SCHOOL OF SCIENCE AND TECHNOLOGY

COURSEWORK FOR
BSC (HONS) IN COMPUTER SCIENCE
BSC (HONS) INFORMATION SYSTEMS
BSC (HONS) INFORMATION TECHNOLOGY
BACHELOR OF SOFTWARE ENGINEERING (HONS)
BSC (HONS) INFORMATION SYSTEMS (BUSINESS ANALYTICS)
BIS (HONS) IN MOBILE COMPUTING WITH ENTREPRENEURSHIP
BSC (HONS) INFORMATION TECHNOLOGY (COMPUTER NETWORKING AND SECURITY)

YEAR 1; ACADEMIC SESSION AUGUST 2020

SEG1201: DATABASE FUNDAMENTALS **DEADLINE: 22 November 2020, 1pm**

STUDENT NAMES: _____ _____
STUDENT IDS: _____ _____

---

i) This final assessment (Assignment 2) contributes 50% to your final grade.
ii) This two-member group assignment is for Course Learning Outcome 2 - Implement a database design group project using appropriate tools such as Oracle SQL

IMPORTANT

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

- Coursework submitted after the deadline but within 1 week will be accepted for a maximum mark of 40%.

- Work handed in following the extension of 1 week after the original deadline will be regarded as a non-submission and marked zero.

Lecturer Remark (Use additional sheet if required)

I..... (Name) .....std. ID received the assignment and read the

comments..... (Signature/date)

Academic Honesty Acknowledgement

"We ....., ……..... verify that this paper contains entirely our own work. We have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, we have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. We realize the penalties *(refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme)* for any kind of copying or collaboration on any assignment."

....., ……..... (Student signatures / Date)

Contents

Part 1: Design the database

i) Attributes and their data type

Table ‘MEMBER’’

Attribute name	Attribute description	Data type	Justification
MemberID	PK of table ‘MEMBER’	NUMBER	Numbers can be incremented
Name	Name of a member	VARCHAR(40)	Names might have symbols
Gender	Gender of a member	CHAR(1)	Either F or M
IC	IC number of a member	Number(12)	IC has 12 digits
Email	Email Address of member	VARCHAR2(40)	Email has symbols
ContactNumber	Phone number of member	NUMBER(10)	Malaysian phone number consist of 10 digits

Table ‘BOOKING’

Attribute name	Attribute description	Data type	Justification
BookingID	PK of table ‘BOOKING’	NUMBER	Numbers can be incremented
MemberID	FK : From table ‘MEMBER’	-	-
RoomID	FK : From table ‘ROOM’	-	-
CheckIn	Check In date	DATE	Date format
CheckOut	Checkout date	DATE	Date format
SpecialRequest	Special requests made by member upon booking	VARCHAR2(100)	Contains multiple sentences
BookingStatus	Status of the booking, whether it got canceled or not	CHAR(1)	True or false values can be assigned, T or F

CancellationDate	Date of the booking got canceled	DATE	Date format
Reason	Reason for canceling booking	VARCHAR2(100)	Contains multiple sentences

Table ‘ROOM’

Attribute name	Attribute description	Data type	Justification
RoomNumber	PK of table ‘ROOM’	NUMBER(3)	5 hotels * 25 rooms each = 125 rooms
BranchID	FK : From table ‘HOTELBRANCH’	-	-
TypeID	FK : From table ‘ROOMTYPE’	-	-
DatesAvailable	Dates when the room is available	DATE	Date format
DaysAvailable	??	??	

Table ‘ROOMTYPE’

Attribute name	Attribute description	Data type	Justification
TypeID	PK : of table ‘ROOMTYPE’	NUMBER(1)	There are only 4 room types
RoomName	Name of Room Type	VARCHAR2(30)	Room Type names Eg: Suite, Cabana
RoomRate	Rate of Room Type	NUMBER(*,2)	Number with 2 decimals, Money

Table ‘HOTELBRANCH’

Attribute name	Attribute description	Data type	Justification

BranchID	PK of table ‘HOTELBRANCH’	NUMBER(1)	There are only 5 branches
BranchName	Name of hotel branch	VARCHAR2(40)	Name of hotel branch has “spaces” eg:Royal Belum

Table ‘MAINTENANCE’

