

GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

**CT042-3-1-IDB**

INTRODUCTION TO DATABASES

HAND OUT DATE: 7th APRIL 2021

HAND IN DATE: 2nd JULY 2021

WEIGHTAGE: 40%

TEAM MEMBERS:

* KEVIN MATTHEW ADYAN (TP058466)
* MARCELL AGUNG WAHYUDI (TP058650)
* SELVAN NICHOLAS (TP058084)
* WATARU SHINZATO (TP061217)

Table of Contents

[1.0 DATABASE SCHEMA (STRUCTURE OF DATABASE TABLES / DATA DICTIONARY) 1](#_Toc76059088)

[1.1 Database Schema 1](#_Toc76059089)

[1.2Finalised ERD (Entity Relationship Diagram) – Output of the conceptual database design phase 4](#_Toc76059090)

[1.3Finalised EERD (Enhanced Entity Relationship Diagram) – Output of the logical database design phase 5](#_Toc76059091)

[1.4 Database diagram generated from the MS SQL Server 2019 8](#_Toc76059092)

[2.0 SQL-DATA DEFINITION LANGUAGE (DDL) 11](#_Toc76059093)

[2.1 SQL DDL USED 11](#_Toc76059094)

[2.1.1 SQL DDL used to create AIRLINE\_T 11](#_Toc76059095)

[2.1.2 SQL DDL used to create BOOKING\_OFFICE\_T 11](#_Toc76059096)

[2.1.3 SQL DDL used to create FLIGHT\_T 11](#_Toc76059097)

[2.1.4 SQL DDL used to create PILOT\_T 12](#_Toc76059098)

[2.1.5 SQL DDL used to create FLIGHT\_ATTENDANT\_T 12](#_Toc76059099)

[2.1.6 SQL DDL used to create CUSTOMER\_T 13](#_Toc76059100)

[2.1.7 SQL DDL used to create CUST\_EMAIL\_T 13](#_Toc76059101)

[2.1.8 SQL DDL used to create CUST\_HPHONE\_NUMBER\_T 13](#_Toc76059102)

[2.1.9 SQL DDL used to create FLIGHT\_BOOKING\_T 14](#_Toc76059103)

[2.1.10 SQL DDL used to create ATTENDANT\_SCHEDULE\_T 14](#_Toc76059104)

[2.1.11 SQL DDL used to create PILOT\_SCHEDULE\_T 15](#_Toc76059105)

[2.1.12 SQL DDL used to create CUST\_BOOKING\_PLACE\_T 15](#_Toc76059106)

[2.2 SAMPLES OF DATA USED 16](#_Toc76059107)

[2.2.1 AIRLINE\_T Table 16](#_Toc76059108)

[2.2.2 BOOKING\_OFFICE\_T Table 16](#_Toc76059109)

[2.2.3 FLIGHT\_T Table 16](#_Toc76059110)

[2.2.4 PILOT\_T Table 17](#_Toc76059111)

[2.2.5 FLIGHT\_ATTENDANT\_T Table 17](#_Toc76059112)

[2.2.6 CUSTOMER\_T Table 17](#_Toc76059113)

[2.2.7 CUST\_EMAIL\_T Table 18](#_Toc76059114)

[2.2.8 CUST\_HPHONE\_NUMBER\_T Table 18](#_Toc76059115)

[2.2.9 FLIGHT\_BOOKING\_T Table 19](#_Toc76059116)

[2.2.10 ATTENDANT\_SCHEDULE\_T Table 19](#_Toc76059117)

[2.2.11 PILOT\_SCHEDULE\_T Table 20](#_Toc76059118)

[2.2.12 CUST\_BOOKING\_PLACE\_T Table 20](#_Toc76059119)

[3.0 SQL DML - DATA MANIPULATION LANGUAGE 21](#_Toc76059120)

[3.1 SQL DML USED 21](#_Toc76059121)

[3.1.1 SQL DDL commands to insert tuples into AIRLINE\_T 21](#_Toc76059122)

[3.1.2 SQL DML commands to insert tuples into BOOKING\_OFFICE\_T 21](#_Toc76059123)

[3.1.3 SQL DML commands to insert tuples into FLIGHT\_T 21](#_Toc76059124)

[3.1.4 SQL DML commands to insert tuples into PILOT\_T 22](#_Toc76059125)

[3.1.5 SQL DML commands to insert tuples into FLIGHT\_ATTENDANT\_T 23](#_Toc76059126)

[3.1.6 SQL DML commands to insert tuples into CUSTOMER\_T 23](#_Toc76059127)

[3.1.7 SQL DML commands to insert tuples into CUST\_EMAIL\_T 24](#_Toc76059128)

[3.1.8 SQL DML commands to insert tuples into CUST\_HPHONE\_NUMBER\_T 24](#_Toc76059129)

[3.1.9 SQL DML commands to insert tuples into FLIGHT\_BOOKING\_T 25](#_Toc76059130)

[3.1.10 SQL DML commands to insert tuples into ATTENDANT\_SCHEDULE\_T 25](#_Toc76059131)

[3.1.11 SQL DML commands to insert tuples into PILOT\_SCHEDULE\_T 25](#_Toc76059132)

[3.1.12 SQL DML commands to insert tuples into CUST\_BOOKING\_PLACE\_T 26](#_Toc76059133)

[3.2 SQL Questions 26](#_Toc76059134)

[3.2.1 - Question 1 (Marcell Agung W) 26](#_Toc76059135)

[3.2.2 - Question 2 (Selvan Nicholas) 28](#_Toc76059136)

[3.2.3- Question 3 (Kevin Matthew Adyan) 31](#_Toc76059137)

[3.2.4 - Question 4 (Wataru) 35](#_Toc76059138)

[4.0 CONCLUSION 38](#_Toc76059139)

[5.0 WORKLOAD MATRIX 39](#_Toc76059140)

[6.0 REFERENCES 41](#_Toc76059141)

