CS331 Term Project Final Report- Airline Reservation

CS331

Section 003

Group11

Group 11 Members:
Jose Cabrera (jmc258@njit.edu)
Dustin La (drl3@njit.edu)
Natalia Smith (nrs5@njit.edu)

2. Summary of System Requirements - Airline Reservation

System Requirements

- **Passengers** are identified by a passenger ID and have a name, and birthdate. Passengers are able to book tickets for flights and have their own baggage. Passengers are able to book multiple tickets.

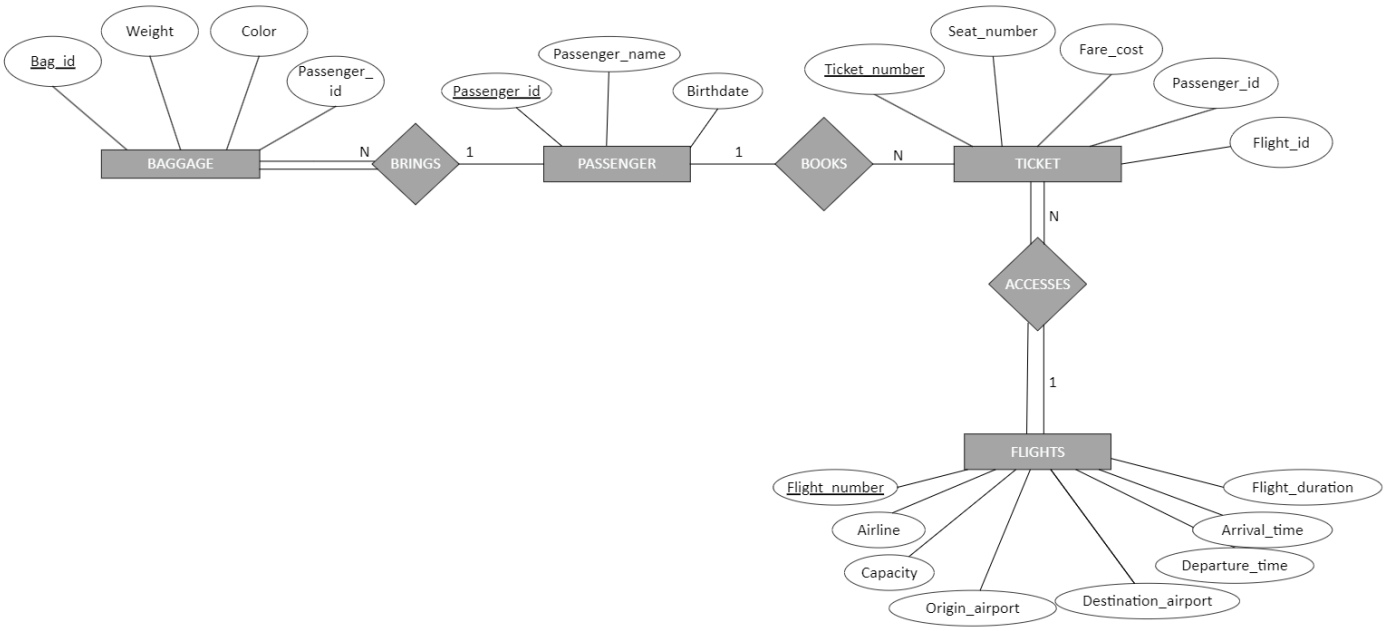
- **Tickets** are identified by ticket number and have a seat number, fare cost, passenger ID and flight ID. All flights must have a ticket.

- **Baggage** are identified by a bag ID and have weight, color, and passenger ID. A passenger can have zero, one, or multiple bags.

- **Flights** are identified by flight number and have an airline, capacity, origin airport, destination airport, departure time, arrival time, and flight duration. Flights can have multiple tickets. Tickets are required to access flights.

We chose these four entities as it shows the most vital processes used when booking a flight. Passengers are able to book tickets and have baggage. Passengers can also book multiple tickets and have multiple baggage. We made tickets required to access Flights to ensure all flights had tickets. We originally had more entities which we later removed due to redundancy. We instead made these redundant entities into attributes of the four entities. Some of these redundant or unnecessary entities made into attributes include fares, airline, and airport. Additionally, some of our previous entities had attributes that were more vague or not represented well in a database. We instead took these attributes and split them into specific attributes. An example of this was Flight_information as an entity, made into an attribute, and finally split into the attributes found in Flights which include departure time, arrival time, and flight duration.

2. Entity-Relationship Diagram



Our Entity-Relationship Diagram is seen above. These include all four entities with three connection relations. The attributes are based on the system requirements.

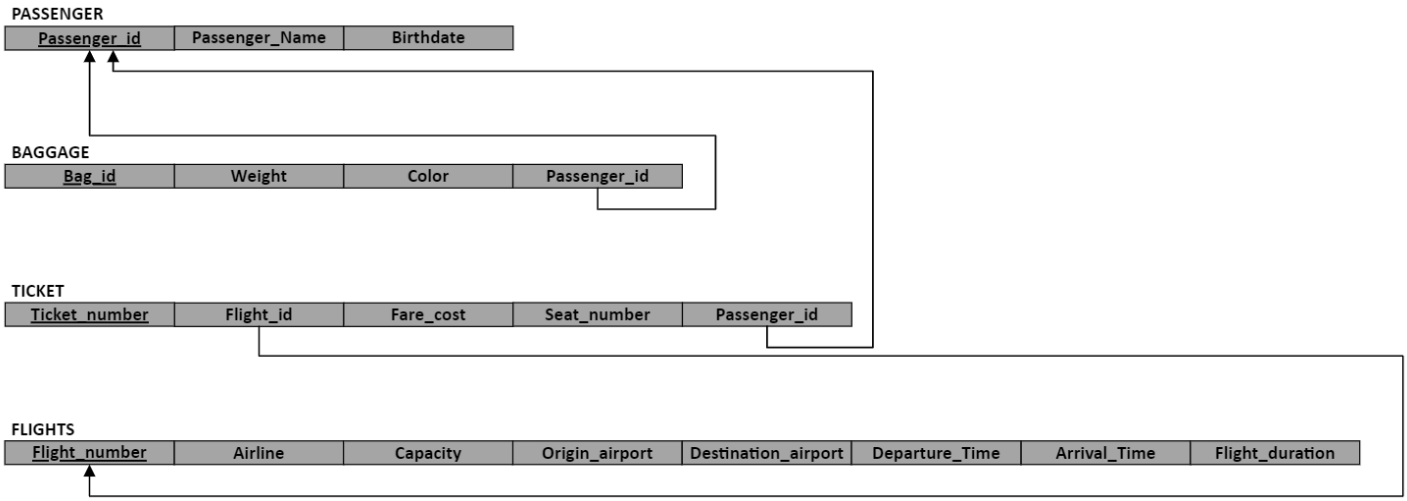
This Entity-Relationship Diagram is leaps and bounds improved versus our Entity-Relationship Diagram for Phase 1. In our original Entity-Relationship Diagram, we had many unnecessary relations (such as HAVE which was redundant), as well as not having helpful entities such as Ticket. We took all of the feedback given to us after Phase 1 and made a much clearer and cleaner Entity-Relationship diagram.

