



GROUP ASSIGNMENT

Technology Park Malaysia

CT042-3-1-IDB

INTRODUCTION TO DATABASES

**APU1F2006CS / APU1F2006CS(IS) / APU1F2006IT(IOT) /
APU1F2006CS(DF)**

HAND OUT DATE: 11 JANUARY 2021

HAND IN DATE: 8 MARCH 2021

WEIGHTAGE: 50%

No.	Name	TP Number
1.	LIM SHENG JIAN	TP056549
3.	LING JIN HAN	TP059609
4.	SIA DE LONG	TP060810

Table Of Content

Introduction	2
Database and Database Management System	4
Disadvantages of file-based system	4
Advantages of Database and DBMS	5
Business Rules	7
Structural Rule	7
Procedural Rule	8
Entity Relationship Diagram	9
Workload Matrix For Part 1	13
Database Schema	14
Normalisation	14
UNF	14
1NF	14
2NF	15
3NF	17
Relational Schema	19
Data Dictionary	20
Database Diagram	25
SQL-Data Manipulation Language (DML)	34
Student 1 (Ling Jin Han TP059609)	34
Student 2 (Lim Sheng Jian TP056549)	35
Student 3 (Sia De Long TP060810)	36
Workload Matrix For Part 2	39
References	40

Introduction

The core element of database technology in data management is "active database technology." Active database technology allows the database system to provide users with active service information, while completing various active requirements in a convenient and aspect. The database function is integrated in a unified database. So far, the application mechanism of active database technology is mainly through the existing technical rules, embedded data management methods in the database system to achieve the active response function, so the data management system using active database technology is also called "with rules Data management system". The integration of database technology and data management systems can make many responses to data management and database changes, by looking for support time, then judging whether the preset conditions are reached, and finally taking actions. Database technology and data management system integration application fields are relatively wide, such as network data management, office data management, warehouse data management, traffic data management, etc.

In the 1970s, "exception handling" was an important programming measure in program execution. In database technology, it is necessary to actively judge the consistency and completeness of data. Therefore, database technology itself has certain proactive characteristics, but this type of The functional characteristics are not yet perfect, the types of events and preset values are not perfect, and users cannot set up an active response mechanism according to their needs. Nowadays, database technology has become more mature, and the application of database technology in data management has a corresponding realization path: add monitors in data management systems → introduce triggers in database systems and SQL3 standards → set event trigger thresholds.

This report mainly analyzes database technology, and designs a central reservation database according to airline requirements for use by all airline reservation offices. We are talking about developing a database system that can be used to record various data such as flight number, total number of seats in business class of each airline, total number of seats in economy class, pilots have employee numbers, employee surnames, employee surnames, age, work Experience, position, flight time and salary. To become a senior pilot, he/she needs to fly 20,000 hours.

Flight attendants have their own ID, full name, position, salary, phone number and address. Address contains street, city, province or state, postal code and country/region. Flight reservations have a unique reservation number, reservation status, reservation date, flight number, departure date, departure time, date arrival time, arrival time, category indicators, total price, status indicators, outstanding balance, etc.

Database and Database Management System

Disadvantages of file-based system

Understanding the inherent problems of file-based systems can avoid duplication of these problems in the database system. In other words, one should learn from the mistakes made before. In fact, the use of the word "error" is somewhat derogatory and fails to give sufficient recognition to the useful services provided by the file system over the years. However, people have learned to deal with data in a better way. Want to convert a file-based system into a database system, although it is not necessary, it is helpful to understand how the file system works

Separation and data isolation are the disadvantages of data systems, because each program defines and manages its own data, so users of one program may not know the potentially useful data owned by other programs.

duplication of data is also one of the shortcomings of data-based systems. Since the same data is saved by different programs, repeated data will bring a lot of unnecessary shortcomings, wasted space, and the same item may have different values and/or different formats. (SINGH, 2019)

Program data dependence is also the shortcomings of file-based systems, that is because the file structure is defined in the program code, and the overall form of the program structure program, especially focusing on the various components of the program and the relationship between these components. The program is often referred to as well structured or poorly structured.

Incompatible file formats, which are based on the fact that the programs are written in different languages and therefore cannot easily access each other's files.

Proliferation of fixed queries/applications. As programs are written to meet specific functions, any new requirements require a new program, and data with various requirements must be supported by a specific system, so this is also one of its disadvantages. (Anon., 2020)

Advantages of Database and DBMS

Accurate data is important for an organisation to let it be reliable for analysis and planning. Therefore, a database management system (DBMS) should be a system which can fulfil this requirement. DBMS is a computing system that allows users to interact with database systems where there is data linked to each other with relationships (Gunjal, 2003). Hence, DBMS should best fit the case study system design because it also includes other advantages compared with file-based systems.

Control of Data Redundancy

First and foremost, DBMS is very effective in reducing data redundancy compared to file-based systems. This is because file-based systems usually consist of multiple files that are stored in many different locations in a system or across systems which might lead to couple copies of the same will exist in the system. DBMS will solve this issue by giving all the data dependency to eliminate all isolated files and link all data with relationship. This will be very useful for the case study because if there are many same file copies in the system then some of them might not sync with the real data, then the system cannot be trusted and relied on by the customers.

Sharing of Data

Besides that, another highlighted advantage for DBMS would be sharing of data which means all the users can share and access to the same data even from different locations while there will be levels of authorisation to access the data. Especially the case study about airline companies which have stations in six states, this advantage is very beneficial to the system. Therefore, all the stations can access the same data in the same system.

High Security

Moreover, DBMS is having improved security compared to file-based systems. Since DBMS is most likely run on a database server then usually the server might provide some sort of security method such as authentication, the database is hard to access directly without any third-party interface if the user is not the owner or having the identity to access it. In short, manipulation or stolen information rarely happens with all this security. This feature is very important to the case study as it protects customers' privacy and information from getting stolen or manipulated.

Data Consistency

Last but not least, data consistency also is one of the advantages for DBMS. it refers to when a data from one table changes then it will also affect another table of the same data. Although it is not affecting, the management system will not let you delete the primary data. This is because in DBMS every data has foreign key to refer to another table, data consistency will ensure that the primary data is there to refer to. It is very important that the customer record is always complete but not missing data.

Business Rules

Structural Rule

No.	Rule
1.	An airline must have airline ID, airline name, number of business seats and number of economy seats.
2.	A staff must have ID, first name, last name, salary, date of birth, experience and flying hours.
3.	A flight attendant must have an ID, full name, salary, zero or one phoneNum and zero or one address.
4.	Staff and flight attendants may pilot or ride one or more flights in different flight travel with different positions.
5.	A flight travel must have an ID, departure station, departure date and time, arrival date and time, destination and airline details.
6.	A flight travel must be piloted by two staff (captain and co-captain) and maximum of three flight attendants (leader, co-leader, assistant).
7.	A flight travel from the same airline must have the same number of business seats and number of economy seats.
8.	A customer must have an ID, first name , last name and mail address.
9.	Customers may have zero or more unique house phone numbers and zero or more unique email addresses. An email address only can be registered to one customer while a house phone number can be shared by another customer.
10.	All booking from customers must have a booking date, class indicator, status indicator, total price, amount-paid-so-far, flight details and customers details.

11.	A customer may book one or more flights.
12.	All the addresses recorded have street, city, province or state, postal code, and country.

Procedural Rule

No.	Rule
1.	Staff must have at least 20,000 flying hours to be a senior pilot.
2.	Booking status will be scratched if the customer had not paid in full 30 days prior to the departure.
3.	Money currency must be converted to local currency before recorded.
4.	Time must be converted to the local timezone before recording.

