERT INTO CONTACT VALUES (3,'Mesut','Özil','ozilmesut@hotmail.com','05356548798');

INSERT INTO Flight_Info VALUES (102,'BOEING 727',TO_DATE('05/25/2021','MM/DD/YYYY'),'Istanbul');
INSERT INTO Flight_Info VALUES (329,'DC9',TO_DATE('05/25/2021','MM/DD/YYYY'),'Izmir');
INSERT INTO Flight_Info VALUES (507,'BOEING 747',TO_DATE('05/25/2021','MM/DD/YYYY'),'Ankara');
```

INSERT INTO Flight VALUES (TO_DATE('05/25/2021','MM/DD/YYYY'),TO_TIMESTAMP('09:00','HH.MI'));
INSERT INTO Flight VALUES (TO_DATE('05/25/2021','MM/DD/YYYY'),TO_TIMESTAMP('09:10','HH.MI'));
INSERT INTO Flight VALUES (TO_DATE('05/25/2021','MM/DD/YYYY'),TO_TIMESTAMP('12:45','HH.MI'));

INSERT INTO Payment VALUES (102, 1, 'VISA',TO_DATE('05/04/2021','MM/DD/YYYY'));
INSERT INTO Payment VALUES (329, 2, 'MASTER',TO_DATE('05/04/2021','MM/DD/YYYY'));
INSERT INTO Payment VALUES (507, 3, 'VISA',TO_DATE('05/04/2021','MM/DD/YYYY'));

INSERT INTO Card VALUES ('5235758','VISA');
INSERT INTO Card VALUES ('5891345','MASTER');
INSERT INTO Card VALUES ('4331001', 'VISA');

5)SQL Queries

-- Find the names and telephone numbers of the customers. (JOIN)

SELECT customer.fname, customer.lname, telephone_no
FROM customer, contact
WHERE customer.customer_id = contact.customer_id;

```
SELECT customer.fname, customer.lname, telephone_no
FROM customer, contact
WHERE customer.customer_id = contact.customer_id;
```

Script Output | Query Result

Task completed in 0.256 seconds

FNAME	LNAME	TELEPHONE_NO
Berk	Avcı	05075080498
Begüm	Doğru	05389857714
Mesut	Özil	05356548798

-- Find the airport name and the airport city of the plane Boeing 727 (JOIN)

SELECT airport_city, airport_name, plane_id

FROM airport, airplane
WHERE airport.airport_id = airplane.airport_id and plane_id = 'BOEING 727';

```
SELECT airport_city, airport_name, plane_id
FROM airport, airplane
WHERE airport.airport_id = airplane.airport_id and plane_id = 'BOEING 727';
```

Script Output | Query Result
Task completed in 0.265 seconds

```
AIRPORT_CITY         AIRPORT_NAME                   PLANE_ID
-------------------- ------------------------------ ---------------
Istanbul             Istanbul Airport               BOEING 727
```

--Find the customer's name who has VISA card type. (JOIN)

SELECT lname, fname, card_type
FROM customer , payment
WHERE customer.customer_id = payment.customer_id and card_type = 'VISA';

```
SELECT lname, fname, card_type
FROM  customer , payment
WHERE customer.customer_id = payment.customer_id and card_type = 'VISA';
```

Script Output | Query Result
Task completed in 0.262 seconds

```
LNAME                FNAME                CARD_TYPE
-------------------- -------------------- ----------
Avcı                 Berk                 VISA
Özil                 Mesut                VISA
```

--Find the customer name who has VISA card type. (NESTED)

SELECT lname, fname, card_type
FROM customer , payment
WHERE customer.customer_id = payment.customer_id and card_type

IN
(SELECT card_type
FROM payment
WHERE UPPER(card_type) = 'VISA');

```
SELECT lname, fname, card_type
FROM  customer , payment
WHERE customer.customer_id = payment.customer_id  and card_type
IN
(SELECT card_type
 FROM payment
 WHERE UPPER(card_type) = 'VISA');
```

Script Output | Query Result

Task completed in 0.323 seconds

```
LNAME                FNAME                CARD_TYPE
-------------------- -------------------- ----------
Özil                 Mesut                VISA
Avcı                 Berk                 VISA
```

-- Find the airport name and the airport city of the plane Boeing 727 (NESTED)

SELECT airport_city, airport_name, plane_id
FROM airport, airplane
WHERE airport.airport_id = airplane.airport_id and plane_id =
(SELECT plane_id
FROM airplane
WHERE UPPER(plane_id) = 'BOEING 727');

```
SELECT airport_city, airport_name, plane_id
FROM airport, airplane
WHERE airport.airport_id = airplane.airport_id and plane_id =
(SELECT plane_id
FROM airplane
WHERE UPPER(plane_id)  = 'BOEING 727');
```

Script Output | Query Result

Task completed in 0.333 seconds

```
AIRPORT_CITY         AIRPORT_NAME                   PLANE_ID
-------------------- ------------------------------ ---------------
Istanbul             Istanbul Airport               BOEING 727
```

--Display all the customer's name and ID(SET)

SELECT customer_id,fname, lname FROM contact
UNION
SELECT customer_id,fname, lname FROM customer;

```
SELECT customer_id,fname, lname FROM contact
UNION
SELECT customer_id,fname, lname FROM customer;
```

Script Output | Query Result

Task completed in 0.261 seconds

```
CUSTOMER_ID FNAME                LNAME
----------- -------------------- --------------------
          1 Berk                 Avcı
          2 Begüm                Doğru
          3 Mesut                Özil
          4 LeBron               James
          5 Tom                  Brady
          6 Nazım                Sangare
```

--Display customers' ID who have already paid(SET)

select customer_id from CUSTOMER
intersect
select customer_id from Payment

```
select customer_id from CUSTOMER
intersect
select customer_id from Payment
```

Script Output | Query Result

Task completed in 0.259 seconds

```
CUSTOMER_ID
-----------
          1
          2
          3
```

--Display the number of customers who already paid(AGG)

SELECT count (customer_id)
from customer
where customer_id

IN(
select customer_id
from payment
);

```
SELECT count ( customer_id )
from customer
where customer_id
IN(
select customer_id
from payment
);
```

Script Output | Query Result

Task completed in 0.254 secor

```
COUNT(CUSTOMER_ID)
------------------
                 3
```

--Display the earliest flight time of that day(AGG)

select MIN(flight_time)
from FLIGHT,Flight_info
where flight.flight_date = flight_info.flight_date

```
select MIN(flight_time)
from FLIGHT,Flight_info
where flight.flight_date = flight_info.flight_date
```

Script Output | Query Result

Task completed in 0.261 seconds

```
MIN(FLIGHT_TIME)
----------------------------
01/05/2021 00:45:00,000000000
```