For BAGGAGE and PASSENGER, we made it N:1 as Passengers can bring multiple baggage and multiple baggage can come from one passenger. Passengers are able to have zero to multiple baggage. We also made baggage total participation because if passengers do have baggage, this relation must exist.

For PASSENGER and TICKET, we made it 1:N as a single passenger can book multiple tickets and multiple tickets can belong to a single passenger.

For TICKET and FLIGHTS, we made it N:1 as multiple tickets can be used to access flights and a single flight can have multiple tickets. We made both sides of this relation total participation as a ticket is needed to access a flight and a flight must have a ticket.

2. Entity-Relationship Schema



Seen here is our Entity-Relationship Schema. It includes all the attributes from the system requirements for each entity.

Our original relational schema implied that a Passenger could only own a single ticket, so we fixed that for our current schema. We also got rid of the entities that were no longer in our updated Entity-Relationship Diagram, such as Fare.

PASSENGER has the primary key of Passenger_id along with attributes of Passenger_Name and Birthdate.

BAGGAGE has the primary key of Bag_id and attributes of weight, color, and Passenger_id. Passenger_id is a foreign key that references the PASSENGER entity.

TICKET has the primary key of Ticket_number and attributes of Flight_id, Fare_cost, Seat_number, and Passenger_id. Flight_id is a foreign key that references the FLIGHTS entity and Passenger_id is a foreign key that references the PASSENGER entity.

FLIGHTS has the primary key of Flight_number that has the attributes of Airline, Capacity, Origin_airport, Destination_airport, Departure_Time, Arrival_time, and Flight_duration.

Our SQL Code: Table Creation

FLIGHTS Table

```
CREATE TABLE FLIGHTS (
   Flight_number VARCHAR(30),
   Airline VARCHAR(64) NOT NULL,
   Capacity INT NOT NULL,
   Origin_airport VARCHAR(64) NOT NULL,
   Destination_airport VARCHAR(64) NOT NULL,
   Departure_time Timestamp NOT NULL,
   Arrival_Time Timestamp NOT NULL,
   Flight_duration CHAR(8) NOT NULL,
   PRIMARY KEY (Flight_number)
);
```

PASSENGER Table

```
CREATE TABLE PASSENGER (
   Passenger_id VARCHAR(30),
   Passenger_name VARCHAR(64) NOT NULL,
   Birthdate DATE NOT NULL,
   PRIMARY KEY (Passenger_id)
);
```

TICKET Table

```
CREATE TABLE TICKET (
   Ticket_number VARCHAR(30),
   Seat_number VARCHAR(30) NOT NULL,
   Fare_cost FLOAT NOT NULL,
   Passenger_id VARCHAR(30) NOT NULL,
   Flight_id VARCHAR(30) NOT NULL,
   PRIMARY KEY (Ticket_number),
   FOREIGN KEY (Passenger_id) REFERENCES PASSENGER (Passenger_id)
   ON DELETE CASCADE,
   FOREIGN KEY (Flight_id) REFERENCES FLIGHTS (Flight_number)
   ON DELETE CASCADE
```

);

BAGGAGE Table

```
CREATE TABLE BAGGAGE (
   Bag_id VARCHAR(30),
   Weight FLOAT NOT NULL,
   Color VARCHAR(30) NOT NULL,
   Passenger_id VARCHAR(30) NOT NULL,
   PRIMARY KEY (Bag_id),
   FOREIGN KEY (Passenger_id) REFERENCES PASSENGER (Passenger_id)
   ON DELETE CASCADE
);
```

Insertions

Inserting into FLIGHTS Table

```
INSERT INTO FLIGHTS(Flight_number, Airline, Capacity, Origin_airport, Destination_airport, Departure_Time, Arrival_Time, Flight_duration)
VALUES ('UA1104', 'United Airlines', '412', 'Newark Liberty International Airport', 'Luis Munoz Marin International Airport', TO_DATE('2022-12-20 09:00:00','YYYY-MM-DD HH24:MI:SS'), TO_DATE('2022-12-20 13:20:00','YYYY-MM-DD HH24:MI:SS'), '04:20:00');

INSERT INTO FLIGHTS(Flight_number, Airline, Capacity, Origin_airport, Destination_airport, Departure_Time, Arrival_Time, Flight_duration)
VALUES ('JB2252', 'JetBlue', '200', 'Birmingham-Shuttlesworth International Airport', 'Cyril E. King Airport', TO_DATE('2022-11-16 07:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_DATE('2022-11-16 10:40:00', 'YYYY-MM-DD HH24:MI:SS'), '03:40:00');

INSERT INTO FLIGHTS(Flight_number, Airline, Capacity, Origin_airport, Destination_airport, Departure_Time, Arrival_Time, Flight_duration)
VALUES ('AA7489', 'American Airlines', '300', 'Henry E. Rohlsen Airport', 'Pago Pago International Airport', TO_DATE('2023-04-05 06:30:00', 'YYYY-MM-DD HH24:MI:SS'), TO_DATE('2023-04-06 18:15:00', 'YYYY-MM-DD HH24:MI:SS'), '35:45:00');

INSERT INTO FLIGHTS(Flight_number, Airline, Capacity, Origin_airport, Destination_airport, Departure_Time, Arrival_Time, Flight_duration)
VALUES ('NK9384', 'Spirit Airlines', '250', 'Sheridan County Airport', 'Kahului Airport', TO_DATE('2023-08-08 08:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_DATE('2023-08-09 22:35:00', 'YYYY-MM-DD HH24:MI:SS'), '14:35:00');
```

INSERT INTO FLIGHTS(Flight_number, Airline, Capacity, Origin_airport, Destination_airport, Departure_Time, Arrival_Time, Flight_duration)
VALUES ('DL6457', 'Delta Airlines', '300', 'John Wayne Airport', 'Palm Beach International Airport', TO_DATE('2023-07-14 05:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_DATE('2023-07-14 16:30:00', 'YYYY-MM-DD HH24:MI:SS'), '11:30:00');