Entity Relationship Diagram

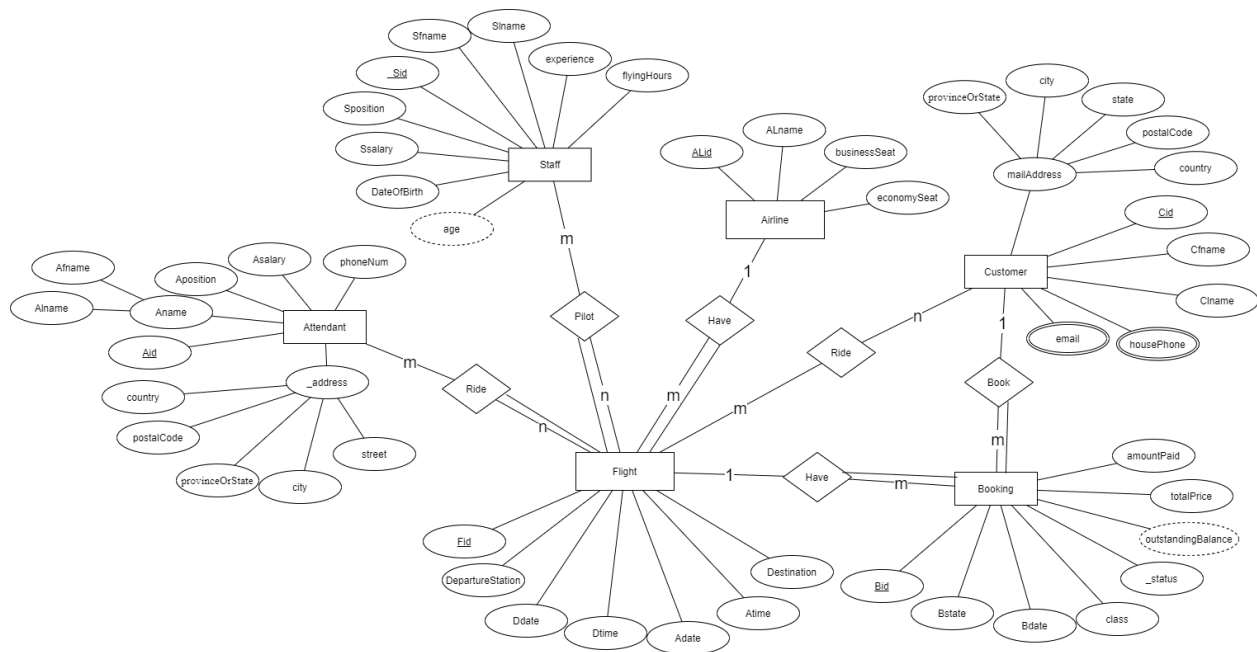


Figure 1: Complete ERD

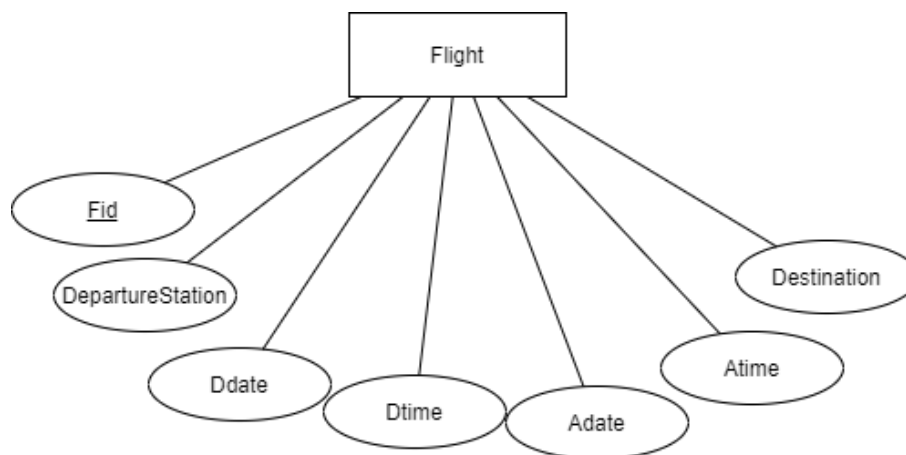


Figure 2: Entity Flight

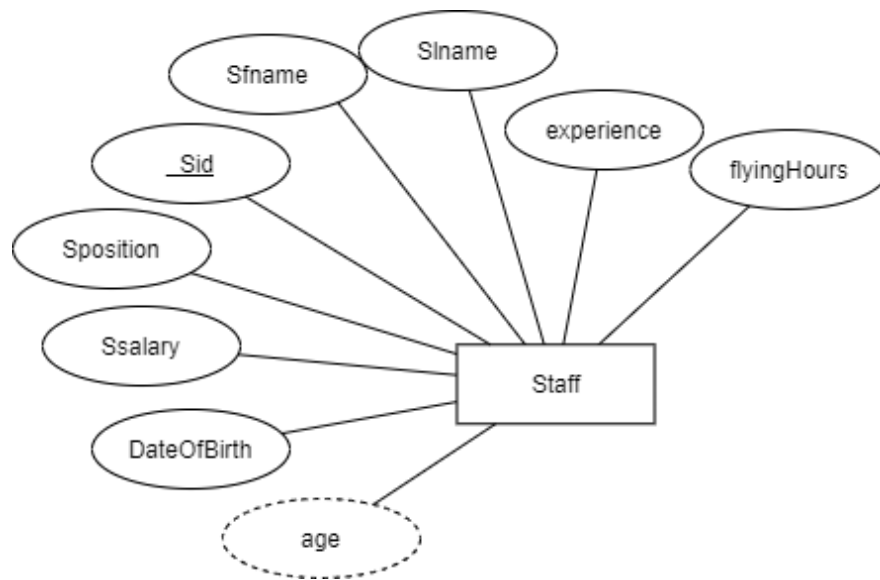


Figure 3: Entity Staff

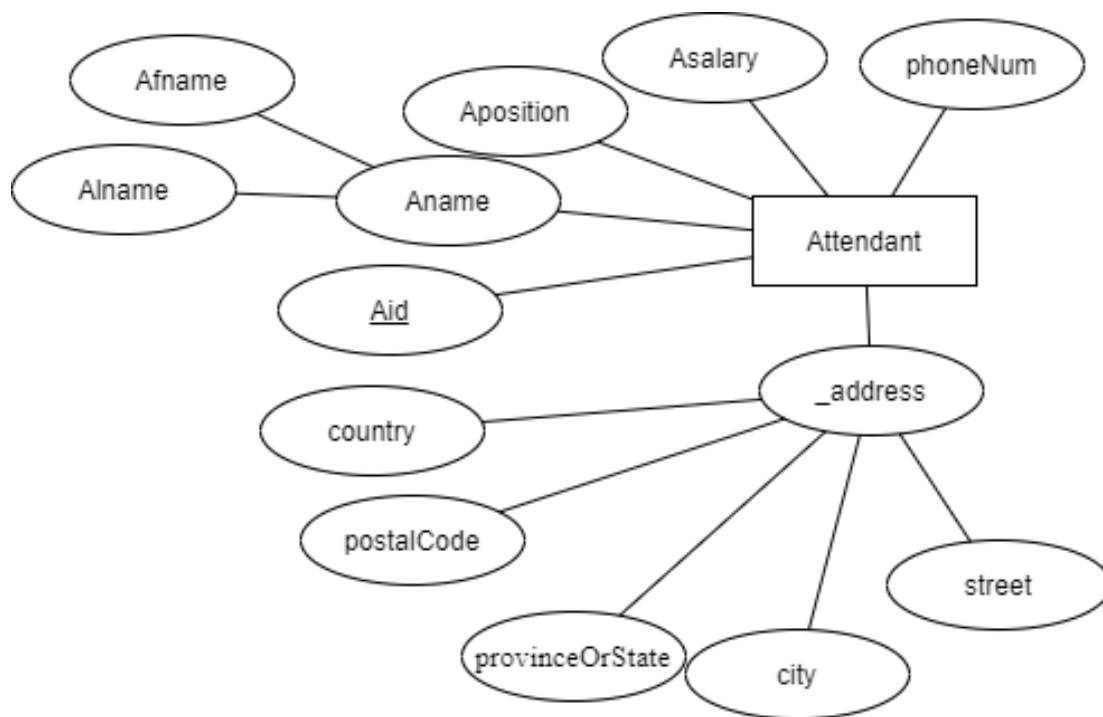


Figure 4: Entity Attendant

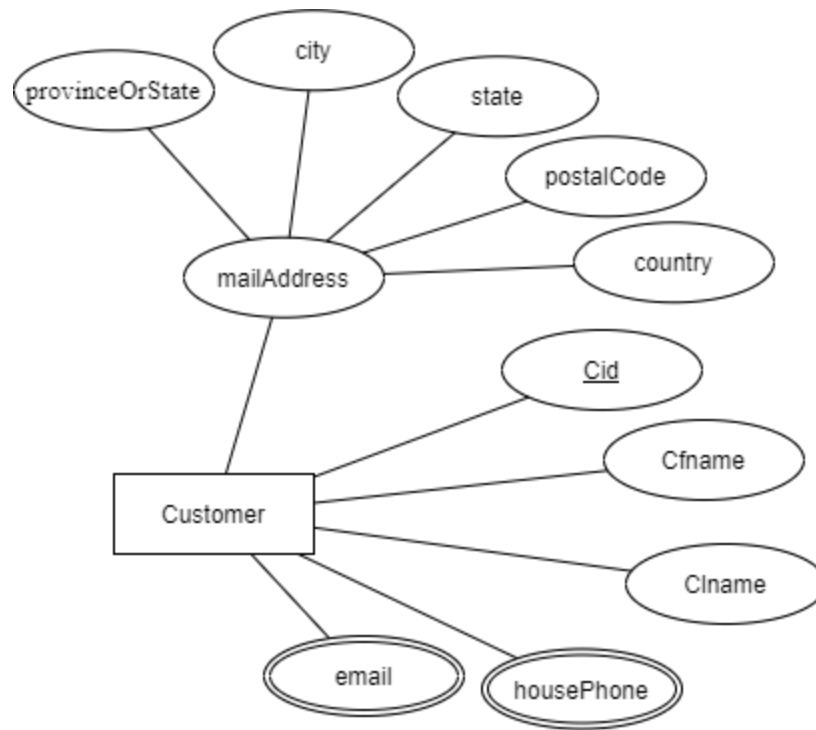


Figure 5: Entity Customer

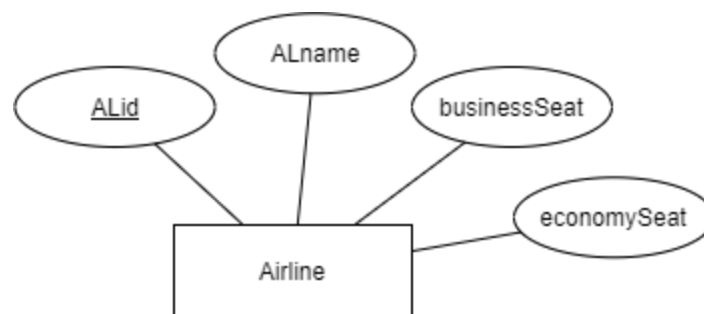


Figure 6: Entity Airline

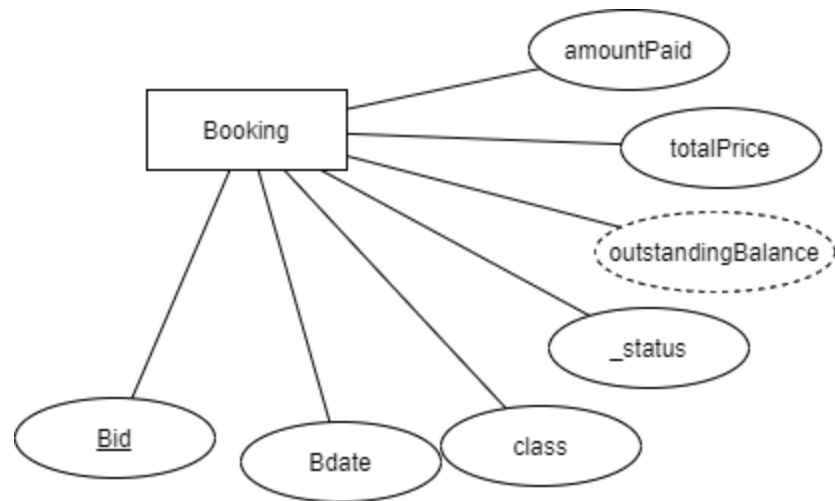


Figure 6: Entity Booking

Workload Matrix For Part 1

Title:	Student:	Sia De Long (TP060810)	Ling Jin Han (TP059609)	Lim Sheng Jian (TP056549)
Database and Database Management System		33%	34%	33%
Business Rule		34%	33%	33%
Entity Relationship Diagram		33%	33%	34%

Student	Sia De Long (TP060810)	Ling Jin Han (TP059609)	Lim Sheng Jian (TP056549)
Signature	<i>DeLong</i>	<i>JinHan</i>	<i>ShengJian</i>

Database Schema

Normalisation

UNF

This is the unnormalized form for customer booking data. It is not a bad approach since it records every detail of customer's booking including flight staff and attendant information but the column for staff and attendant is having multiple values since one flight travel must be piloted by two staff and having a maximum of three flight attendants. Therefore, the team needs to do the first normal form for this data table. The unnormalized form for booking table is shown below.

booking(Bid, Bdate, Fid, departureStation, Ddate, Dtime, Adate, Atime, destination, ALid, ALname, businessSeat, economySeat, class, _status, totalPrice, amountPaid, outstanding_Bal, Cid, Cfname, Clname, mailAddress, housePhone, email, _Sid, Sfname, Slname, dateOfBirth, age, experience, Sposition, flyingHours, Ssalary, Aid, Aname, Aposition, Asalary, phoneNum, _address)

1NF