# DATABASE SCHEMA (STRUCTURE OF DATABASE TABLES / DATA DICTIONARY)

## Database Schema

Figure 1.1 Database Schema Part 1

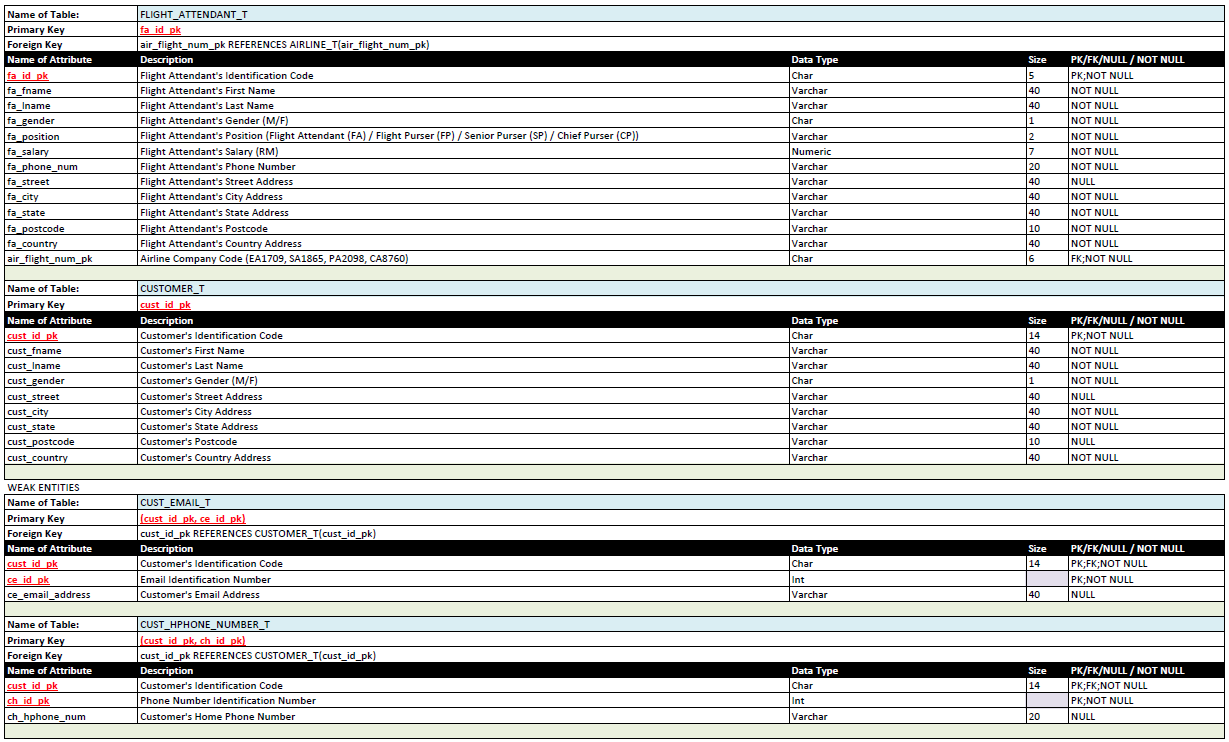
****

Figure 1.2 Database Schema Part 2

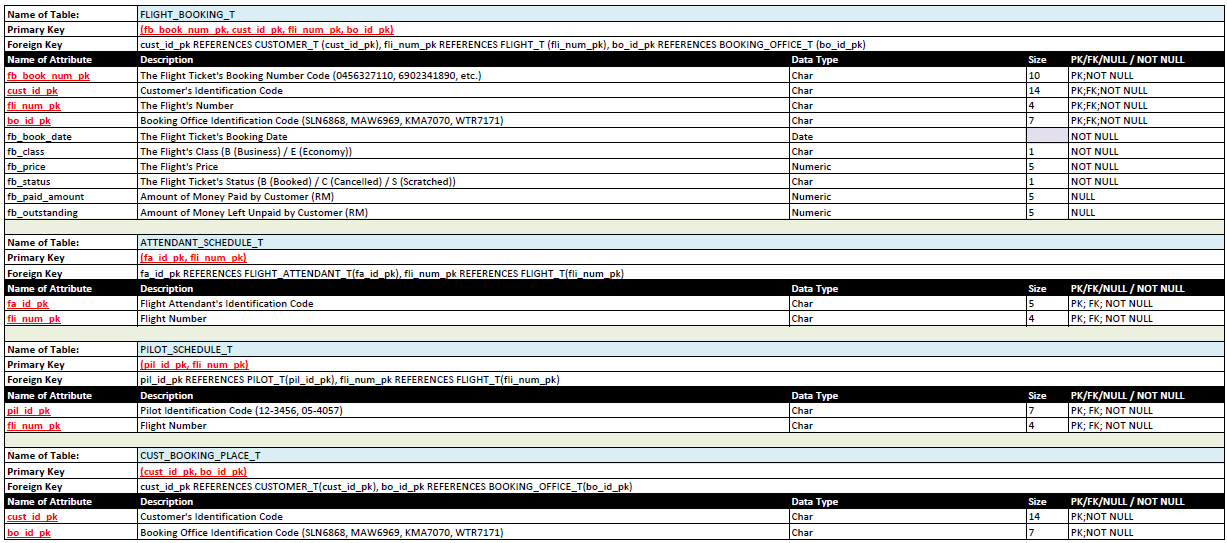
****

Figure 1.3 Database Schema Part 3

## 1.2Finalised ERD (Entity Relationship Diagram) – Output of the conceptual database design phase

Figure 1.4 Finalised ERD

## Diagram Description automatically generated1.3Finalised EERD (Enhanced Entity Relationship Diagram) – Output of the logical database design phase

Figure 1.5 Finalised EERD

Diagram

Description automatically generated

Figure 1.6 Enlarged Finalised Enhanced Entity Relationship Diagram Part 1

Diagram, schematic

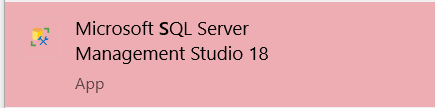
Description automatically generated

Figure 1.7 Enlarged Finalised Enhanced Entity Relationship Diagram Part 2

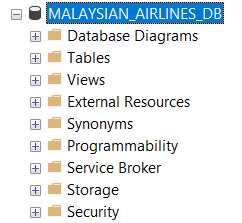
## 1.4 Database diagram generated from the MS SQL Server 2019

Steps to build a database diagram:

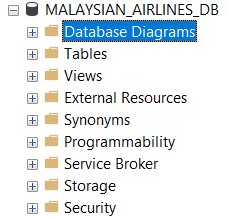
Step 1: Go to MS SQL Server 2019.



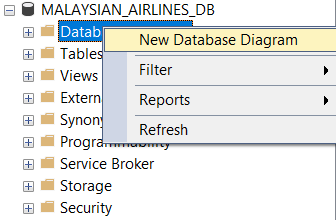
Step 2: Select your database.



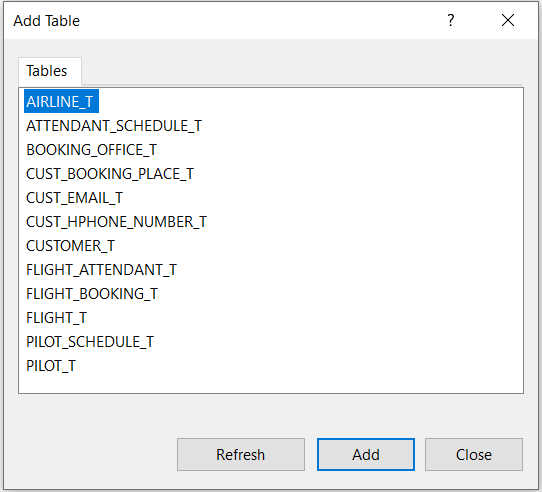
Step 3: Right click on ‘Database Diagrams’.



Step 4: Select ‘New Database Diagram’.



Step 5: Select all the database tables and click ‘Add’.



Step 6: Copy a screenshot and attach it below. That is all.

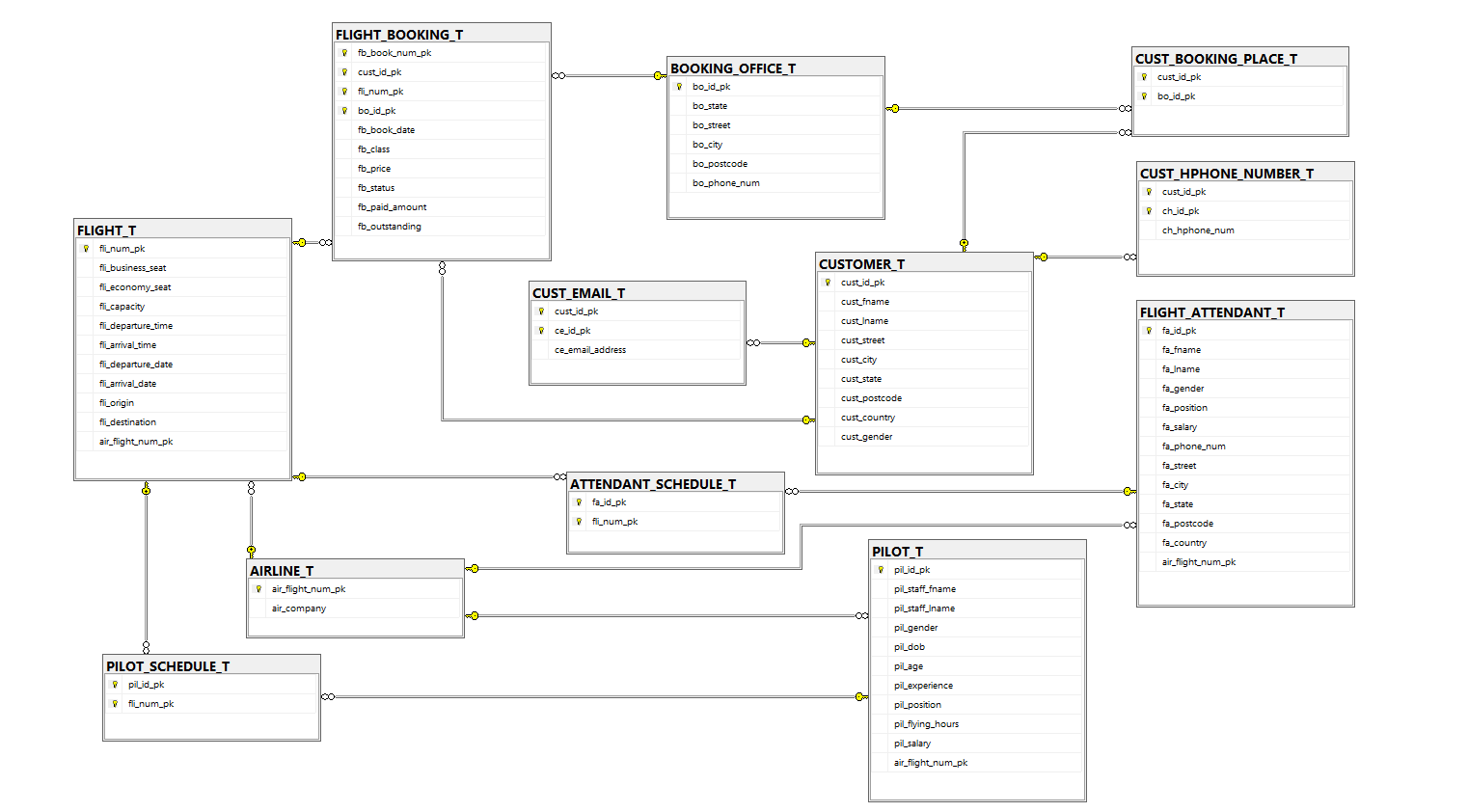


Figure 1.8 Database Diagram

# SQL-DATA DEFINITION LANGUAGE (DDL)

## 2.1 SQL DDL USED

### 2.1.1 SQL DDL used to create AIRLINE\_T

CREATE TABLE AIRLINE\_T

(

air\_flight\_num\_pk CHAR(6) NOT NULL,

air\_company VARCHAR(40) NOT NULL,

PRIMARY KEY (air\_flight\_num\_pk)

);

Diagram

Description automatically generated with medium confidence

Figure 2.1 AIRLINE\_T Primary Key

### 2.1.2 SQL DDL used to create BOOKING\_OFFICE\_T

CREATE TABLE BOOKING\_OFFICE\_T

(

bo\_id\_pk CHAR(7) NOT NULL,

bo\_state VARCHAR(40) NOT NULL,

bo\_street VARCHAR(60) NOT NULL,

bo\_city VARCHAR(40) NOT NULL,

bo\_postcode VARCHAR(10) NULL,

bo\_phone\_num VARCHAR(20) NOT NULL,

PRIMARY KEY (bo\_id\_pk)

);