Inserting into PASSENGER Table

INSERT INTO PASSENGER (Passenger_id, Passenger_name, Birthdate)
VALUES ('L12345678901234', 'Dustin R La', TO_DATE('2001-12-07','yyyy-mm-dd'));

INSERT INTO PASSENGER (Passenger_id, Passenger_name, Birthdate)
VALUES ('C37482746371623', 'Natalia R Smith', TO_DATE('2002-08-22','yyyy-mm-dd'));

INSERT INTO PASSENGER (Passenger_id, Passenger_name, Birthdate)
VALUES ('B19873658071334', 'Jose M Cabrera', TO_DATE('2002-09-25','yyyy-mm-dd'));

INSERT INTO PASSENGER (Passenger_id, Passenger_name, Birthdate)
VALUES ('D78123648271356', 'Rishi Sakunala', TO_DATE('2002-07-05','yyyy-mm-dd'));

INSERT INTO PASSENGER (Passenger_id, Passenger_name, Birthdate)
VALUES ('H21978346981237', 'Gautham Nagrajan', TO_DATE('2002-03-12','yyyy-mm-dd'));

Inserting into TICKET Table

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)
VALUES('98372049857281', '34B' ,'238.21', 'L12345678901234', 'UA1104');

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)
VALUES('45758362801283', '22A', '216.23', 'C37482746371623', 'JB2252');

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)
VALUES('75845610293613', '46B', '522.43', 'B19873658071334', 'AA7489');

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)
VALUES('19238476193242', '67D', '522.43', 'D78123648271356', 'NK9384');

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)
VALUES('21598724297644', '26G', '478.32', 'H21978346981237', 'DL6457');

Inserting to BAGGAGE Table

```
INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('AHGIU56739', '27.23', 'Blue', 'L12345678901234');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('DSGER78134', '25.72', 'Blue', 'C37482746371623');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('UIOSE83754', '31.67', 'Green', 'B19873658071334');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('NIMDT56129', '49.99', 'Red', 'D78123648271356');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('ALPID64812', '20.82', 'Orange', 'H21978346981237');
```

Tables after Insertions

FLIGHTS:

	FLIGHT_NUMBER	AIRLINE	CAPACITY	ORIGIN_AIRPORT	DESTINATION_AIRPORT	DEPARTURE_TIME	ARRIVAL_TIME	FLIGHT_DURATION
1	UA1104	United Airlines	412	Newark Liberty International Airport	Luis Munoz Marin International Airport	20-DEC-22 09.00.00.000000000 AM	20-DEC-22 01.20.00.000000000 PM	04:20:00
2	JB2252	JetBlue	200	Birmingham-Shuttlesworth International Airport	Cyril E. King Airport	16-NOV-22 07.00.00.000000000 AM	16-NOV-22 10.40.00.000000000 AM	03:40:00
3	AA7489	American Airlines	300	Henry E. Rohlsen Airport	Pago Pago International Airport	05-APR-23 06.30.00.000000000 AM	06-APR-23 06.15.00.000000000 PM	35:45:00
4	NK9384	Spirit Airlines	250	Sheridan County Airport	Kahului Airport	08-AUG-23 08.00.00.000000000 AM	09-AUG-23 10.35.00.000000000 PM	14:35:00
5	DL6457	Delta Airlines	300	John Wayne Airport	Palm Beach International Airport	14-JUL-23 05.00.00.000000000 AM	14-JUL-23 04.30.00.000000000 PM	11:30:00

PASSENGER:

	PASSENGER_ID	PASSENGER_NAME	BIRTHDATE
1	L12345678901234	Dustin R La	07-DEC-01
2	C37482746371623	Natalia R Smith	22-AUG-02
3	B19873658071334	Jose M Cabrera	25-SEP-02
4	D78123648271356	Rishi Sakunala	05-JUL-02
5	H21978346981237	Gautham Nagrajan	12-MAR-02

TICKET:

	TICKET_NUMBER	SEAT_NUMBER	FARE_COST	PASSENGER_ID	FLIGHT_ID
1	98372049857281	34B	238.21	L12345678901234	UA1104
2	45758362801283	22A	216.23	C37482746371623	JB2252
3	75845610293613	46B	522.43	B19873658071334	AA7489
4	19238476193242	67D	522.43	D78123648271356	NK9384
5	21598724297644	26G	478.32	H21978346981237	DL6457

BAGGAGE:

	BAG_ID	WEIGHT	COLOR	PASSENGER_ID
1	AHGIU56739	27.23	Blue	L12345678901234
2	DSGER78134	25.72	Blue	C37482746371623
3	UIOSE83754	31.67	Green	B19873658071334
4	NIMDT56129	49.99	Red	D78123648271356
5	ALPID64812	20.82	Orange	H21978346981237

Updates and Deletions

UPDATE and DELETE on PASSENGER

```
UPDATE PASSENGER
SET Passenger_name = 'Hrishikesh Sakunala'
WHERE Passenger_id = 'D78123648271356';

DELETE
FROM PASSENGER
WHERE Passenger_name = 'Gautham Nagrajan';
```

UPDATE and DELETE on BAGGAGE

```
UPDATE Baggage
SET Passenger_id = 'D78123648271356'
WHERE Bag_id = 'DSGER78134';

DELETE
FROM BAGGAGE
WHERE Bag_id = 'UIOSE83754';
```

UPDATE and DELETE on TICKET

```
UPDATE TICKET
SET Flight_id = 'AA7489'
WHERE Ticket_number = '45758362801283';

DELETE
FROM TICKET
WHERE Ticket_number = '98372049857281';
```

UPDATE and DELETE on FLIGHTS

```
UPDATE FLIGHTS
SET Destination_airport = 'Kahului Airport'
WHERE Flight_number = 'JB2252';

DELETE
FROM FLIGHTS
WHERE Flight_Number = 'UA1104';
```

Tables after Updates and Deletes

FLIGHTS:

	FLIGHT_NUMBER	AIRLINE	CAPACITY	ORIGIN_AIRPORT	DESTINATION_AIRPORT	DEPARTURE_TIME	ARRIVAL_TIME	FLIGHT_DURATION
1	JB2252	JetBlue	200	Birmingham-Shuttlesworth International Airport	Kahului Airport	16-NOV-22 07.00.00.000000000 AM	16-NOV-22 10.40.00.000000000 AM	03:40:00
2	AA7489	American Airlines	300	Henry E. Rohlsen Airport	Pago Pago International Airport	05-APR-23 06.30.00.000000000 AM	06-APR-23 06.15.00.000000000 PM	35:45:00
3	NK9384	Spirit Airlines	250	Sheridan County Airport	Kahului Airport	08-AUG-23 08.00.00.000000000 AM	09-AUG-23 10.35.00.000000000 PM	14:35:00
4	JK1257	Delta Airlines	300	John Wayne Airport	Palm Beach International Airport	14-JUL-23 05.00.00.000000000 AM	14-JUL-23 04.30.00.000000000 PM	11:30:00