In the first normal form, the team should get rid of repeating elements or groups of elements in other words atomicity. In order to achieve that each row of the customer's booking data must have an unique identifier or Primary Key. In this case, the first primary key would be Bid but another problem appeared because if the team only let the Bid unique some duplication data might still exist such as the same customer booked to another same flight. Hence, composite keys would be more suitable which means Bid, Cid and Fid all together become primary keys at the same time. Although the flight problem is solved, it will not let the system record multiple staff and attendants to the same flight, so, the final composite key should be Bid, Cid Fid, _Sid and Aid. Eventually, only a different combination for these five primary keys can exist so that the team ensures that duplication of records is not allowed. Lastly, due to the house phone and email of customers can be zero or more than one which is a multivalued attribute, so these two information will be separate to another table while the email will be the primary key in

customer_email table, this is to ensure that one email can only be registered to one customer then both hpNum and Cid will be the primary key in customer_housephone, this is to let one house phone can be shared by many customer while a same customer cannot have the same house phone. Finally, The final first normal form for booking table is shown below.

booking(Bid (PK), Bdate, Fid (PK), departureStation, Ddate, Dtime, Adate, Atime, destination, ALid, ALname, businessSeat, economySeat, class, _status, totalPrice, amountPaid, outstanding_Bal, Cid (PK), Cfname, Clname, mailAddress, _Sid (PK), Sfname, Slname, dateOfBirth, age, experience, Sposition, flyingHours, Ssalary, Aid (PK), Aname, Aposition, Asalary, phoneNum, _address)
customer_housephone(hpNum, Cid)
customer_email(email, Cid)

2NF

In the second normal form, the team is required to eliminate all partial dependencies on all tables that exist as a composite key. First and foremost, of course the attributes in the booking table will be checked.