Text

Description automatically generated with medium confidence

Figure 2.2 BOOKING\_OFFICE\_T Primary Key

### 2.1.3 SQL DDL used to create FLIGHT\_T

CREATE TABLE FLIGHT\_T

(

fli\_num\_pk CHAR(4) NOT NULL,

fli\_business\_seat NUMERIC(3) NULL,

fli\_economy\_seat NUMERIC(3) NOT NULL,

fli\_capacity NUMERIC(3) NOT NULL,

fli\_departure\_time TIME NOT NULL,

fli\_arrival\_time TIME NOT NULL,

fli\_departure\_date DATE NOT NULL,

fli\_arrival\_date DATE NOT NULL,

fli\_origin VARCHAR(40) NOT NULL,

fli\_destination VARCHAR(40) NOT NULL,

air\_flight\_num\_pk CHAR(6) NOT NULL,

PRIMARY KEY (fli\_num\_pk),

FOREIGN KEY (air\_flight\_num\_pk) REFERENCES AIRLINE\_T(air\_flight\_num\_pk)

);

Text

Description automatically generated

Figure 2.3 FLIGHT\_T Primary Key and Foreign Key

### 2.1.4 SQL DDL used to create PILOT\_T

CREATE TABLE PILOT\_T

(

pil\_id\_pk CHAR(7) NOT NULL,

pil\_staff\_fname VARCHAR(40) NOT NULL,

pil\_staff\_lname VARCHAR(40) NOT NULL,

pil\_gender CHAR(1) NOT NULL,

pil\_dob DATE NOT NULL,

pil\_age NUMERIC(3) NOT NULL,

pil\_experience NUMERIC(2) NOT NULL,

pil\_position VARCHAR(2) NOT NULL,

pil\_flying\_hours NUMERIC(6) NOT NULL,

pil\_salary NUMERIC(8) NOT NULL,

air\_flight\_num\_pk CHAR(6) NOT NULL,

PRIMARY KEY (pil\_id\_pk),

FOREIGN KEY (air\_flight\_num\_pk) REFERENCES AIRLINE\_T(air\_flight\_num\_pk)

);

Text

Description automatically generated

Figure 2.4 PILOT\_T Primary Key and Foreign Key

### 2.1.5 SQL DDL used to create FLIGHT\_ATTENDANT\_T

CREATE TABLE FLIGHT\_ATTENDANT\_T

(

fa\_id\_pk CHAR(5) NOT NULL,

fa\_fname VARCHAR(40) NOT NULL,

fa\_lname VARCHAR(40) NOT NULL,

fa\_gender CHAR(1) NOT NULL,

fa\_position VARCHAR(2) NOT NULL,

fa\_salary NUMERIC(7) NOT NULL,

fa\_phone\_num VARCHAR(20) NOT NULL,

fa\_street VARCHAR(40) NULL,

fa\_city VARCHAR(40) NOT NULL,

fa\_state VARCHAR(40) NOT NULL,

fa\_postcode VARCHAR(10) NOT NULL,

fa\_country VARCHAR(40) NOT NULL,

air\_flight\_num\_pk CHAR(6) NOT NULL,

PRIMARY KEY (fa\_id\_pk),

FOREIGN KEY (air\_flight\_num\_pk) REFERENCES AIRLINE\_T(air\_flight\_num\_pk)

);

Text

Description automatically generated

Figure 2.5 FLIGHT\_ATTENDANT\_T Primary Key and Foreign Key

### 2.1.6 SQL DDL used to create CUSTOMER\_T

CREATE TABLE CUSTOMER\_T

(

cust\_id\_pk CHAR(14) NOT NULL,

cust\_fname VARCHAR(40) NOT NULL,

cust\_lname VARCHAR(40) NOT NULL,

cust\_street VARCHAR(40) NULL,

cust\_city VARCHAR(40) NOT NULL,

cust\_state VARCHAR(40) NOT NULL,

cust\_postcode VARCHAR(10) NULL,

cust\_country VARCHAR(40) NOT NULL,

cust\_gender CHAR(1) NOT NULL,

PRIMARY KEY (cust\_id\_pk)

);

Text

Description automatically generated

Figure 2.6 CUSTOMER\_T Primary Key

### 2.1.7 SQL DDL used to create CUST\_EMAIL\_T

CREATE TABLE CUST\_EMAIL\_T

(

cust\_id\_pk CHAR(14) NOT NULL,

ce\_id\_pk INT NOT NULL,

ce\_email\_address VARCHAR(40) NOT NULL,

PRIMARY KEY (cust\_id\_pk, ce\_id\_pk),

FOREIGN KEY (cust\_id\_pk) REFERENCES CUSTOMER\_T(cust\_id\_pk)

);

Text, letter

Description automatically generated

Figure 2.7 CUST\_EMAIL\_T Primary Key and Foreign Key

### 2.1.8 SQL DDL used to create CUST\_HPHONE\_NUMBER\_T

CREATE TABLE CUST\_HPHONE\_NUMBER\_T

(

cust\_id\_pk CHAR(14) NOT NULL,

ch\_id\_pk INT NOT NULL,

ch\_hphone\_num VARCHAR(20) NOT NULL,

PRIMARY KEY (cust\_id\_pk, ch\_id\_pk),

FOREIGN KEY (cust\_id\_pk) REFERENCES CUSTOMER\_T(cust\_id\_pk)

);

Text, letter

Description automatically generated

Figure 2.8 CUST\_HPHONE\_NUMBER Primary Key and Foreign Key

### 2.1.9 SQL DDL used to create FLIGHT\_BOOKING\_T

CREATE TABLE FLIGHT\_BOOKING\_T

(

fb\_book\_num\_pk CHAR(10) NOT NULL,

cust\_id\_pk CHAR(14) NOT NULL,

fli\_num\_pk CHAR(4) NOT NULL,

bo\_id\_pk CHAR(7) NOT NULL,

fb\_book\_date DATE NOT NULL,

fb\_class CHAR(1) NOT NULL,

fb\_price NUMERIC(5) NOT NULL,

fb\_status CHAR(1) NOT NULL,

fb\_paid\_amount NUMERIC(5) NULL,

fb\_outstanding NUMERIC(5) NULL,

PRIMARY KEY (fb\_book\_num\_pk, cust\_id\_pk, fli\_num\_pk, bo\_id\_pk),

FOREIGN KEY (cust\_id\_pk) REFERENCES CUSTOMER\_T(cust\_id\_pk),

FOREIGN KEY (fli\_num\_pk) REFERENCES FLIGHT\_T(fli\_num\_pk),

FOREIGN KEY (bo\_id\_pk) REFERENCES BOOKING\_OFFICE\_T(bo\_id\_pk)

);

Text, letter

Description automatically generated

Figure 2.9 FLIGHT\_BOOKING\_T Primary Key and Foreign Keys

### 2.1.10 SQL DDL used to create ATTENDANT\_SCHEDULE\_T

CREATE TABLE ATTENDANT\_SCHEDULE\_T

(

fa\_id\_pk CHAR(5) NOT NULL,

fli\_num\_pk CHAR(4) NOT NULL,

PRIMARY KEY (fa\_id\_pk, fli\_num\_pk),

FOREIGN KEY (fa\_id\_pk) REFERENCES FLIGHT\_ATTENDANT\_T (fa\_id\_pk),

FOREIGN KEY (fli\_num\_pk) REFERENCES FLIGHT\_T (fli\_num\_pk)

);

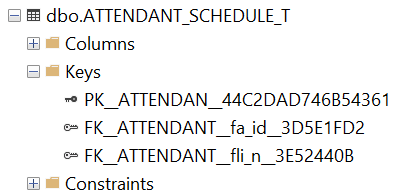


Figure 2.10 ATTENDANT\_SCHEDULE\_T Primary Key and Foreign Keys

### 2.1.11 SQL DDL used to create PILOT\_SCHEDULE\_T

CREATE TABLE PILOT\_SCHEDULE\_T

(

pil\_id\_pk CHAR(7) NOT NULL,

fli\_num\_pk CHAR(4) NOT NULL,

PRIMARY KEY (pil\_id\_pk, fli\_num\_pk),

FOREIGN KEY (pil\_id\_pk) REFERENCES PILOT\_T (pil\_id\_pk),

FOREIGN KEY (fli\_num\_pk) REFERENCES FLIGHT\_T (fli\_num\_pk)

);

Text, letter

Description automatically generated

Figure 2.11 PILOT\_SCHEDULE\_T Primary Key and Foreign Keys

### 2.1.12 SQL DDL used to create CUST\_BOOKING\_PLACE\_T

CREATE TABLE CUST\_BOOKING\_PLACE\_T

(

cust\_id\_pk CHAR(14) NOT NULL,

bo\_id\_pk CHAR(7) NOT NULL,

PRIMARY KEY (cust\_id\_pk, bo\_id\_pk),

FOREIGN KEY (cust\_id\_pk) REFERENCES CUSTOMER\_T(cust\_id\_pk),

FOREIGN KEY (bo\_id\_pk) REFERENCES BOOKING\_OFFICE\_T(bo\_id\_pk)

);