PASSENGER:

	PASSENGER_ID	PASSENGER_NAME	BIRTHDATE
1	L12345678901234	Dustin R La	07-DEC-01
2	C37482746371623	Natalia R Smith	22-AUG-02
3	B19873658071334	Jose M Cabrera	25-SEP-02
4	D78123648271356	Hrishikesh Sakunala	05-JUL-02

TICKET:

	TICKET_NUMBER	SEAT_NUMBER	FARE_COST	PASSENGER_ID	FLIGHT_ID
1	45758362801283	22A	216.23	C37482746371623	AA7489
2	75845610293613	46B	522.43	B19873658071334	AA7489
3	19238476193242	67D	310.21	D78123648271356	NK9384

BAGGAGE:

	BAG_ID	WEIGHT	COLOR	PASSENGER_ID
1	AHGIU56739	27.23	Blue	L12345678901234
2	DSGER78134	25.72	Blue	D78123648271356
3	NIMDT56129	49.99	Red	D78123648271356

3. Normalization of Relations

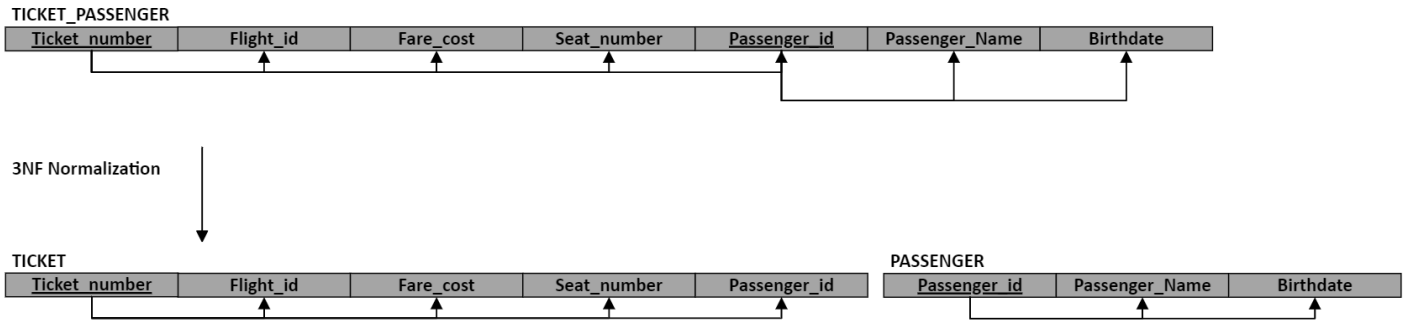
A. TICKET_PASSENGER
(Ticket_number, Fare_cost, Airline, Seat_number, Passenger_id, Flight_id, Passenger_name, Birthdate)

a. Primary key: Ticket_number, Passenger_id
Foreign key: Flight_id

B. SELECT *
FROM TICKET, PASSENGER
WHERE TICKET.Passenger_id = PASSENGER.Passenger_id;

	TICKET_NUMBER	SEAT_NUMBER	FARE_COST	PASSENGER_ID	FLIGHT_ID	PASSENGER_ID_1	PASSENGER_NAME	BIRTHDATE
1	98372049857281	34B	238.21	L12345678901234	UA1104	L12345678901234	Dustin R La	07-DEC-01
2	45758362801283	22A	216.23	C37482746371623	JB2252	C37482746371623	Natalia R Smith	22-AUG-02
3	75845610293613	46B	522.43	B19873658071334	AA7489	B19873658071334	Jose M Cabrera	25-SEP-02
4	19238476193242	67D	310.21	D78123648271356	NK9384	D78123648271356	Rishi Sakunala	05-JUL-02
5	21598724297644	26G	478.32	H21978346981237	JK1257	H21978346981237	Gautham Nagrajan	12-MAR-02

C/D. The primary keys of the relation are Ticket_number and Passenger_id with Flight_id being a foreign key of FLIGHTS. This relation is in 2NF as all non-prime attributes depend on a primary key and it does not feature any composite or multi-valued attributes. The relation however is not in 3NF and required further normalization to reach 3NF where the passenger and ticket are split apart because Passenger_id's dependent attributes are transitively dependent on Ticket_number, due to the fact that a passenger may have multiple tickets thus the passenger's ID cannot determine a specific ticket. This normalized relation is also in BCNF as the non-prime attributes all depend on a prime attribute. None of the attributes other than Ticket_number in TICKET are part of the key. Additionally for PASSENGER none of the attributes other than Passenger_id are part of the key.



A. PASSENGER_BAGGAGE

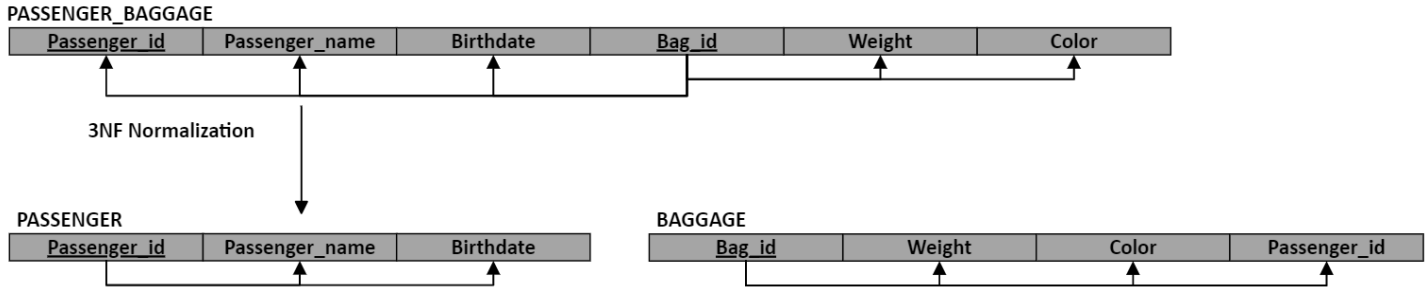
(Passenger_id, Passenger_name, Birthdate, Bag_id, Weight, Color)