Bdate is the date on when the booking of a flight from the customer is made. Therefore, it relies on Bid, Fid and Cid and it can still exist if without _Sid and Aid so Bdate fails second normal form.

departureStation is the location of the flight departure where the customer booked then of course it will rely on Bid, Fid and Cid but there is still partial dependency relation because it can still exist without Bid and Cid, so it should rely on Fid Fully, therefore departureStation fails second normal form. This is the same with **Ddate, Dtime, Adate, Atime** and **Destination** which is related to departure date and time, arrival date and time as well as the destination of a flight travel, obviously it will rely on Fid only and they all fail second normal form.

ALid is the ID for an airline of flight. Obviously it relies on Fid while all other attributes of the airline will also be the same situation and fail second normal form.

class is the class of seat which is chosen by the customer when booking for a flight travel, obviously it will rely on Bid, Fid and Cid but it can still exist if without **_Sid** and **Aid**. **_status**, **totalPrice** and **amountPaid** also have the same situation with class where **_status** is the status of the customer's booking followed by flight travel while total price and amount paid is just meant by the meaning of the word, hence they all fail the second normal form. Noticed that **outstanding_Bal** is not included in this situation this is because **outstanding_Bal** should not exist in the database since it can be easily calculated when **totalPrice** and **amountPaid** is recorded in the database, therefore it is removed from the table.

Cfname is the first name of a customer while **Clname** is the last name and **mailAddress** is the address of the customer where can mail documents to it. All these three are obviously about customer details, of course it will rely on Bid, Fid and Cid but there is still partial dependency relation because it can still exist without Bid and Fid, so it should rely on Cid Fully, hence they all fail second normal form.

Sfname is the first name of a staff while **Slname** is the last name, **dateOfBirth** is the date of birth for the staff, **Experience** is year of experience the staff in the pilot job, **flyingHours** is total hours the staff used on flight and **Ssalary** is the salary of the staff. Obviously, it is about the staff information on the flight, so it will rely on Fid and **_Sid** but there is still a partial dependency relation because it can still exist without Fid. Notice that **Sposition** is not included because every staff can have different positions in different flights, so it will rely on both Fid and **_Sid**. All of them can still exist if without other primary keys so they fail the second normal form. **age** of staff is removed because it can be calculated if we have the date of birth for that staff.

Aname is the full name of flight attendant, **Asalary** is salary of flight attendant, **phoneNum** is phone number of flight attendant and **_address** is address of flight attendant. Obviously, it is about the attendant information on the flight, so it will rely on Fid and Aid but there is still a partial dependency relation because it can still exist without Fid. Notice that **Aposition** is not included because it has the same situation with Sposition, so it will rely on both Fid and Aid. All of them can still exist if without other primary keys so they fail the second normal form.

After the team completes the first phase of the second normalisation, the team notices that every attribute fails the second normal form and has the primary they rely on. Therefore, the first phase of second normalisation will form the table shown below.

booking(Bid, Fid, Cid, Bstate, Bdate, class, _status, totalPrice, amountPaid)

flight(Fid, departureStation, Ddate, Dtime, Adate, Atime, destination, ALid, ALname, businessSeat, economySeat)

customer(Cid, Cfname, Clname, mailAddress, housePhone, email)

customer_housephone(hpNum, Cid)

customer_email(email, Cid)

staff(_Sid, Sfname, Slname, dateOfBirth, experience, flyingHours, Ssalary)

attendant(Aid, Aname, Asalary, phoneNum, _address)

staff_flight(_Sid, Fid, Sposition)

attendant_flight(Aid, Fid, Aposition)

In the second phase of second normalisation, we will still focus on the table which still has composite keys. In the booking table, all the attributes are checked and they cannot exist without any one of the three primary keys, so they passed the second normal form. The situation is the same when checking staff_flight and attendant_flight table, position cannot rely on only one of the primary keys. Therefore, all of the tables passed the second normal form but notice that ALid in the flight table actually does not rely on Fid, it can still exist if without any flight, so the team marked it as unknown and worried about them when they reached third normal form.

3NF

In the third normal form, the team will be eliminated dependencies on non-key attributes which are obviously **ALid**. Notice that ALid information is kept repeating when a new flight travel is recorded. Firstly, **ALname**, **businessSeat** and **economySeat** are confirmed rely on ALid and not Fid, therefore ALid and form an airline table with itself without any issues as shown below.

airline(ALid, ALname, businessSeat, economySeat)
flight(Fid, departureStation, Ddate, Dtime, Adate, Atime, destination)
staff(_Sid, Sfname, Slname, dateOfBirth, experience, flyingHours, Ssalary)
attendant(Aid, Aname, Asalary, phoneNum, _address)
booking(Bid, Fid, Cid, Bstate, Bdate, class, _status, totalPrice, amountPaid)
customer(Cid, Cfname, Clname, mailAddress, housePhone, email)
customer_housephone(hpNum, Cid)
customer_email(email, Cid)
attendant_Flight(Aid, Fid, Aposition)
staff_Flight(_Sid, Fid, Sposition)

However, the relation between airline and flight is now broken. Therefore, a foreign key of ALid should be added to the flight table to complete third normal form as shown below.

airline(ALid, ALname, businessSeat, economySeat)
flight(Fid, departureStation, Ddate, Dtime, Adate, Atime, destination, ALid)
staff(_Sid, Sfname, Slname, dateOfBirth, experience, flyingHours, Ssalary)
attendant(Aid, Aname, Asalary, phoneNum, _address)
booking(Bid, Fid, Cid, Bstate, Bdate, class, _status, totalPrice, amountPaid)
customer(Cid, Cfname, Clname, mailAddress, housePhone, email)
customer_housephone(hpNum, Cid)
customer_email(email, Cid)
attendant_Flight(Aid, Fid, Aposition)
staff_Flight(_Sid, Fid, Sposition)

Finally, normalisation until the third level is completed for the database.