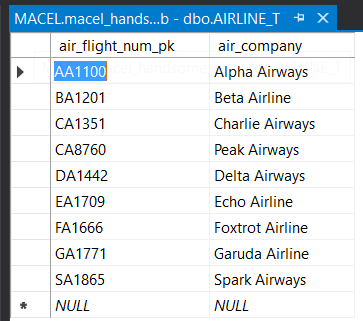
Text, letter

Description automatically generated

Figure 2.12 CUST\_BOOKING\_PLACE\_T Primary Key and Foreign Keys

## 2.2 SAMPLES OF DATA USED

### 2.2.1 AIRLINE\_T Table

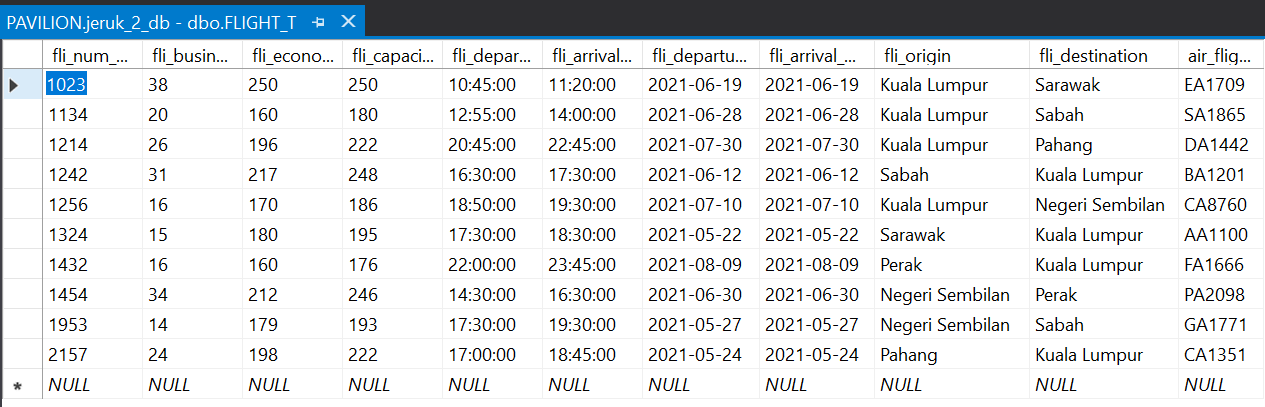


### 2.2.2 BOOKING\_OFFICE\_T Table

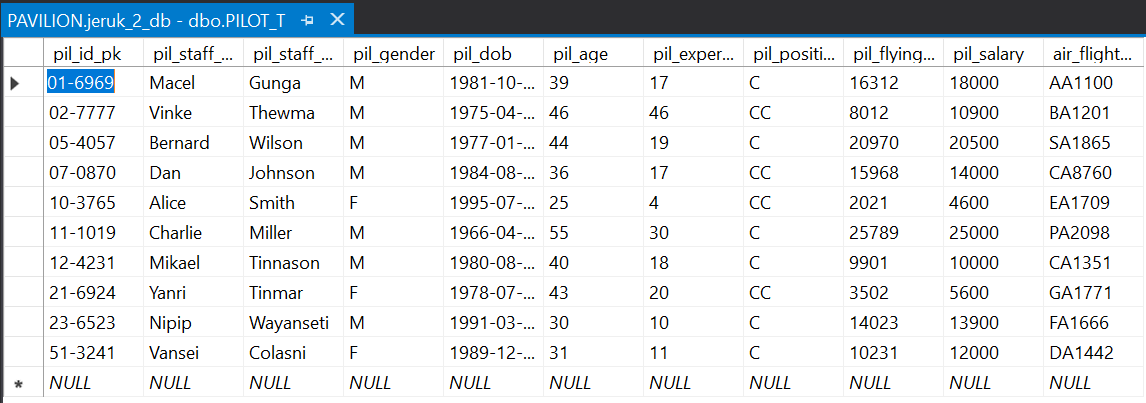
Graphical user interface, text, application, email

Description automatically generated

### 2.2.3 FLIGHT\_T Table



### 2.2.4 PILOT\_T Table



### 2.2.5 FLIGHT\_ATTENDANT\_T Table

### 2.2.6 CUSTOMER\_T Table

### 2.2.7 CUST\_EMAIL\_T Table

Table

Description automatically generated

### 2.2.8 CUST\_HPHONE\_NUMBER\_T Table

Table

Description automatically generated

### 2.2.9 FLIGHT\_BOOKING\_T Table

Graphical user interface, table

Description automatically generated with medium confidence

### 2.2.10 ATTENDANT\_SCHEDULE\_T Table

Table

Description automatically generated

### 2.2.11 PILOT\_SCHEDULE\_T Table

Table

Description automatically generated

### 2.2.12 CUST\_BOOKING\_PLACE\_T Table

Table

Description automatically generated

# SQL DML - DATA MANIPULATION LANGUAGE

## 3.1 SQL DML USED

### 3.1.1 SQL DDL commands to insert tuples into AIRLINE\_T

INSERT INTO AIRLINE\_T

(

air\_flight\_num\_pk,

air\_company)

VALUES　('AA1100','Alpha Airways'),

('BA1201','Beta Airline'),

('CA1351','Charlie Airways'),

('CA8760','Core Airways'),

('PA2098','Peak Airways'),

('DA1442','Delta Airways'),

('EA1709','Echo Airline'),

('FA1666','Foxtrot Airline'),

('GA1771','Garuda Airline'),

('SA1865','Spark Airline');

### 3.1.2 SQL DML commands to insert tuples into BOOKING\_OFFICE\_T

INSERT INTO BOOKING\_OFFICE\_T

(

bo\_id\_pk,

bo\_state,

bo\_street,

bo\_city,

bo\_postcode,

bo\_phone\_num)

VALUES ('KDH6969','Kedah','14, Lorong Air Masin','Kuala Kedah','35000','08461345144'),

('KLT0590','Kelantan','69, Jalan Dusun Raja','Kota Bhuru','46000','08124657981'),

('NSN6969','Negeri Sembilan','8, Millennia Business Centre Jin Tan Sri Manickavasagam','Seremban','70200','0626989333'),

('PHG7070','Pahang','B230,1St Floor Jīn Air Putih','Kuantan','25300','052681214'),

('PKL7373','Wilayah Perseku','12, Lorong Medan Tuanku Satu','Kuala Lumpur','50300','0362033984'),

('PNG1340','Pulau Pinang','11, Lbh Muntri','Georgetown','10200','082111110808'),

('PRK6868','Perak','143, Leboh Perajurit 3 Garden East','Ipoh','31400','0556375098'),

('SBH7171','Sabah','1, Wisma New Far East Taman Far East','Kota Kinabalu','88300','052822931149'),

('SRW7272','Sarawak','1,Jalan Padungan Utara','Kuching','93100','0622247778'),

('TRG3210','Terengganu','14, Jalan Sultan Ismail','Kuala Terengganu','14500','08216549211');

### 3.1.3 SQL DML commands to insert tuples into FLIGHT\_T

INSERT INTO FLIGHT\_T

(

fli\_num\_pk,

fli\_business\_seat,

fli\_economy\_seat,

fli\_capacity,

fli\_departure\_time,

fli\_arrival\_time,

fli\_departure\_date,

fli\_arrival\_date,

fli\_origin,

fli\_destination,

air\_flight\_num\_pk)

VALUES ('1023',38,250,250,'10:45:00','11:20:00','2021-06-19','2021-06-19','Kuala Lumpur','Sarawak','EA1709'),

('1134',20,160,180,'12:55:00','14:00:00','2021-06-28','2021-06-28','Kuala Lumpur','Sabah','SA1865'),

('1214',26,196,222,'20:45:00','22:45:00','2021-07-30','2021-07-30','Kuala Lumpur','Pahang','DA1442'),

('1242',31,217,248,'16:30:00','17:30:00','2021-06-12','2021-06-12','Sabah','Kuala Lumpur','BA1201'),

('1256',16,170,186,'18:50:00','19:30:00','2021-07-10','2021-07-10','Kuala Lumpur','Negeri Sembilan','CA8760'),

('1324',15,180,195,'17:30:00','18:30:00','2021-05-22','2021-05-22','Sarawak','Kuala Lumpur','AA1100'),

('1432',16,160,176,'22:00:00','23:45:00','2021-08-09','2021-08-09','Perak','Kuala Lumpur','FA1666'),

('1454',34,212,246,'14:30:00','16:30:00','2021-06-30','2021-06-30','Negeri Sembilan','Perak','PA2098'),

('1953',14,179,193,'17:30:00','19:30:00','2021-05-27','2021-05-27','Negeri Sembilan','Sabah','GA1771'),

('2157',24,198,222,'17:00:00','18:45:00','2021-05-24','2021-05-24','Pahang','Kuala Lumpur','CA1351');

### 3.1.4 SQL DML commands to insert tuples into PILOT\_T

INSERT INTO PILOT\_T

(

pil\_id\_pk,

pil\_staff\_fname,

pil\_staff\_lname,

pil\_gender,

pil\_dob,

pil\_age,

pil\_experience,

pil\_position,

pil\_flying\_hours,

pil\_salary,

air\_flight\_num\_pk)

VALUES ('01-6969','Macel','Gunga','M','1981-10-13',39,17,'C',16312,18000,'AA1100'),