- Primary key: Passenger_id, Bag_id

B. SELECT *
FROM BAGGAGE, PASSENGER
WHERE BAGGAGE.Passenger_id = PASSENGER.Passenger_id;

	BAG_ID	WEIGHT	COLOR	PASSENGER_ID	PASSENGER_ID_1	PASSENGER_NAME	BIRTHDATE
1	AHGIU56739	27.23	Blue	L12345678901234	L12345678901234	Dustin R La	07-DEC-01
2	DSGER78134	25.72	Blue	C37482746371623	C37482746371623	Natalia R Smith	22-AUG-02
3	UIOSE83754	31.67	Green	B19873658071334	B19873658071334	Jose M Cabrera	25-SEP-02
4	NIMDT56129	49.99	Red	D78123648271356	D78123648271356	Rishi Sakunala	05-JUL-02
5	ALPID64812	20.82	Orange	H21978346981237	H21978346981237	Gautham Nagrajan	12-MAR-02

C/D. The primary keys in this relation are Passenger_id and Bag_id with no foreign keys from other tables. This relation is in 2NF as the non-prime attributes all depend on a primary key and there are no composite or multi-valued attributes. The relation cannot be said to be in 3NF however, as both the Passenger_id's dependent attributes rely on Bag_id transitively to determine unique values, as a Passenger_id can have multiple Bag_ids assigned to them but Bag_ids can only ever have one Passenger_id, allowing all Bag_ids to specifically determine a Passenger_id. This normalized relation is also in BCNF as the non-prime attributes all depend on a prime attribute. None of the attributes other than Bag_id in BAGGAGE are part of the key. Additionally for PASSENGER none of the attributes other than Passenger_id are part of the key.



A. TICKET_FLIGHTS

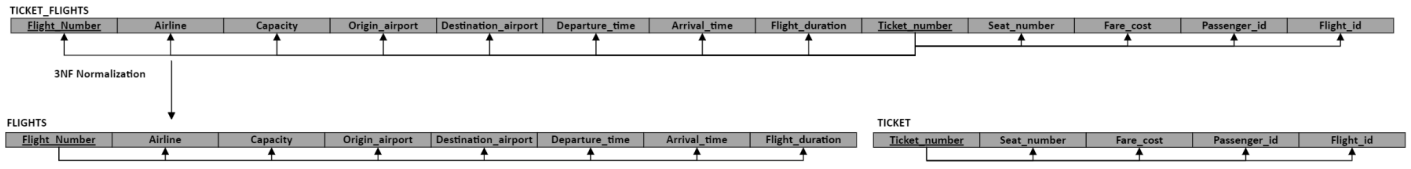
(Ticket_number, Fare_cost, Airline, Seat_number, Passenger_id, Flight_id, Airline, Capacity, Origin_airport, Destination_airport, Departure_time, Arrival_time, Total_time)

- Primary key: Flight_number, Ticket_number
- Foreign key: Passenger_id, Flight_id

B. SELECT *
FROM TICKET, FLIGHTS
WHERE TICKET.Flight_id = FLIGHTS.Flight_number;

	TICKET_NUMBER	SEAT_NUMBER	FARE_COST	PASSENGER_ID	FLIGHT_ID	FLIGHT_NUMBER	AIRLINE	CAPACITY	ORIGIN_AIRPORT	DESTINATION_AIRPORT	DEPARTURE_TIME	ARRIVAL_TIME	FLIGHT_DURATION
1	98372049857	34B	238.21	L12345678901234	UA1104	UA1104	United Airlines	412	Newark Liberty Internat...	Luis Munoz Marin Internat...	20-DEC-22 09.00...	20-DEC-22 01.20...	04:20:00
2	45758362801283	22A	216.23	C3748274637162	JB2252	JB2252	JetBlue	200	Birmingham-Shuttleswort...	Cyril E. King Airport	16-NOV-22 07.00...	16-NOV-22 10.40...	03:40:00
3	75845610293613	46B	522.43	B19873658071334	AA7489	AA7489	American Airlines	300	Henry E. Rohlsen Airport	Pago Pago International A...	05-APR-23 06.30...	06-APR-23 06.15...	35:45:00
4	19238476193242	67D	310.21	D78123648271356	NK9384	NK9384	Spirit Airlines	250	Sheridan County Airport	Kahului Airport	08-AUG-23 08.00...	09-AUG-23 10.35...	14:35:00
5	21598724297644	26G	478.32	H21978346981237	JK1257	JK1257	Delta Airlines	300	John Wayne Airport	Palm Beach International ...	14-JUL-23 05.00...	14-JUL-23 04.30...	11:30:00

C/D. The Primary keys in this relation are Flight_number and Ticket_number with the Passenger_id being a foreign key from PASSENGER and the Flight_id being a foreign key in TICKET from FLIGHTS. This relation is also in 2NF as its non-prime attributes are all depending on the primary keys of Ticket_number and Flight_id and none of the attributes are multi-valued or composite. The relation can be decomposed into 3NF by splitting the Flights and Tickets, as a Flight can be uniquely identified by tickets, but since each flight can have multiple tickets we cannot uniquely determine a ticket number based on the flight. This makes a flight's attributes transitively dependent on Ticket_number. This normalized relation is also in BCNF as the non-prime attributes all depend on a prime attribute. None of the attributes other than Ticket_number in TICKET are part of the key. Additionally for FLIGHTS none of the attributes other than Flight_Number are part of the key. In theory a ticket's number would be determined by a combination of the Passenger_id and Flight_id but because a passenger can buy several tickets for one flight both of these cannot determine a Ticket_number alone nor together.