Relational Schema

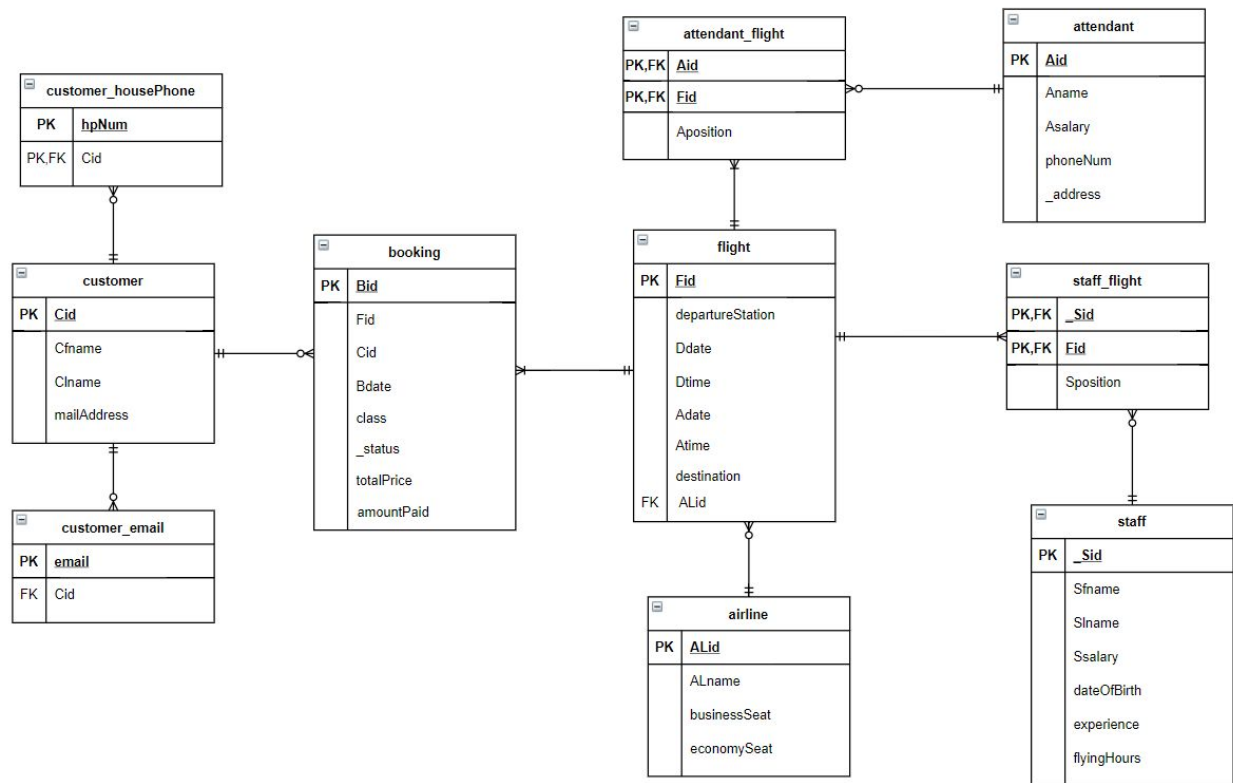


Figure 7: Relational Schema

Data Dictionary

Table: airline

Name	Data Type	Null	Constraint Name	Description
ALid	varchar(10)	-	flight_PK	Primary key (ALid)
ALname	varchar(50)	-	-	-
businessSeat	int	-	-	-
economySeat	int	-	-	-

Table: flight

Name	Data Type	Null	Constraint Name	Description
Fid	varchar(10)	-	flight_PK	Primary key (Fid)
			CHK_Fid	CHECK (Fid LIKE 'F%')
departureStation	varchar(32)	-	-	-
Ddate	date	-	-	-
Dtime	time(0)	-	-	-
Adate	date	-	-	-
Atime	time(0)	-	-	-
destination	varchar(32)	-	-	-
ALid	varchar(10)	-	flight_FK	Foreign key (ALid) REFERENCES airline(ALid)

Table: staff

Name	Data Type	Null	Constraint Name	Description
_Sid	varchar(10)	-	staff_PK	Primary key (_Sid)
			CHK_Sid	CHECK (_Sid LIKE 'S%')
Sfname	varchar(20)	-	-	-
Slname	varchar(20)	-	-	-
Ssalary	float(2)	-	-	-
dateOfBirth	date	-	-	-
experience	float(1)	-	-	-
flyingHours	float(1)	-	-	-

Table: attendant

Name	Data Type	Null	Constraint Name	Description
Aid	varchar(10)	-	attendant_PK	Primary key (Aid)
			CHK_Aid	CHECK (Aid LIKE 'A%')
Aname	varchar(50)	-	-	-
Asalary	float(2)	-	-	-
phoneNum	varchar(12)	✓	CHK_phoneNum	(phoneNum LIKE '____-%')
_address	varchar(100)	✓	-	-

Table: customer

Name	Data Type	Null	Constraint Name	Description
Cid	varchar(10)	-	customer_PK	Primary key (Cid)
			CHK_Cid	CHECK (Cid LIKE 'C%')
Cfname	varchar(20)	-	-	-
Clname	varchar(20)	-	-	-
mailAddress	varchar(100)	-	-	-

Table: booking

Name	Data Type	Null	Constraint Name	Description
Bid	int IDENTITY (1,1)	-	booking_PK	Primary key (Bid, Fid, Cid)
Fid	varchar(10)	-	booking_flight_FK	Foreign key (Fid) REFERENCES flight(Fid)
Cid	varchar(10)	-	booking_customer_FK	Foreign key (Cid) REFERENCES customer(Cid)
Bdate	date	-		
class	varchar(8) DEFAULT 'economy'	-	CHK_class	CHECK (class = 'business' OR class = 'economy')
_status	varchar(9) DEFAULT 'booked'	-	CHK_status	CHECK (_status = 'booked' OR _status = 'canceled' OR _status = 'scratched')
totalPrice	float(2)	-	-	-
amountPaid	float(2)	-	-	-

Table: Customer_housephone

Name	Data Type	Null	Constraint Name	Description
hpNum	varchar(11)	-	housePhone_Pk	Primary key (hpNum, Cid),
			CHK_housePhone	CHECK (hpNum LIKE '___-%')
Cid	varchar(10)	-	housePhone_FK	Foreign key (Cid) REFERENCES customer(Cid)

Table: Customer_email

Name	Data Type	Null	Constraint Name	Description
email	varchar(50)	-	email_Pk	Primary key (email)
			CHK_email	CHECK (email LIKE '%_@__%.__%')
Cid	varchar(10)	-	email_FK	Foreign key (Cid) REFERENCES customer(Cid)

Table: Attendant_Flight

Name	Data Type	Null	Constraint Name	Description
Aid	varchar(10)	-	attendant_flight_FK	Foreign key (Aid) REFERENCES attendant(Aid)
			attendant_flight_PK	Primary key (Aid, Fid)
Fid	varchar(10)	-	flight_attendant_FK	Foreign key (Fid) REFERENCES flight(Fid)
Aposition	varchar(9) DEFAULT 'assistant'	-	CHK_Aposition	CHECK (Aposition = 'leader' OR Aposition = 'co-leader' OR Aposition = 'assistant')

Table: Staff_Flight

Name	Data Type	Null	Constraint Name	Description
_Sid	varchar(10)	-	staff_flight_FK	Foreign key (_Sid) REFERENCES staff(_Sid)
			staff_flight_PK	Primary key (_Sid, Fid)
Fid	varchar(10)	-	flight_staff_FK	Foreign key (Fid) REFERENCES flight(Fid)
Sposition	varchar(10) DEFAULT 'co-captain'	-	CHK_Sposition	CHECK (Sposition = 'captain' OR Sposition = 'co-captain')

Database Diagram

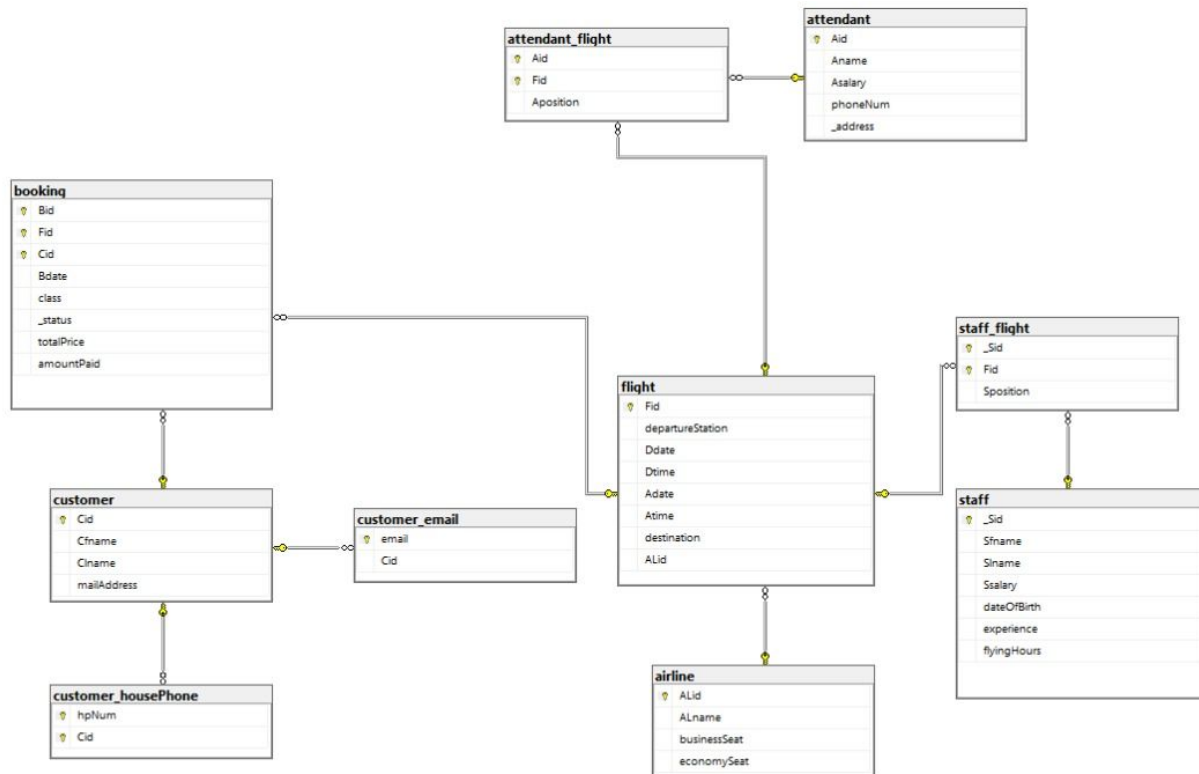


Figure 8: Database Diagram

Data Definition Language (DDL)

```
--CREATE
-- Create Database
CREATE Database Air_Reservation;

-- use DataBase
use Air_Reservation;
```