('02-7777','Vinke','Thewma','M','1975-04-02',46,46,'CC',8012,10900,'BA1201'),

('05-4057','Bernard','Wilson','M','1977-01-01',44,19,'C',20970,20500,'SA1865'),

('07-0870','Dan','Johnson','M','1984-08-05',36,17,'CC',15968,14000,'CA8760'),

('10-3765','Alice','Smith','F','1995-07-27',25,4,'CC',2021,4600,'EA1709'),

('11-1019','Charlie','Miller','M','1966-04-08',55,30,'C',25789,25000,'PA2098'),

('12-4231','Mikael','Tinnason','M','1980-08-17',40,18,'C',9901,10000,'CA1351'),

('21-6924','Yanri','Tinmar','F','1978-07-30',43,20,'CC',3502,5600,'GA1771'),

('23-6523','Nipip','Wayanseti','M','1991-03-22',30,10,'C',14023,13900,'FA1666'),

('51-3241','Vansei','Colasni','F','1989-12-13',31,11,'C',10231,12000,'DA1442');

### 3.1.5 SQL DML commands to insert tuples into FLIGHT\_ATTENDANT\_T

INSERT INTO FLIGHT\_ATTENDANT\_T

(

fa\_id\_pk,

fa\_fname,

fa\_lname,

fa\_gender,

fa\_position,

fa\_salary,

fa\_phone\_num,

fa\_street,

fa\_city,

fa\_state,

fa\_postcode,

fa\_country,

air\_flight\_num\_pk)

VALUES ('0-564','Ann','Ross','F','CP',8000,'6026933777','Jalan Tanjung Chat 4','Kota Bharu','Kelantan','15400','Malaysia','SA1865'),

('0-565','John','Taylor','M','FP',3500,'6521245618','678A Jurong West Street 64','Singapore','Singapore','641678','Singapore','CA8760'),

('1-235','Catherine','Smith','F','FA',2800,'6024154233','Jalan Teknologi 5','Kuala Lumpur','Wilayah Persekutuan Kuala Lumpur','57000','Malaysia','EA1709'),

('1-333','Mike','Hunt','M','FA',3000,'6223897534','Jalan Benteng Jaya 16','Kota Tangerang','Banten','15111','Indonesia','PA2098'),

('2-252','Rihen','Kelmai','M','FA',3200,'62965748351','Jalan Soekarno Hatta 61','Kota Balikpapan','East Kalimantan','76214','Indonesia','DA1442'),

('4-562','Kelmic','Kom','M','SP',6900,'6598217168','Jalan Setiabudhi 8','Kota Bandung','Jawa Barat','40141','Indonesia','BA1201'),

('6-693','Terpe','Yenbri','M','FP',5000,'65789451345','Jalan Bung Tomo 4','Kota Surabaya','East Java','60245','Indonesia','FA1666'),

('6-969','Ranzah','Nihaf','F','FA',2800,'62212691611','Jalan Shonda Makurmpad','Kota Malang','East Java','65113','Indonesia','GA1771'),

('9-325','Vinke','Wansei','F','CP',8100,'6519452014','Jalan Dermaga 69','Kota Samarinda','East Kalimantan','75242','Indonesia','AA1100'),

('9-936','Nathanyo','Cenvi','F','FA',3000,'62821450142','Grusiva Niyamasubar St','City of Seria','Belait','2733','Brunei','CA1351');

### 3.1.6 SQL DML commands to insert tuples into CUSTOMER\_T

INSERT INTO CUSTOMER\_T

(

cust\_id\_pk,

cust\_fname,

cust\_lname,

cust\_street,

cust\_city,

cust\_state,

cust\_postcode,

cust\_country,

cust\_gender)

VALUES ('123400-56-9876','Ching','Chong','Jiu Wei Cun 81 Hao','Sha Tian','Guang Dong','523000','China','F'),

('164973-59-1576','Usman','Mailis','No.10 Kampung','Bota','Perak','32600','Malaysia','M'),

('550106-12-5821','Dig','Beck','Seksyen Bb 11, Bukit Beruntung','Rawang','Selangor','48300','Malaysia','M'),

('581235-61-2456','Dhason','Padnakunar','Lot 3706 Jin Tengah','Bandar Seri Begawan','Brunei and Muara','8411','Brunei','M'),

('610522-10-5378','Ligma','Bllas','No.12 Jalan Bersatu (13/4)','Petaling Jaya','Selangor','46200','Brunei','F'),

('615493-59-7984','Sivaguru','Kael','No.3 Jln Sek Keb','Bukit Payong','Terengganu','21400','Malaysia','M'),

('923497-36-5951','Tanvir','Subarnamiyn','20-1 A Jln Pjs 3/36 Taman Sri Manja','Petaling Jaya','Selangor','58200','Malaysia','M'),

('930906-14-7043','Ben','Dover','No.8 Jin Air Panas','Kuala Lumpur','Wilayah Persekutuan Kuala Lumpur','53200','Malaysia','M'),

('941130-07-5153','Icewallow','Kam','Jalan Setiabudi 10','Malang','Jawa Timur','65111','Indonesia','F'),

('985632-41-6932','Chad','Upin','66A Jln Tabuan','Kuching','Sarawak','93100','Malaysia','M'),

('691234-43-7872','Ahmad','Qafi','13 Jalan Payah, Kampung Aik Hwa,','Muar','Johor','84200','Malaysia','M'),

('771203-90-9922','Rosa','Fatima', '13 Jalan Payah, Kampung Aik Hwa,','Muar','Johor','84200','Malaysia','F');

### 3.1.7 SQL DML commands to insert tuples into CUST\_EMAIL\_T

INSERT INTO CUST\_EMAIL\_T

(

cust\_id\_pk,

ce\_id\_pk,

ce\_email\_address)

VALUES ('550106-12-5821',1,'digbeck@hotmail.com'),

('930906-14-7043',2,'bendover69@yahoo.com'),

('941130-07-5153',3,'wallowicekam123@gmail.com'),

('610522-10-5378',4,'ligmaballs@hotmail.com'),

('123400-56-9876',5,'chingchongchang@yahoo.com'),

('581235-61-2456',6,'dhason@apu.edu.my'),

('615493-59-7984',7,'tanvir@apu.edu.my'),

('923497-36-5951',8,'sivaguru123@yahoo.com'),

('164973-59-1576',9,'usmanthegreat@yahoo.com'),

('985632-41-6932',10,'xXDemoLordXx@yahoo.com')

### 3.1.8 SQL DML commands to insert tuples into CUST\_HPHONE\_NUMBER\_T

INSERT INTO CUST\_HPHONE\_NUMBER\_T

(

cust\_id\_pk,

ch\_id\_pk,

ch\_hphone\_num)

VALUES ('550106-12-5821',1,'6084901516'),

('930906-14-7043',2,'60249241562'),

('941130-07-5153',3,'62498456852'),

('610522-10-5378',4,'67180200120'),

('123400-56-9876',5,'8156562001'),

('581235-61-2456',6,'67199998877'),

('615493-59-7984',7,'60164624389'),

('923497-36-5951',8,'608214944201'),

('164973-59-1576',9,'60121249895'),

('985632-41-6932',10,'60333321644');

### 3.1.9 SQL DML commands to insert tuples into FLIGHT\_BOOKING\_T

INSERT INTO FLIGHT\_BOOKING\_T

(

fb\_book\_num\_pk,

fli\_num\_pk,

cust\_id\_pk,

bo\_id\_pk,

fb\_book\_date,

fb\_class,

fb\_price,

fb\_status,

fb\_paid\_amount,

fb\_outstanding)

VALUES ('0456327110','1023','550106-12-5821','PKL7373','2021-04-17','B',600,'S',100,400),

('6902341890','1134','930906-14-7043','PKL7373','2020-08-09','E',220,'B',200,20),

('9736440028','1454','941130-07-5153','NSN6969','2020-08-16','E',230,'C', NULL, NULL),

('8969912362','1256','610522-10-5378','PKL7373','2020-12-21','E',170,'B',170, NULL),

('6124953478','1324','123400-56-9876','SBH7171','2020-12-14','B',1000,'B',900,100),

('8453125549','1242','581235-61-2456','SRW7272','2021-02-06','B',575,'B',550,25),

('1324875613','1432','615493-59-7984','PHG7070','2021-01-09','E',235,'C', NULL, NULL),

('4950351204','2157','923497-36-5951','PKL7373','2021-04-12','E',220,'S', NULL, NULL),

('0326592175','1953','164973-59-1576','PRK6868','2020-10-13','E',215,'S', NULL, NULL),

('1203562047','1214','985632-41-6932','NSN6969','2020-12-19','B',240,'C', NULL, NULL);

### 3.1.10 SQL DML commands to insert tuples into ATTENDANT\_SCHEDULE\_T

INSERT INTO ATTENDANT\_SCHEDULE\_T

(

fa\_id\_pk,

fli\_num\_pk)

VALUES ('1-235','1023'),

('0-564','1134'),

('1-333','1454'),

('0-564','1256'),

('9-325','1324'),

('4-562','1242'),

('9-936','2157'),

('2-252','1214'),

('6-693','1432'),

('6-969','1953');

### 3.1.11 SQL DML commands to insert tuples into PILOT\_SCHEDULE\_T

INSERT INTO PILOT\_SCHEDULE\_T

(

pil\_id\_pk,

fli\_num\_pk)

VALUES ('10-3765','1023'),

('05-4057','1134'),

('11-1019','1454'),

('07-0870','1256'),

('01-6969','1324'),

('02-7777','1242'),

('12-4231','2157'),

('51-3241','1214'),

('23-6523','1432'),

('21-6924','1953');

### 3.1.12 SQL DML commands to insert tuples into CUST\_BOOKING\_PLACE\_T

INSERT INTO CUST\_BOOKING\_PLACE\_T

(

cust\_id\_pk,

bo\_id\_pk)