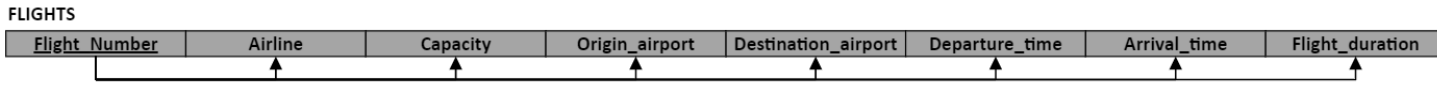
A. FLIGHTS (Flight_number, Airline, Capacity, Origin_airport, Destination_airport, Departure_time, Arrival_time, Total_time)

- Primary key: Flight_number

B. SELECT *
FROM FLIGHTS;

	FLIGHT_NUMBER	AIRLINE	CAPACITY	ORIGIN_AIRPORT	DESTINATION_AIRPORT	DEPARTURE_TIME	ARRIVAL_TIME	FLIGHT_DURATION
1	UA1104	United Airlines	412	Newark Liberty International Airport	Luis Munoz Marin International Airport	20-DEC-22 09.00.00.000000000 AM	20-DEC-22 01.20.00.000000000 PM	04:20:00
2	JB2252	JetBlue	200	Birmingham-Shuttlesworth International Airport	Cyril E. King Airport	16-NOV-22 07.00.00.000000000 AM	16-NOV-22 10.40.00.000000000 AM	03:40:00
3	AA7489	American Airlines	300	Henry E. Rohlsen Airport	Pago Pago International Airport	05-APR-23 06.30.00.000000000 AM	06-APR-23 06.15.00.000000000 PM	35:45:00
4	NK9384	Spirit Airlines	250	Sheridan County Airport	Kahului Airport	08-AUG-23 08.00.00.000000000 AM	09-AUG-23 10.35.00.000000000 PM	14:35:00
5	JK1257	Delta Airlines	300	John Wayne Airport	Palm Beach International Airport	14-JUL-23 05.00.00.000000000 AM	14-JUL-23 04.30.00.000000000 PM	11:30:00

C/D. The primary key of this relation is Flight_number with no foreign keys from other tables. This relation is in 3NF as the primary key Flight_number is the only dependency of all the other attributes in the entity, there are no composite or multivalued attributes, and none of the attributes transitively depend on the primary key. This relation is in BCNF as all the attributes other than Flight_Number are not keys.



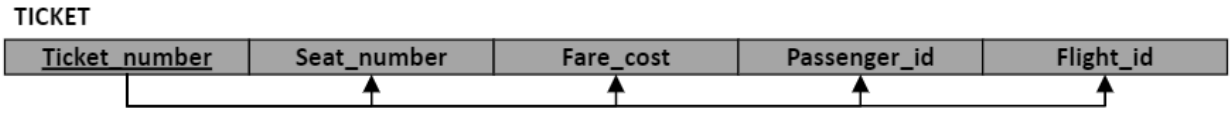
A. TICKET (Ticket_number, Fare_cost, Airline, Seat_number, Passenger_id, Flight_id)

- Primary key: Ticket_number
- Foreign key: Passenger_id, Flight_id

B. SELECT *
FROM TICKET;

	TICKET_NUMBER	SEAT_NUMBER	FARE_COST	PASSENGER_ID	FLIGHT_ID
1	98372049857281	34B	238.21	L12345678901234	UA1104
2	45758362801283	22A	216.23	C37482746371623	JB2252
3	75845610293613	46B	522.43	B19873658071334	AA7489
4	19238476193242	67D	310.21	D78123648271356	NK9384
5	21598724297644	26G	478.32	H21978346981237	JK1257

C/D. The primary key in this relation is Ticket_number and the foreign key in the table is Passenger_id and Flight_id. The ticket relation is in 3NF as all the non-prime attributes depend on the Ticket_number which is the primary key. None of the attributes are composite or multi-valued nor are transitively dependent on Ticket_number. This relation is in BCNF as it satisfies the requirements for 3NF and none of its attributes other than the prime attribute, Ticket_number, are keys.



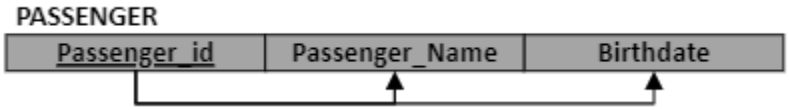
A. PASSENGER (Passenger_id, Passenger_name, Seat_number, Birthdate)

- Primary key: Passenger_id

B. SELECT *
FROM PASSENGER;

	PASSENGER_ID	PASSENGER_NAME	BIRTHDATE
1	L12345678901234	Dustin R La	07-DEC-01
2	C37482746371623	Natalia R Smith	22-AUG-02
3	B19873658071334	Jose M Cabrera	25-SEP-02
4	D78123648271356	Rishi Sakunala	05-JUL-02
5	H21978346981237	Gautham Nagrajan	12-MAR-02

C/D. The primary key in this relation is Passenger_id and there are no foreign keys in the relation. There are no non-prime attributes in this relation which depend on anything other than the primary key, Passenger_id, and none of the attributes are composite or multi-valued. There are no transitive dependencies in the relation either which classifies it as a 3NF relationship. This relationship satisfies BCNF as it satisfies 3NF and Passenger_name nor Birthdate are part of the key.



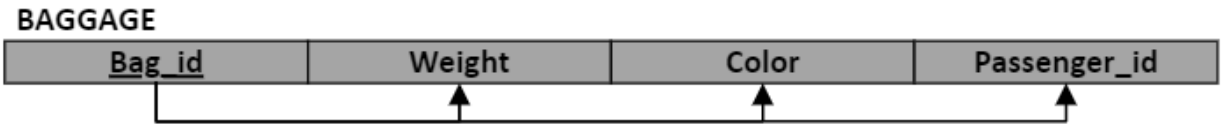
A. BAGGAGE (Bag_id, Weight, Color, Passenger_id)

- Primary key: Bag_id
- Foreign key: Passenger_id

B. SELECT *
FROM BAGGAGE;

	BAG_ID	WEIGHT	COLOR	PASSENGER_ID
1	AHGIU56739	27.23	Blue	L12345678901234
2	DSGER78134	25.72	Blue	C37482746371623
3	UIOSE83754	31.67	Green	B19873658071334
4	NIMDT56129	49.99	Red	D78123648271356
5	ALPID64812	20.82	Orange	H21978346981237

C/D. The primary key of the BAGGAGE relation is Bag_id with a foreign key of Passenger_id from PASSENGER which helps link the bags to their respective owners.The BAGGAGE relation is in 3NF as the Weight and Color attributes are fully dependent on the Bag_id with no transitive dependencies and there are no multivalued or composite attributes. The relation also satisfies the requirements for BCNF as the Weight, Color, and Passenger_id attributes are not keys in the BAGGAGE relationship.