Figure 9: Create and Use database Air_Reservation

```
-- Create Table airline
CREATE TABLE airline(
  ALid varchar(10) NOT NULL,
  ALname varchar(50) NOT NULL,
  businessSeat int NOT NULL,
  economySeat int NOT NULL,
  CONSTRAINT airline_PK Primary key (ALid) );
```

Figure 10: Table airline

```
-- Create Table flight
CREATE TABLE flight(
  Fid varchar(10) NOT NULL,
  departureStation varchar(32) NOT NULL,
  Ddate date NOT NULL,
  Dtime time(0) NOT NULL,
  Adate date NOT NULL,
  Atime time(0) NOT NULL,
  destination varchar(32) NOT NULL,
  ALid varchar(10) NOT NULL,
  CONSTRAINT flight_PK Primary key (Fid),
  CONSTRAINT CHK_date CHECK (Adate >= Ddate),
  CONSTRAINT CHK_Fid CHECK (Fid LIKE 'F%'),
  CONSTRAINT flight_FK Foreign key (ALid) REFERENCES airline(ALid) );
```

Figure 11: Table flight

```
-- Create Table staff
CREATE TABLE staff(
  _Sid varchar(10) NOT NULL,
  Sfname varchar(20) NOT NULL,
  Slname varchar(20) NOT NULL,
  Ssalary float(2) NOT NULL,
  dateOfBirth date NOT NULL,
  experience float(1) NOT NULL,
  flyingHours float(1) NOT NULL,
  CONSTRAINT staff_PK Primary key (_Sid),
  CONSTRAINT CHK_Sid CHECK (_Sid LIKE 'S%') );
```

Figure 12: Table staff

```
-- Create Table attendant
CREATE TABLE attendant(
Aid varchar(10) NOT NULL,
Aname varchar(50) NOT NULL,
Asalary float(2) NOT NULL,
phoneNum varchar(12) NULL,
_address varchar(100) NULL,
CONSTRAINT attendant_PK Primary key (Aid),
CONSTRAINT CHK_phoneNum CHECK (phoneNum LIKE '____-%'),
CONSTRAINT CHK_Aid CHECK (Aid LIKE 'A%') );
```

Figure 13: Table attendant

```
-- Create Table customer
CREATE TABLE customer(
Cid varchar(10) NOT NULL,
Cfname varchar(20) NOT NULL,
Clname varchar(20) NOT NULL,
mailAddress varchar(100) NULL,
CONSTRAINT customer_PK Primary key (Cid),
CONSTRAINT CHK_Cid CHECK (Cid LIKE 'C%') );
```

Figure 14: Table customer

```
-- Create Table housePhone
CREATE TABLE customer_housePhone(
hpNum varchar(11) NOT NULL,
Cid varchar(10) NOT NULL,
CONSTRAINT housePhone_Pk Primary key (hpNum, Cid),
CONSTRAINT CHK_housePhone CHECK (hpNum LIKE '____-%'),
CONSTRAINT housePhone_FK Foreign key (Cid) REFERENCES customer(Cid) );
```

Figure 15: Table housePhone

```
--Create Table email
CREATE TABLE customer_email(
  email varchar(50) NOT NULL,
  Cid varchar(10) NOT NULL,
  CONSTRAINT email_Pk Primary key (email),
  CONSTRAINT CHK_email CHECK (email LIKE '%_@_%._%'),
  CONSTRAINT email_FK Foreign key (Cid) REFERENCES customer(Cid) );
```

Figure 16: Table email

```
-- Create Table booking
CREATE TABLE booking(
  Bid int IDENTITY (1,1) NOT NULL,
  Fid varchar(10) NOT NULL,
  Cid varchar(10) NOT NULL,
  Bdate date NOT NULL,
  class varchar(8) NOT NULL DEFAULT 'economy',
  _status varchar(9) NOT NULL DEFAULT 'booked',
  totalPrice float(2) NOT NULL,
  amountPaid float(2) NOT NULL,
  CONSTRAINT booking_PK Primary key (Bid, Fid, Cid),
  CONSTRAINT CHK_class CHECK (class = 'business' OR class = 'economy'),
  CONSTRAINT CHK_status CHECK (_status = 'booked' OR _status = 'canceled' OR _status = 'scratched'),
  CONSTRAINT booking_customer_FK Foreign key (Cid) REFERENCES customer(Cid),
  CONSTRAINT booking_flight_FK Foreign key (Fid) REFERENCES flight(Fid) );
```

Figure 17: booking

```
-- Create Table staff_flight
CREATE TABLE staff_flight(
  _Sid varchar(10) NOT NULL,
  Fid varchar(10) NOT NULL,
  Sposition varchar(10) NOT NULL DEFAULT 'co-captain',
  CONSTRAINT staff_flight_PK Primary key (_Sid, Fid),
  CONSTRAINT CHK_Sposition CHECK (Sposition = 'captain' OR Sposition = 'co-captain'),
  CONSTRAINT staff_flight_FK Foreign key (_Sid) REFERENCES staff(_Sid),
  CONSTRAINT flight_staff_FK Foreign key (Fid) REFERENCES flight(Fid) );
```

Figure 18: staff_flight


```
-- Create Table attendant_flight
CREATE TABLE attendant_flight(
Aid varchar(10) NOT NULL,
Fid varchar(10) NOT NULL,
Aposition varchar(9) NOT NULL DEFAULT 'assistant',
CONSTRAINT attendant_flight_PK Primary key (Aid, Fid),
CONSTRAINT CHK_Aposition CHECK (Aposition = 'leader' OR Aposition = 'co-leader' OR Aposition = 'assistant'),
CONSTRAINT attendant_flight_FK Foreign key (Aid) REFERENCES attendant(Aid),
CONSTRAINT flight_attendant_FK Foreign key (Fid) REFERENCES flight(Fid) );
```

Figure 19: Table attendant_flight

Table Data

	ALid	ALname	businessSeat	economySeat
1	CA8760	Core Airways	30	20
2	EA1709	Echo Airline	10	20
3	PA2098	Peak Airways	20	60
4	SA1865	Spark Airways	15	30