VALUES ('550106-12-5821','PKL7373'),

('930906-14-7043','PKL7373'),

('941130-07-5153','NSN6969'),

('610522-10-5378','PKL7373'),

('123400-56-9876','SBH7171'),

('581235-61-2456','SRW7272'),

('615493-59-7984','PHG7070'),

('923497-36-5951','PKL7373'),

('164973-59-1576','PRK6868'),

('985632-41-6932','NSN6969');

## 3.2 SQL Questions

### 3.2.1 - Question 1 (Marcell Agung W)

1. Display all customers whose first name ends with consonant ‘*d*’ and lives in *Sarawak.*

**SQL DML CODE**

SELECT c.cust\_fname AS 'Customer First Name',

c.cust\_lname AS 'Customer Last Name'

FROM CUSTOMER\_T c

WHERE ( c.cust\_fname LIKE '%d' ) AND

( c.cust\_state LIKE 'Sarawak' )

**SQL SERVER RESULT**

Text

Description automatically generated

Figure 3.1 Question 1 Student 1 SQL Server Result

1. Display the average of salary for pilots who have reached or exceeded *20,000* flying hours.

**SQL DML CODE**

SELECT AVG(p.pil\_salary) AS 'Average Pilot Salary'

FROM PILOT\_T p

WHERE ( p.pil\_flying\_hours >= 20000 )

**SQL SERVER RESULT**

Graphical user interface, text

Description automatically generated

Figure 3.2 Question 2 Student 1 SQL Server Result

1. List the first name, last name, age, and experience of pilots who have piloted the flight for *Spark Airways.*

**SQL DML CODE**

SELECT p.pil\_staff\_fname AS 'Pilot First Name',

p.pil\_staff\_lname AS 'Pilot Last Name',

p.pil\_age AS 'Pilot Age',

p.pil\_experience AS 'Pilot Experience'

FROM PILOT\_T p, AIRLINE\_T a

WHERE ( p.air\_flight\_num\_pk = a.air\_flight\_num\_pk ) AND

( a.air\_company LIKE 'Spark Airways' )

**SQL SERVER RESULT**A picture containing text, screenshot, monitor

Description automatically generated

Figure 3.3 Question 3 Student 1 SQL Server Result

### 3.2.2 - Question 2 (Selvan Nicholas)

1. Display customer’s first name and last name who have made bookings.

**SQL DML CODE**

SELECT c.cust\_fname AS 'CUSTOMER FIRST NAME',

c.cust\_lname AS 'CUSTOMER LAST NAME'

FROM CUSTOMER\_T c, CUST\_BOOKING\_PLACE\_T bp

WHERE ( ( c.cust\_id\_pk = bp.cust\_id\_pk ) )

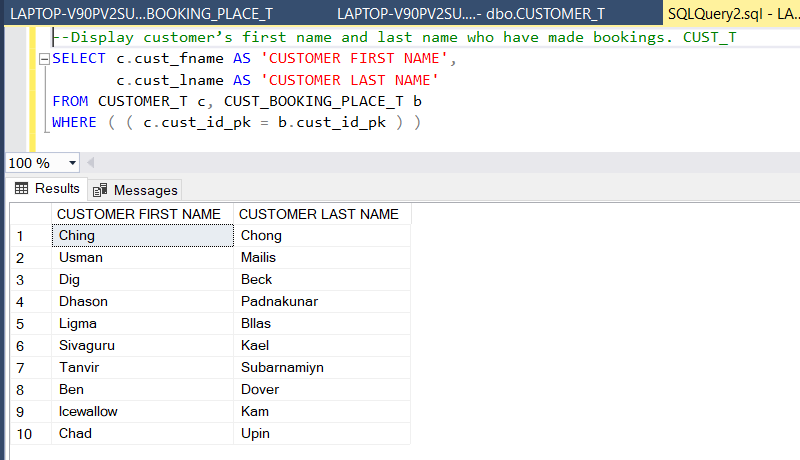
**SQL SERVER RESULT**

Figure 3.4 Question 1 Student 2 SQL Server Result

1. List flight attendant’s full name and position who have worked in the same flight as a pilot named ‘*Mikael Tinnason*’.

**SQL DML CODE**

SELECT fa.fa\_fname AS 'FLIGHT ATTENDANT FIRST NAME',

fa.fa\_lname AS 'FLIGHT ATTENDANT LAST NAME',

fa.fa\_position AS 'FLIGHT ATTENDANT POSITION'

FROM FLIGHT\_ATTENDANT\_T fa, ATTENDANT\_SCHEDULE\_T att

WHERE ( (fa.fa\_id\_pk = att.fa\_id\_pk) AND

att.fli\_num\_pk =

( SELECT ps.fli\_num\_pk AS 'PILOT FLIGHT NUMBER'

FROM PILOT\_T p, PILOT\_SCHEDULE\_T ps

WHERE ( ( ps.pil\_id\_pk = p.pil\_id\_pk ) AND

( p.pil\_staff\_fname = 'Mikael' AND

( p.pil\_staff\_lname = 'Tinnason' ) )

)

)

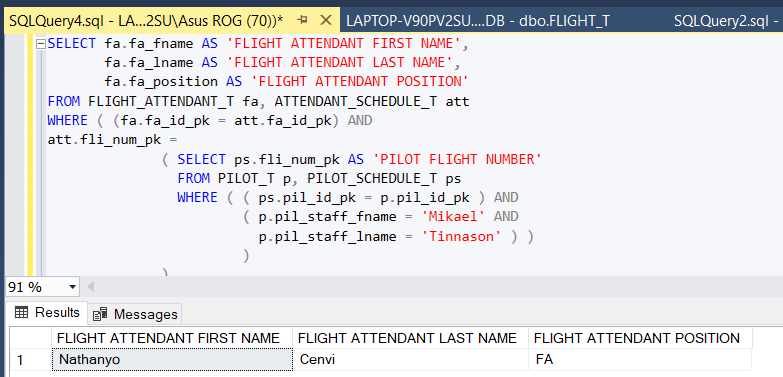


Figure 3.5 Question 2 Student 2 SQL Server Result

**SQL SERVER RESULT**

1. List all customers who did not live in any of the airline offices located. Please display the customer first name, last name, and customer’s state.

**SQL DML CODE**

SELECT c.cust\_fname AS 'CUSTOMER FIRST NAME',

c.cust\_lname AS 'CUSTOMER LAST NAME',

c.cust\_state AS 'CUSTOMER STATE'

FROM CUSTOMER\_T c

WHERE (

c.cust\_state NOT IN ( SELECT bo\_state AS 'BOOKING OFFICE STATE'

FROM BOOKING\_OFFICE\_T bo

)

)

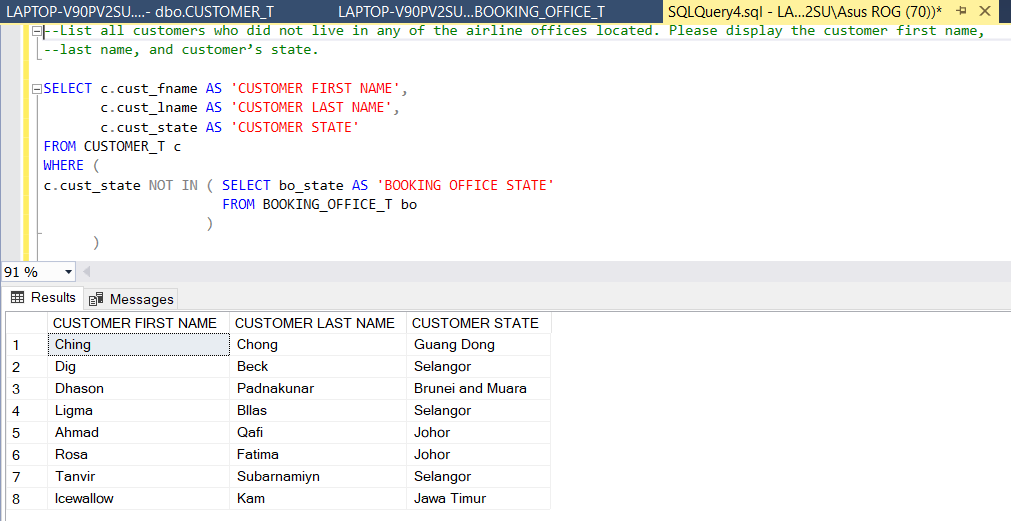


Figure 3.6 Question 3 Student 2 SQL Server Result

**SQL SERVER RESULT**

### 3.2.3- Question 3 (Kevin Matthew Adyan)

1. Display customer’s first name, flight booking number, booking date, flight airlines name, date of departure, time of departure, and status indicator is *Canceled.*

**SQL DML CODE**

SELECT c.cust\_fname AS 'Customer First Name',

fb.fb\_book\_num\_pk AS 'Flight Booking Number',

fb.fb\_book\_date AS 'Flight Booking Date',

a.air\_company AS 'Airline Name',

f.fli\_departure\_date AS 'Departure Date',

f.fli\_departure\_time AS 'Departure Time',

fb.fb\_status AS 'Status'

FROM CUSTOMER\_T c, FLIGHT\_BOOKING\_T fb, AIRLINE\_T a, FLIGHT\_T f

WHERE ( ( fb.fb\_status = 'C' ) AND

( fb.fli\_num\_pk = f.fli\_num\_pk ) AND

( fb.cust\_id\_pk = c.cust\_id\_pk ) AND

( f.air\_flight\_num\_pk = a.air\_flight\_num\_pk ) )