Additional Insertions to help with Queries

INSERT INTO FLIGHTS(Flight_number, Airline, Capacity, Origin_airport, Destination_airport, Departure_Time, Arrival_Time, Flight_duration)
VALUES ('JK3216', 'Delta Airlines', '350', 'John Wayne Airport', 'Palm Beach International Airport', TO_DATE('2023-07-14 07:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_DATE('2023-07-14 12:30:00', 'YYYY-MM-DD HH24:MI:SS'), '05:30:00');

INSERT INTO FLIGHTS(Flight_number, Airline, Capacity, Origin_airport, Destination_airport, Departure_Time, Arrival_Time, Flight_duration)
VALUES ('AA1257', 'American Airlines', '250' ,'Newark Liberty International Airport', 'Kahului Airport', TO_DATE('2023-09-07 09:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_DATE('2023-09-07 14:10:00', 'YYYY-MM-DD HH24:MI:SS'), '05:10:00');

INSERT INTO FLIGHTS(Flight_number, Airline, Capacity, Origin_airport, Destination_airport, Departure_Time, Arrival_Time, Flight_duration)
VALUES ('JK5190', 'Delta Airlines', '340' ,'Pago Pago International Airport', 'Kahului Airport', TO_DATE('2023-09-07 10:50:00', 'YYYY-MM-DD HH24:MI:SS'), TO_DATE('2023-09-07 17:50:00', 'YYYY-MM-DD HH24:MI:SS'), '07:00:00');

INSERT INTO PASSENGER (Passenger_id, Passenger_name, Birthdate)
VALUES ('L10987654321023', 'Carlos Ulberg', TO_DATE('2003-11-02','yyyy-mm-dd'));

INSERT INTO PASSENGER (Passenger_id, Passenger_name, Birthdate)
VALUES ('K87364120983728', 'Alex Pereira', TO_DATE('1982-1-05','yyyy-mm-dd'));

INSERT INTO PASSENGER (Passenger_id, Passenger_name, Birthdate)
VALUES ('W92384019287364', 'Derek Brunson', TO_DATE('1928-3-28','yyyy-mm-dd'));

INSERT INTO PASSENGER (Passenger_id, Passenger_name, Birthdate)
VALUES ('P18317783817182', 'John Smith', TO_DATE('1978-6-28','yyyy-mm-dd'));

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)
VALUES ('34168724297312', '57A', '473.64', 'L10987654321023','JK3216');

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)
VALUES ('29132720995412', '90D', '300.56', 'W92384019287364','AA1257');

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)

VALUES ('71463910335113', '86B', '250.50', 'K87364120983728','JK5190');

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)
VALUES ('89979792779178', '21B', '437.20', 'P18317783817182','JK5190');

INSERT INTO TICKET(Ticket_number, Seat_number, Fare_cost, Passenger_id, Flight_id)
VALUES('21598724297644', '26G', '478.32', 'H21978346981237', 'JK1257');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('DOEPS93201', '25.61', 'Purple', 'W92384019287364');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('DWEPS93207', '10.32', 'Black', 'W92384019287364');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('ZOEPS34201', '15.91', 'Grey', 'W92384019287364');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('HJKSP92381', '20.12', 'Maroon', 'K87364120983728');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('ZFKSP92382', '16.58', 'Green', 'K87364120983728');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('WHYUZ64923', '25.20', 'Gray', 'L10987654321023');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('MPVWJ84601', '30.35', 'Blue', 'L10987654321023');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('FWPQZ53019', '27.81', 'Purple', 'L10987654321023');

INSERT INTO BAGGAGE(Bag_id, Weight, Color, Passenger_id)
VALUES ('PLQMZ80352', '52.13', 'Yellow', 'L10987654321023');

Tables after additional insertions

FLIGHTS:

	FLIGHT_NUMBER	AIRLINE	CAPACITY	ORIGIN_AIRPORT	DESTINATION_AIRPORT	DEPARTURE_TIME	ARRIVAL_TIME	FLIGHT_DURATION
1	JB2252	JetBlue	200	Birmingham-Shuttlesworth International Airport	Kahului Airport	16-NOV-22 07.00.00.000000000 AM	16-NOV-22 10.40.00.000000000 AM	03:40:00
2	AA7489	American Airlines	300	Henry E. Rohlsen Airport	Pago Pago International Airport	05-APR-23 06.30.00.000000000 AM	06-APR-23 06.15.00.000000000 PM	35:45:00
3	NK9384	Spirit Airlines	250	Sheridan County Airport	Kahului Airport	08-AUG-23 08.00.00.000000000 AM	09-AUG-23 10.35.00.000000000 PM	14:35:00
4	JK1257	Delta Airlines	300	John Wayne Airport	Palm Beach International Airport	14-JUL-23 05.00.00.000000000 AM	14-JUL-23 04.30.00.000000000 PM	11:30:00
5	JK3216	Delta Airlines	350	John Wayne Airport	Palm Beach International Airport	14-JUL-23 07.00.00.000000000 AM	14-JUL-23 12.30.00.000000000 PM	05:30:00
6	AA1257	American Airlines	250	Newark Liberty International Airport	Kahului Airport	07-SEP-23 09.00.00.000000000 AM	07-SEP-23 02.10.00.000000000 PM	05:10:00
7	JK5190	Delta Airlines	340	Pago Pago International Airport	Kahului Airport	07-SEP-23 10.50.00.000000000 AM	07-SEP-23 05.50.00.000000000 PM	07:00:00

PASSENGER:

	PASSENGER_ID	PASSENGER_NAME	BIRTHDATE
1	L12345678901234	Dustin R La	07-DEC-01
2	C37482746371623	Natalia R Smith	22-AUG-02
3	B19873658071334	Jose M Cabrera	25-SEP-02
4	D78123648271356	Hrishikesh Sakunala	05-JUL-02
5	L10987654321023	Carlos Ulberg	02-NOV-03
6	K87364120983728	Alex Pereira	05-JAN-82
7	W92384019287364	Derek Brunson	28-MAR-28
8	P18317783817182	John Smith	28-JUN-78

TICKET:

	TICKET_NUMBER	SEAT_NUMBER	FARE_COST	PASSENGER_ID	FLIGHT_ID
1	45758362801283	22A	216.23	C37482746371623	AA7489
2	75845610293613	46B	522.43	B19873658071334	AA7489
3	19238476193242	67D	310.21	D78123648271356	NK9384
4	34168724297312	57A	473.64	L10987654321023	JK3216
5	29132720995412	90D	300.56	W92384019287364	AA1257
6	71463910335113	86B	250.5	K87364120983728	JK5190
7	89979792779178	21B	437.2	P18317783817182	JK5190
8	90875983671986	23C	237.23	P18317783817182	NK9384

BAGGAGE:

	BAG_ID	WEIGHT	COLOR	PASSENGER_ID
1	AHGIU56739	27.23	Blue	L12345678901234
2	DSGER78134	25.72	Blue	D78123648271356
3	NIMDT56129	49.99	Red	D78123648271356
4	DOEPS93201	25.61	Purple	W92384019287364
5	DWEPS93207	10.32	Black	W92384019287364
6	ZOEPS34201	15.91	Grey	W92384019287364
7	HJKSP92381	20.12	Maroon	K87364120983728
8	ZFKSP92382	16.58	Green	K87364120983728
9	WHYUZ64923	25.2	Gray	L10987654321023
10	MPVWJ84601	30.35	Blue	L10987654321023
11	FWPQZ53019	27.81	Purple	L10987654321023
12	PLQMZ80352	52.13	Yellow	L10987654321023

4. SQL Queries

1. **GROUP BY Query**

Find the average fare cost of a given flight as well as the Airline. Select the Flight_id and airline and group all the tickets based on these attributes to find the average fare cost, and show the average price paid by the passengers of a given flight.

```
SELECT TICKET.Flight_id, AVG(TICKET.Fare_cost), FLIGHTS.Airline
FROM TICKET, FLIGHTS
WHERE TICKET.Flight_id = FLIGHTS.Flight_number
GROUP BY TICKET.Flight_id, FLIGHTS.Airline;
```

	FLIGHT_ID	AVG(TICKET.FARE_COST)	AIRLINE
1	JK3216	473.64	Delta Airlines
2	AA1257	300.56	American Airlines
3	AA7489	369.33	American Airlines
4	NK9384	273.72	Spirit Airlines
5	JK5190	343.85	Delta Airlines

2. **GROUP BY and HAVING Query**

Find the total bag weight of a passenger if they have at least two bags. Select the Passenger_name from Passenger and the Passenger_id and Weight from Baggage. If a Passenger_id in Passenger and Baggage matches and the Passenger has at least two bags, return the total weight of the bags and group by the Passenger_id and Passenger_name. Display the total bag weight of each passenger if they have more than two bags.

```
SELECT PASSENGER.Passenger_name, BAGGAGE.Passenger_id, SUM(BAGGAGE.Weight)
FROM PASSENGER, BAGGAGE
WHERE PASSENGER.Passenger_id = BAGGAGE.Passenger_id
GROUP BY BAGGAGE.Passenger_id, PASSENGER.Passenger_name
HAVING COUNT(*) >=2;
```

	PASSENGER_NAME	PASSENGER_ID	SUM(BAGGAGE.WEIGHT)
1	Hrishikesh Sakunala	D78123648271356	75.71
2	Alex Pereira	K87364120983728	36.7
3	Derek Brunson	W92384019287364	51.84
4	Carlos Ulberg	L10987654321023	135.49

3. ALL Query

Find the Passenger_name and Passenger_id of the Passenger(s) who purchased a ticket that is greater than all the costs individually of the tickets with Destination_airport of “Kahului Airport”. Select the Passenger_id and Fare_cost from Ticket and Passenger_name from Passenger. Match the Passengers and Tickets based on Passenger_id. Then, use a nested query to find all the fare cost of the tickets with destination of “Kahului Airport”. Use this tuple to compare with all the tickets. The tickets that have a higher Fare_cost than all of the tuple’s values from the nested query are returned with their Passenger_name and Passenger_id.

```
SELECT TICKET.Passenger_id, PASSENGER.Passenger_name, TICKET.Fare_Cost,
FLIGHTS.Destination_airport
FROM TICKET, PASSENGER, FLIGHTS
WHERE TICKET.Passenger_id = Passenger.Passenger_id
AND Fare_cost >= ALL (
   SELECT TICKET.Fare_Cost
   FROM TICKET, FLIGHTS
   WHERE TICKETS.Flight_id = FLIGHTS.Flight_number
   Destination_airport = 'Kahului Airport');
```

	PASSENGER_ID	PASSENGER_NAME	FARE_COST
1	B19873658071334	Jose M Cabrera	522.43
2	L10987654321023	Carlos Ulberg	473.64

4. **IN Query**

Return the total baggage weight being carried by any airline with flights originating from Pago Pago International Airport, Henry E. Rohlsen Airport, and Birmingham-Shuttlesworth International Airport. The query can be used so that airlines can compare how much baggage, in terms of weight, they have moved in total from certain locations. Select the sum of the baggage weight and airline where the Passenger_id in Baggage and Passenger match along with the Passenger_id in Passenger and Ticket. Then match the Passenger with their Ticket using Passenger_id. Also match the Flight_id from Ticket with the Flight_number from Flights and find the airline from the nested query that finds the airlines who have Destination_airports of Pago Pago International Airport, Henry E. Rohlsen Airport, and/or Birmingham-Shuttlesworth International Airport. Finally, group the baggage weights by the Airline.

```
SELECT SUM(BAGGAGE.Weight), FLIGHTS.Airline
FROM TICKET, PASSENGER, BAGGAGE, FLIGHTS
WHERE PASSENGER.Passenger_id = BAGGAGE.Passenger_id
AND PASSENGER.Passenger_id = TICKET.Passenger_id
AND TICKET.Flight_id = FLIGHTS.Flight_number
AND FLIGHTS.Airline IN(
   SELECT Airline
   FROM FLIGHTS
   WHERE Origin_airport = 'Pago Pago International Airport'
   OR Origin_airport = 'Henry E. Rohlsen Airport'
   OR Origin_airport = 'Birmingham-Shuttlesworth International Airport'
)
GROUP BY FLIGHTS.Airline;
```

	SUM(BAGGAGE.WEIGHT)	AIRLINE
1	51.84	American Airlines
2	172.19	Delta Airlines

5. Conclusion

a) Our group experienced a number of ups and downs throughout all phases of the database project. We found the hardest part to be designing an Entity-Relationship diagram in Phase 1. We ended up making it a lot more complicated than it needed to be. We had to rebuild it from the ground up in order to make it cleaner and more simple. Another part that we found difficult was writing queries and ensuring that they were logical and worked correctly. We found ourselves going through a lot of trial and error in order to ensure that the queries worked as intended.

However, there were also elements in the project that seemed to come to us quicker than others, such as normalization. We also were able to easily understand our schema after making major changes after Phase 1. We also all agreed that one of the easiest parts was creating the tables, as well as making insertions and deletions. Additionally, our group thought that functional relations were not too complex.

If we had to do this project all over again, we would certainly not make as complicated of an Entity-Relationship diagram as we did initially. It made things difficult to pinpoint or fully understand. It also would've greatly helped us succeed in the proceeding parts of the project. Something that we learned that we did not imagine we would, is how powerful joining queries together would be. We ended up getting very useful information out of our queries through the combination of tables, something that is surely applicable to real-world database usage.

b) We are quite proud of our IN query. We believe we did a great job incorporating all of our entities. We also believe we effectively used what we learned in the class to create this query.