Figure 20: airline data

	Fid	departureStation	Ddate	Ddate	Adate	Dtime	destination	ALid
1	F1	Perak	2021-03-02	19:30	2021-03-03	08:00	Sabah	EA1709
2	F10	Wilayah Persekutuan Kuala Lumpur	2021-08-12	15:40	2021-08-12	18:00	Pahang	PA2098
3	F2	Pahang	2021-03-02	19:30	2021-03-03	08:00	Wilayah Persekutuan Kuala Lumpur	PA2098
4	F3	Wilayah Persekutuan Kuala Lumpur	2021-05-30	10:00	2021-05-30	11:00	Pahang	CA8760
5	F4	Perak	2021-06-20	11:00	2021-06-20	12:30	Wilayah Persekutuan Kuala Lumpur	SA1865
6	F5	Negeri Sembilan	2021-06-21	14:00	2021-06-21	15:00	Sarawak	CA8760
7	F6	Sarawak	2021-06-20	11:00	2021-06-20	12:30	Wilayah Persekutuan Kuala Lumpur	EA1709
8	F7	Pahang	2021-07-15	08:30	2021-07-15	09:50	Perak	EA1709
9	F8	Sabah	2021-07-02	12:30	2021-07-02	15:00	Sarawak	SA1865
10	F9	Sarawak	2021-08-02	16:30	2021-08-02	18:00	Perak	EA1709

Figure 21: flight data

	Sid	Sfname	Slname	Ssalary	dateOfBirth	experience	flyingHours
1	S1	Leon	Lim	5000	1974-04-29	3	25000
2	S10	Kenneth	Cooke	7100	1975-05-01	5	20000
3	S2	Mary	Jane	7000	1997-09-09	5	35000
4	S3	John	Weak	4000	1991-03-04	1	10000
5	S4	Mikael	Tinnason	4500	1971-07-20	2	20000
6	S5	Franky	Owen	7800	1982-08-03	6	40000
7	S6	Honor	Knowles	5700	1978-02-09	10	60000
8	S7	Chand...	Pittman	6800	1983-09-11	8	30000
9	S8	Aayat	Mcmanus	9700	1975-10-18	11	60000
10	S9	Mitchel	Holloway	4100	1980-04-21	9	75000

Figure 22: staff data

	Aid	Aname	Asalary	phoneNum	_address
1	A1	Denise Stephens	3500	012-6543210	Jalan 7/40, Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, 52000, Malaysia
2	A10	Poppy Gilmore	3300	016-2587064	Lorong Jaya 9, Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, 56000 Malaysia
3	A2	Jordon Tanner	3000	013-7765432	Jalan Chin Hwa, Ipoh, Perak, 30250, Malaysia
4	A3	Ayden Bloggs	2500	011-5551111	Simpang 32-38-15, Bandar Seri Begawan, Brunei, BB4513, Brunei
5	A4	Deon Mac	2800	017-9994567	Jalan Lestari 10, Raub, Pahang, 27600, Malaysia
6	A5	Alex Leal	3200	016-1247568	Lorong Tenggiling 1, Kota Kinabalu, Sabah, 88300, Malaysia
7	A6	Chiara Waters	3600	013-6783489	Jalan Angsana, Kuching, Sarawak, 93150, Malaysia
8	A7	Patricia Wu	2300	016-9518758	Lorong B2, Kuching, Sarawak, 93350, Malaysia
9	A8	Bertie Hume	2750	012-3021875	Jalan Pine 6, Jerantut, Pahang, 27000, Malaysia
10	A9	Dominic Tomli...	3100	016-9854726	Jalan Seri Mawar 17, Seremban, Negeri Sembilan, 70450, Malaysia

Figure 23: attendant data

	Cid	Cfname	Clname	mailAddress
1	C1	Shane	Wok	Jalan Selasih, Tanah Merah, Kelantan, 17500, Malaysia
2	C10	Maud	Tierney	Jalan Desa Melor 1, Seremban, Negeri Sembilan, 70450, Malaysia
3	C2	Jane	Wok	Lrg Pakda Wail, Kota Bharu, Kelantan, 15200, Malaysia
4	C3	Anne	Dang	Jalan Tj 7, Temerloh, Pahang, 28000, Malaysia
5	C4	Donnie	Yan	Hala Lapangan Perdana 5, Ipoh, Perak, 31650, Malaysia
6	C5	Edward	Send	Lorong Nikmat 11, Kuching, Sarawak, 58200, Malaysia
7	C6	Rihanna	Sinclair	Lorong Seladang, Kota Kinabalu, Sabah, 88300, Malaysia
8	C7	Ameera	Mcdaniel	Simpang 77-19-21, Bandar Seri Begawan, Brunei, BB4513, Brunei
9	C8	Youssef	Nielsen	Lorong Seladang, Kota Kinabalu, Sabah, 88300, Malaysia
10	C9	Ema	Lowe	Dataran Perpaduan 13, Ulu Kinta, Perak, 31150, Malaysia

Figure 24: customer data

	hpNum	Cid
1	03-54872504	C10
2	03-57139878	C2
3	03-66874759	C6
4	03-66874759	C8
5	09-05415640	C10
6	09-18569478	C4
7	09-23157654	C6
8	09-24756864	C2
9	09-34567813	C1
10	09-68894759	C3

Figure 25: housePhone data

	email	Cid
1	anne@gmail.com	C3
2	ed879@gmail.com	C5
3	edwardsend@gm...	C5
4	edwsnd34@gmai...	C5
5	maud654@gmail...	C10
6	niel@gmail.com	C8
7	shn@gmail.com	C1
8	swok564@gmail....	C1
9	tier8@gmail.com	C10
10	yan@gmail.com	C4

Figure 26: email data

	Bid	Fid	Cid	Bdate	class	_status	totalPrice	amountPaid
1	1	F1	C1	2021-01-01	business	booked	150	50
2	2	F1	C2	2021-03-01	economy	booked	150	100
3	3	F2	C3	2021-03-27	business	booked	75	25
4	4	F2	C6	2021-03-27	economy	scratched	50	0
5	5	F3	C1	2021-04-15	economy	booked	100	100
6	6	F3	C10	2021-04-15	economy	booked	100	75
7	7	F4	C4	2021-04-20	business	canceled	90	0
8	8	F5	C7	2021-05-18	economy	booked	100	0
9	9	F6	C5	2021-05-20	business	canceled	115	0
10	10	F7	C1	2021-06-18	economy	booked	70	50
11	11	F8	C2	2021-06-18	economy	booked	35	35
12	12	F9	C2	2021-06-20	business	canceled	150	70
13	13	F10	C4	2021-07-18	economy	booked	80	50

Figure 27: booking data

	Sid	Fid	Sposition
1	S1	F10	captain
2	S1	F4	co-captain
3	S1	F8	captain
4	S10	F6	co-captain
5	S10	F7	captain
6	S10	F9	captain
7	S2	F5	captain
8	S3	F2	co-captain
9	S4	F2	captain
10	S4	F3	captain
11	S5	F1	captain
12	S6	F10	co-captain
13	S6	F7	co-captain
14	S7	F4	captain
15	S7	F8	co-captain
16	S7	F9	co-captain
17	S8	F1	co-captain
18	S8	F5	co-captain
19	S8	F6	captain
20	S9	F3	co-captain

Figure 28: staff_flight data

	Aid	Fid	Aposition
1	A1	F1	leader
2	A1	F6	assistant
3	A10	F2	assistant
4	A10	F9	leader
5	A2	F2	leader
6	A3	F1	assistant
7	A3	F4	leader
8	A4	F3	leader
9	A5	F1	co-leader
10	A5	F8	leader
11	A6	F3	assistant
12	A6	F7	leader
13	A7	F2	co-leader
14	A7	F6	leader
15	A8	F5	leader
16	A9	F10	leader

Figure 29: attendant_flight

SQL-Data Manipulation Language (DML)

Student 1 (Ling Jin Han TP059609)

```
--1)Display all customers whose first name ends with consonant
SELECT * FROM customer
WHERE Cfname like '%d' AND mailAddress LIKE '%Sarawak%';
```

Figure 30: DML question 1

	Cid	Cfname	Clname	mailAddress
1	C5	Edward	Send	Lorong Nikmat 11, Kuching, Sarawak, 58200, Malaysia

Figure 31: result 1

```
--2)Display the average of salary for pilots who have reached
SELECT AVG(Ssalary) FROM staff
WHERE flyingHours>=20000;
```