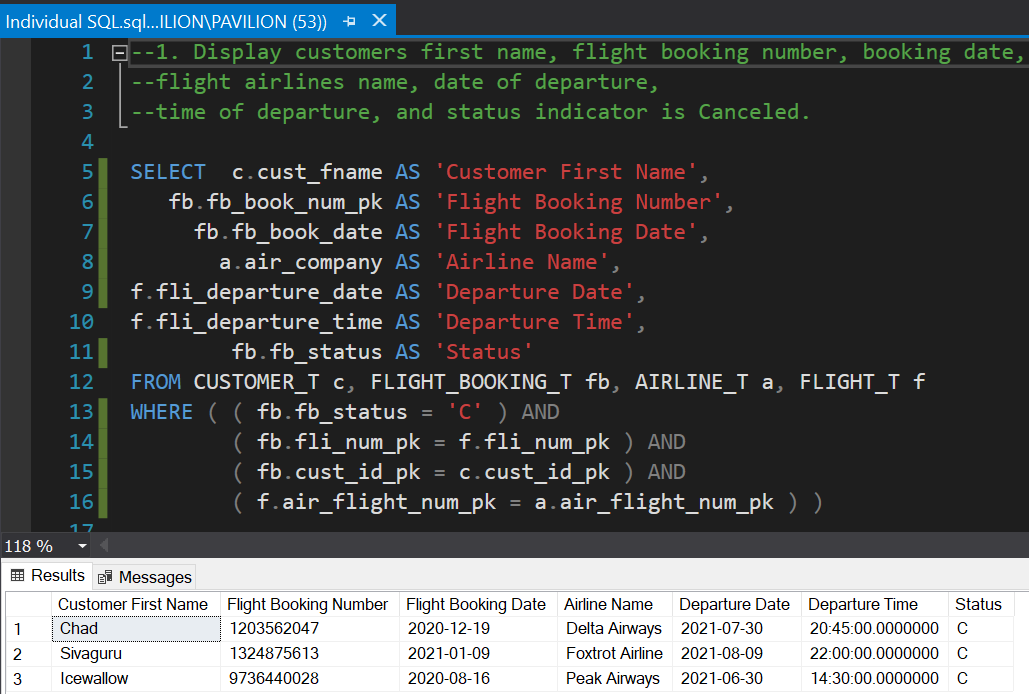
**SQL Server Result**

Figure 3.7 Question 1 Student 3 SQL Server Result

1. List all customer’s first name and last name who did not place any booking. Sort the records by customer id in descending order.

**SQL DML CODE**

SELECT c.cust\_id\_pk AS 'Customer ID',

c.cust\_fname AS 'First Name',

c.cust\_lname AS 'Last Name'

FROM CUSTOMER\_T c

WHERE ( c.cust\_id\_pk NOT IN ( SELECT fb.cust\_id\_pk AS 'Customer ID'

FROM FLIGHT\_BOOKING\_T fb) ) --sub query

ORDER BY c.cust\_id\_pk DESC

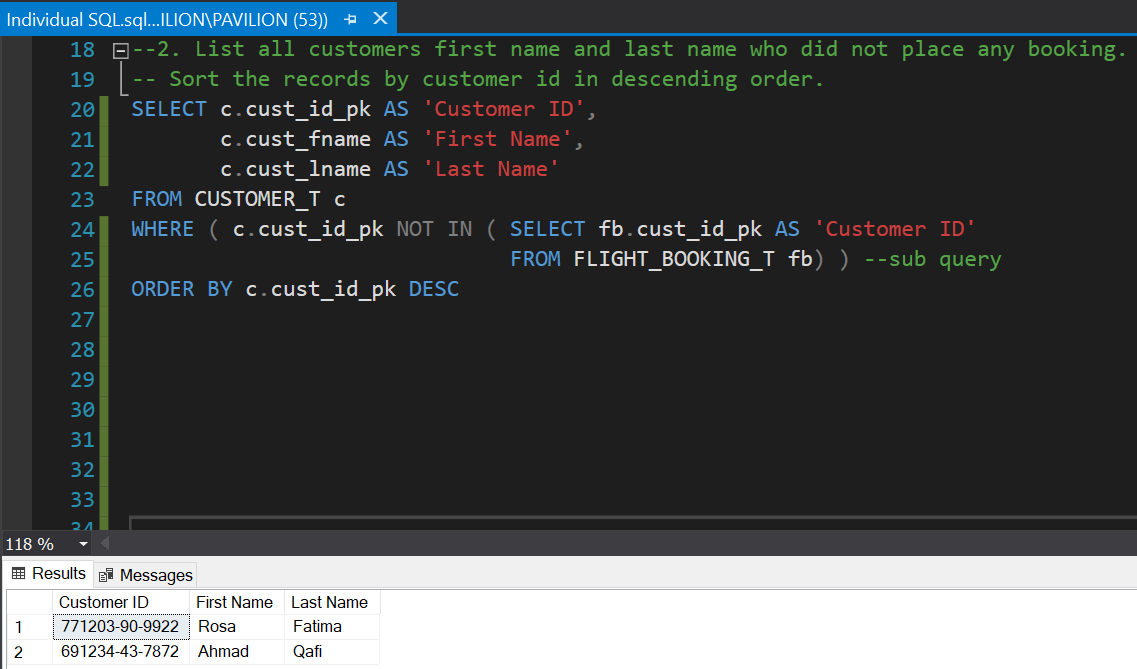
**SQL Server Result**

Figure 3.8 Question 2 Student 3 SQL Server Result

1. List all customer’s first name and last name who still have outstanding balance for their flight bookings. Sort the records based on the outstanding balance in ascending order.

**SQL DML CODE**

SELECT c.cust\_fname AS 'Customer First Name',

c.cust\_lname AS 'Customer Last Name',

fb.fb\_outstanding AS 'Outstanding Balance'

FROM CUSTOMER\_T c, FLIGHT\_BOOKING\_T fb

WHERE ( ( c.cust\_id\_pk = fb.cust\_id\_pk ) AND

( fb.fb\_outstanding IS NOT NULL ) )

ORDER BY fb.fb\_outstanding

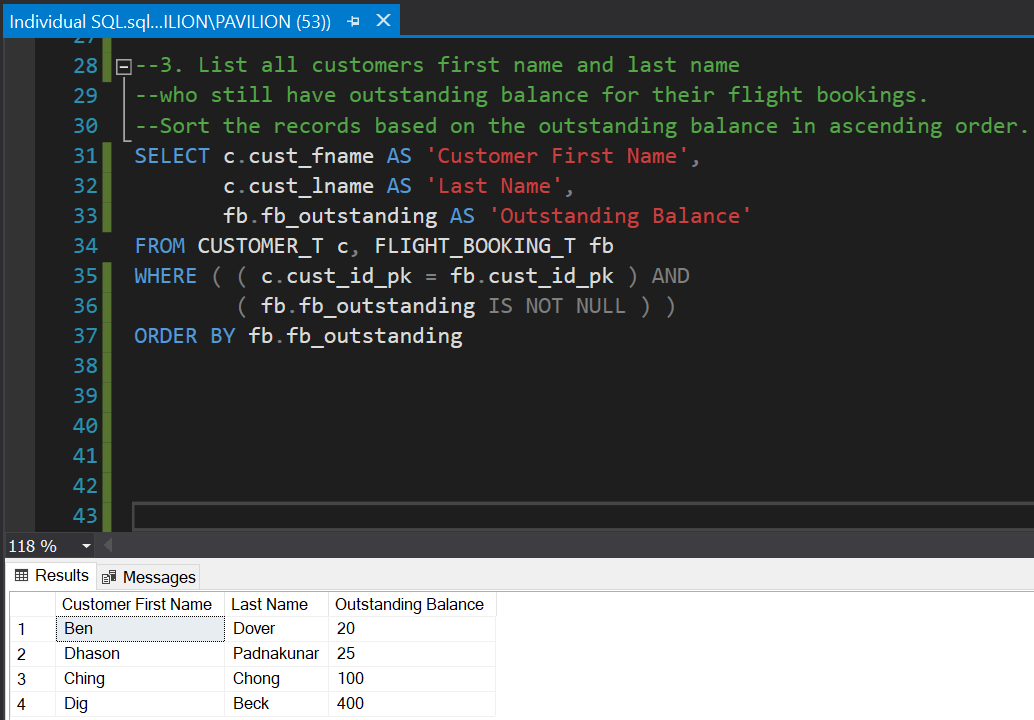
**SQL Server Result**

Figure 3.9 Question 3 Student 3 SQL Server Result

### 3.2.4 - Question 4 (Wataru)

1. Find the highest salary of flight attendants.

**SQL DML CODE**

SELECT MAX (fa.fa\_salary) AS ‘FLIGHT ATTENDANT HIGHEST SALARY’

FROM FLIGHT\_ATTENDANT\_T fa

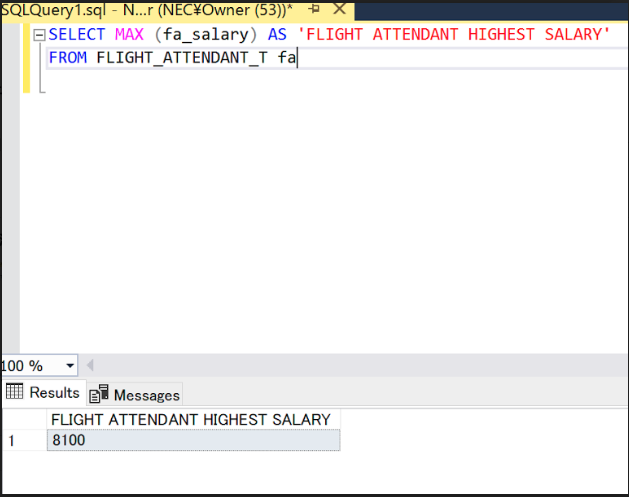
 **SQL Server Result**

Figure 3.10 Question 1 Student 4 SQL Server Result

1. List the name and the position of flight attendants whose salary is neither *2,800* nor *3,500*.

**SQL DML CODE**

SELECT fa.fa\_fname AS ‘ATTENDANT FIRST NAME’,

fa.fa\_lname AS ‘LAST NAME’,

fa.fa\_position AS ‘POSITION’,

fa.fa\_salary AS ‘SALARY’

FROM FLIGHT\_ATTENDANT\_T fa

WHERE NOT (fa.fa\_salary = 2800 OR fa.fa\_salary = 3500)

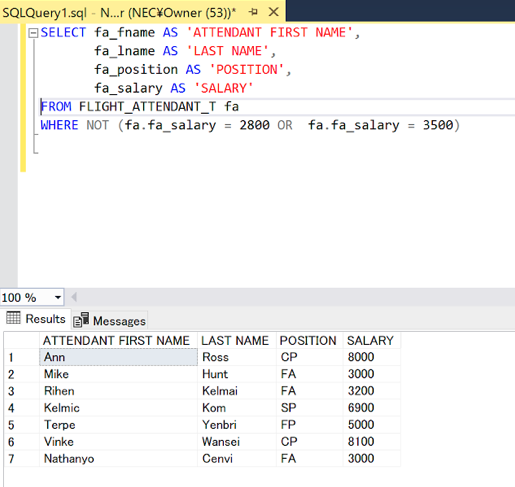
 **SQL Server Result**

Figure 3.11 Question 2 Student 4 SQL Server Result

1. List the booking’s date, flight number, and the client’s last name where the client does not have any outstanding balance.

**SQL DML CODE**

SELECT fb.fb\_book\_date AS ‘BOOKING DATE’,

fb.fli\_num\_pk AS ‘FLIGHT NUMBER’,

c.cust\_lname AS ‘CUSTOMER LAST NAME’

FROM FLIGHT\_BOOKING\_T fb, CUSTOMER\_T c

WHERE ( ( fb.cust\_id\_pk = c.cust\_id\_pk ) AND

( fb.fb\_outstanding IS NULL ) )

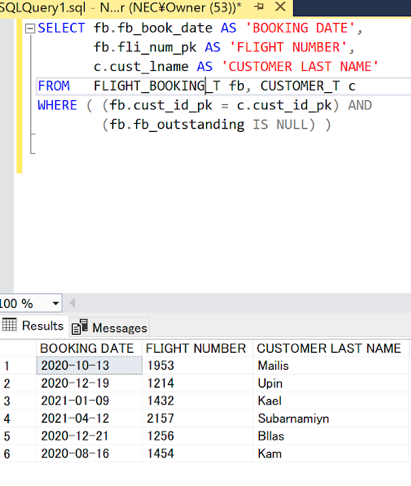
 **SQL Server Result**

Figure 3.12 Question 3 Student 4 SQL Server Result

# CONCLUSION

This assignment was a continuation from the part 1 of the IDB assignment. Just like the first part of the assignment documentation that the team has already done, the team also decides to divide the work equally between the team members. This documentation assignment is made possible with the utilisation of Microsoft Word as the compiler media and Microsoft SQL Server Management Studio as the base of the practical SQL codes and databases.

In doing this part 2 IDB assignment, the group leader, Mr. Kevin Matthew and the team members, Mr. Marcell Agung, Mr. Selvan Nicholas and Mr. Wataru Shinzato worked cooperatively and coordinated the process well to eventually produce this assignment documentation. Additionally, the IDB team’s lecturer and advisor, Mr. Dhason Padmakumar, also helped a lot in guiding and assisting the team in producing this documentation. Without the guidance from Mr. Dhason and the cooperation from the team, this assignment would not have been made possible.

Through this assignment, the team’s knowledge of databases has widened even more. The team learned a lot of database knowhow and expertise that will help them achieve a professional industry-world skillset. Learning how to use a Database Management System, inserting data to tables, writing clean SQL codes and mastering how to apply SQL to produce required data results really has made the team become proficient in managing databases.

The skills that are mentioned above was used by the team to put together a clean and detailed database regarding the Malaysian Airlines Reservation as aforementioned in part 1 of the assignment. The database was constructed in the Microsoft SQL Server Management Studio, which is a Database Management System (DBMS) platform that the team are familiar with. The constructed database is complex yet detailed and clear, making the constructed database a crucial aspect in the Malaysian Airlines Reservation System.

Nonetheless, the team members have put in everything they got and have successfully constructed this thorough database. All the team members have done their part in doing the tasks and completed their tasks well. There are no conflicts and misunderstanding in the process of doing this part of the assignment and everything went well. The teamwork between the team members also improved significantly than the first part of the assignment. To summarize, the team felt that doing this assignment is a meaningful experience and will further improve their skillset in creating and managing databases. The results are also satisfactory to the team and the team hoped to apply the vast amount of knowledge gained at the workplace in the future.

# WORKLOAD MATRIX

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
|
| **TASK ID** | **TASK NAME** | **DURATION** | **START DATE** | **END DATE** | **MEMBER(S) CONTRIBUTED** |
| 1.0 | **DATABASE SCHEMA (STRUCTURE OF DATABASE TABLES / DATA DICTIONARY)** |  | | | |
|  | 1.1Database Schema | 4 days | Wed, 19-May-2021 | Sat, 22-May-2021 | Mr. KEVIN MATTHEW ADYAN (TP058466)    Mr. MARCELL AGUNG WAHYUDI (TP058650) |
|  | 1.2Finalized ERD (Entity Relationship Diagram) – Output of Conceptual database design phase | 1 day | Thu, 3-June-2021 | Thu, 3-June-2021 | Mr. KEVIN MATTHEW ADYAN (TP058466) |
|  | 1.3Finalized EERD (Enhanced Entity Relationship Diagram) – Output of Logical database design phase | 1 day | Thu, 3-June-2021 | Thu, 3-June-2021 | Mr. KEVIN MATTHEW ADYAN (TP058466) |
|  | 1.4 Database diagram generated from the MS SQL Server 2019 | 2 hours | Sun, 13-June-2021 | Sun, 13-June-2021 | Mr. SELVAN NICHOLAS (TP058084) |
| 2.0 | **SQL-DATA DEFINITION LANGUAGE (DDL)** |  | | | |
|  | 2.1 SQL DDL Used | 2 days | Saturday, 12-June-2021 | Monday, 14-June-2021 | Mr. MARCELL AGUNG WAHYUDI (TP058650) |
|  | 2.2 Samples of Data Used | 5 days | Friday, 11-June-2021 | Wednesday, 16-June-2021 | Mr. MARCELL AGUNG WAHYUDI (TP058650) |
| 3.0 | **SQL-DATA MANIPULATION LANGUAGE (DML)** |  | | | |
|  | 3.1 SQL DML Used | 2 days | Friday,  18-June-  2021 | Thursday,  19-June-  2021 | Mr. WATARU SHINZATO (TP061217) |
|  | 3.2 SQL Questions |  | | | |
|  | 3.2.1 Question 1 | 0.5 Hour | Thursday,  1-July-2021 | Thursday,  1-July-2021 | Mr. MARCELL AGUNG WAHYUDI (TP058650) |
|  | 3.2.2 Question 2 | 1 Hour | Saturday, 26-June-2021 | Saturday, 26-June-2021 | Mr. SELVAN NICHOLAS (TP058084) |
|  | 3.2.3 Question 3 | 1.5 hour | Thursday, 17-June-2021 | Thursday, 17-June-2021 | Mr. KEVIN MATTHEW ADYAN (TP058466) |
|  | 3.2.4 Question 4 | 2 days | Saturday,  19-June-  2021 | Sunday,  20-June-  2021 | Mr. WATARU SHINZATO  (TP061217) |
| 4.0 | **CONCLUSION** | 1 hour | Saturday,  26-June-2021 | Saturday,  26-June-2021 | Mr. SELVAN NICHOLAS  (TP058084) |
| 5.0 | **WORKLOAD MATRIX** | 1 hour | Saturday, 26-June-2021 | Saturday, 26-June-2021 | Mr. KEVIN MATTHEW ADYAN (TP058466) |
| 6.0 | **REFERENCES** | 0.5 Hour | Saturday, 26-June-2021 | Saturday, 26-June-2021 | Mr. KEVIN MATTHEW ADYAN (TP058466) |

# REFERENCES

Microsoft, 2012. *FOREIGN KEY Constraints.* [Online]   
Available at: https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms175464(v=sql.105)?redirectedfrom=MSDN  
[Accessed 28 May 2021].

Riggs, R., 2013. *SQL Server: Creating a database diagram.* [Online]   
Available at: https://www.youtube.com/watch?v=N6PhiiXJMeU  
[Accessed 26 June 2021].