Figure 32: DML question 2

	(No column name)
1	6411.111111111111

Figure 33: result 2

```
--3)List the first name, last name, age, and experience of pilots who have
SELECT DISTINCT S.Sfname, S.Slname, S.dateOfBirth, S.experience
FROM staff AS S, staff_flight AS SF, flight AS F, airline AS A
WHERE S._Sid=SF._Sid AND F.Fid=SF.Fid AND A.ALid=F.ALid AND A.ALname='Sf';
```

Figure 34: DML question 3

	Sfname	Slname	dateOfBirth	experience
1	Chandler	Pittman	1983-09-11	8
2	Leon	Lim	1974-04-29	3

Figure 35: result 3

```
--4)List the booking's date, flight number, and the client's last
SELECT B.Bdate, F.ALid, C.Clname
FROM booking AS B, flight AS F, customer AS C
WHERE B.totalPrice=B.amountPaid AND B.Cid=C.Cid AND B.Fid=F.Fid;
```

Figure 36: DML question 4

	Bdate	ALid	Clname
1	2021-04-15	CA8760	Wok
2	2021-06-18	SA1865	Wok

Figure 35: result 4

Student 2 (Lim Sheng Jian TP056549)

```
--1) List the name and the position of flight attendants whose salary is ne:
SELECT a.Aname, af.Aposition, af.Fid
FROM attendant a, attendant_flight af
WHERE a.Aid = af.Aid AND NOT a.Asalary = 2800 AND NOT a.Asalary = 3500
```

Figure 37: DML question 1

	Aname	Aposition	Fid
1	Poppy Gilmore	assistant	F2
2	Poppy Gilmore	leader	F9
3	Jordon Tanner	leader	F2
4	Ayden Bloggs	assistant	F1
5	Ayden Bloggs	leader	F4
6	Alex Leal	co-leader	F1
7	Alex Leal	leader	F8
8	Chiara Waters	assistant	F3
9	Chiara Waters	leader	F7
10	Patricia Wu	co-leader	F2
11	Patricia Wu	leader	F6
12	Bertie Hume	leader	F5
13	Dominic Tom...	leader	F10

Figure 38: result 1

```
--2) Display the customer's first name and last name who have n
SELECT DISTINCT c.Cfname, c.Clname
FROM customer c, booking b
WHERE c.Cid = b.Cid;
```

Figure 39: DML question 2

	Cfname	Cname
1	Anne	Dang
2	Ameera	Mcdaniel
3	Edward	Send
4	Rihanna	Sinclair
5	Maud	Tierney
6	Jane	Wok
7	Shane	Wok
8	Donnie	Yan

Figure 40: result 2

```
--3) List flight attendant's full name and position who have worked in the same flight as a ;
SELECT a.Aname, af.Aposition, af.Fid
FROM attendant a, staff s, staff_flight sf, attendant_flight af
WHERE a.Aid = af.Aid AND sf.Fid = af.Fid AND s.Sid = sf.Sid AND s.Sfname = 'Mikael' AND s.S
```

Figure 41: DML question 3

	Aname	Aposition	Fid
1	Poppy Gilmore	assistant	F2
2	Jordon Tanner	leader	F2
3	Deon Mac	leader	F3
4	Chiara Waters	assistant	F3
5	Patricia Wu	co-leader	F2

Figure 42: result 3

```
--4) List all customers who did not live in any of the airline offices located. Please display the customer's first name, last name, and email address.
SELECT Cfname, Cname, mailAddress
FROM customer
WHERE NOT mailAddress LIKE '%Perak%'
AND NOT mailAddress LIKE '%Negeri Sembilan%'
AND NOT mailAddress LIKE '%Pahang%'
AND NOT mailAddress LIKE '%Sabah%'
AND NOT mailAddress LIKE '%Sarawak%'
AND NOT mailAddress LIKE '%Wilayah Persekutuan Kuala Lumpur%';
```

Figure 43: DML question 4

	Cfname	Cname	mailAddress
1	Shane	Wok	Jalan Selasih, Tanah Merah, Kelantan, 17500, Malaysia
2	Jane	Wok	Lrg Pakda Wail, Kota Bharu, Kelantan, 15200, Malaysia
3	Ameera	Mcdaniel	Simpang 77-19-21, Bandar Seri Begawan, Brunei, BB4513, Brunei

Figure 44: result 4

Student 3 (Sia De Long TP060810)

```
-- Find the highest salary of flight attendants
SELECT MAX(Asalary) AS Highest_Salary
FROM attendant;
```

Figure 45: DML question 1

	Highest_Salary
1	3600

Figure 46: result 1

```
-- Display cancelled booknig
SELECT customer.Cfname, booking.Bid, booking.Bdate, airline.ALname, flight.Ddate, CONVERT(varchar(5), flight.Dtime) AS Dtime, booking._status
FROM customer
INNER JOIN booking ON customer.Cid = booking.Cid
INNER JOIN flight ON booking.Fid = flight.Fid
INNER JOIN airline ON flight.ALid = airline.ALid
WHERE _status = 'canceled';
```

Figure 47: DML question 2

	Cfname	Bid	Bdate	ALname	Ddate	Dtime	_status
1	Donnie	46	2021-04-20	Spark Airways	2021-06-20	11:00	canceled
2	Edward	48	2021-05-20	Echo Airline	2021-06-20	11:00	canceled
3	Jane	51	2021-06-20	Echo Airline	2021-08-02	16:30	canceled

Figure 48: result 2

```
-- List customers who did not place any booking
SELECT Customer.Cfname, Customer.Clname
FROM customer LEFT JOIN booking
ON customer.Cid = booking.Cid
WHERE Bid IS NULL
ORDER BY customer.Cid DESC;
```

Figure 49: DML question 3

	Cfname	Clname
1	Ema	Lowe
2	Youssef	Nielsen

Figure 50: result 3

```
-- List customers who still have outstanding balance
SELECT DISTINCT Customer.Cfname, Customer.Clname, SUM(totalPrice - amountPaid) AS outstanding_Bal
FROM customer INNER JOIN booking
ON customer.Cid = booking.Cid
WHERE (SELECT totalPrice - amountPaid) > 0
GROUP BY Cfname, Clname
ORDER BY outstanding_Bal ASC
```

Figure 51: DML question 4

	Cfname	Clname	oustanding_Bal
1	Maud	Tierney	25
2	Anne	Dang	50
3	Rihanna	Sinclair	50
4	Ameera	Mcdaniel	100
5	Edward	Send	115
6	Shane	Wok	120
7	Donnie	Yan	120
8	Jane	Wok	130

Figure 52: result 4

Workload Matrix For Part 2

Title:	Student:	Sia De Long (TP060810)	Ling Jin Han (TP059609)	Lim Sheng Jian (TP056549)
Database Schema		33%	34%	33%
SQL - Data Definition Language (DDL)		34%	33%	33%
SQL - Data Manipulation Language (DML)		33%	33%	34%

Student	Sia De Long (TP060810)	Ling Jin Han (TP059609)	Lim Sheng Jian (TP056549)
Signature	<i>DeLong</i>	<i>JinHan</i>	<i>ShengJian</i>

References

Anon., 2020. *Traditional File System*. [Online] Available at:
<https://www.geeksforgeeks.org/traditional-file-system/> [Accessed 5 February 2021].

Gunjal, B., 2003. ResearchGate. [Online] Available at:
[https://www.researchgate.net/publication/257298522_Database_Management_Concepts_and_Design#:~:text=A%20database%20management%20system%20\(DBMS,database](https://www.researchgate.net/publication/257298522_Database_Management_Concepts_and_Design#:~:text=A%20database%20management%20system%20(DBMS,database). [Accessed 5th March 2021].

SINGH, C., 2019. *Advantages of DBMS over file system*. [Online] Available at:
<https://beginnersbook.com/2015/04/dbms-vs-file-system/> [Accessed 27 February 2021].