Attribute name	Attribute description	Data type	Justification
MaintenanceID	PK of table ‘MAINTENANCE’	NUMBER	Number is incrementable
RoomNumber	FK : From table ‘Room’	-	-
MaintenanceDate	Maintenance date	DATE	Date format
RepairReason	Reason of repair	VARCHAR(100)	Contains symbols and possible multiple sentences

Table ‘MAINTENANCE_TASK’

Attribute name	Attribute description	Data type	Justification
MaintenanceID	FK : From table ‘MAINTENANCE’	-	-
TaskID	FK : From table ‘Task’	-	-

Table ‘CONDITIONCHECK’

Attribute name	Attribute description	Data type	Justification
CheckDate	PK of table ‘CONDITIONCHECK’	DATE	-
RoomNumber	FK : From table (Room)	-	-
RoomCondition	Condition of Room at time of inspection	CHAR(1)	RoomCondition can be in 1 character like S for superb, E for excellent, G for good, P for poor condition

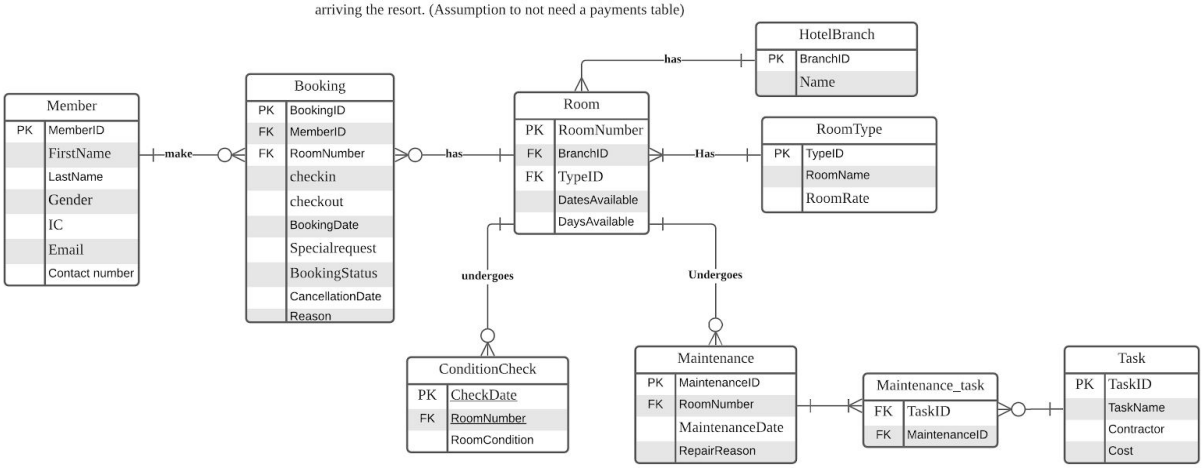
Table ‘TASK’

Attribute name	Attribute description	Data type	Justification
TaskID	FK for table”Task”	NUMBER	Number is incrementable
TaskName	Name of tasks	CHAR(20)	Short, single-word names of task
Contractor	Contractor’s company name	VARCHAR(50)	Might have characters
Cost	Cost of task	NUMBER(*,2)	Number with 2 decimal place

ii) ERD diagram

Additional Assumption: Small booking fee (RM100 flat) collected prior. Guest pays the full sum when arriving the resort. (Assumption to not need a payments table)

Small booking fee (RM100 flat) collected prior. Guest pays the full sum when arriving the resort. (Assumption to not need a payments table)



iii) Relational Database Model

Member *MemberID-pk* *FirstName* *LastName* *Gender* *IC* *Email* *ContactNumber*	***Booking*** *BookingID-pk* *MemberID* *RoomNumber* *CheckIn* *CheckOut* *SpecialRequest* *BookingStatus* *CancellationDate* *Reason* *FOREIGN KEY MemberID* *REFERENCES TABLE* ***Member***
Room *RoomNumber-pk* *BranchID* *TypeID* *FOREIGN KEY BranchID* *REFERENCES TABLE* ***HotelBranch*** *FOREIGN KEY TypeID REFERENCES* *TABLE* ***RoomType***	***HotelBranch*** *BranchID-pk* *Name*
RoomType *TypeID-pk* *RoomName* *RoomRate*	***ConditionCheck*** *Month-pk* *Year-pk* *RoomNumber -pk* *RoomCondition* *FOREIGN KEY RoomNumber* *REFERENCES TABLE* ***Room***

Maintenance *MaintenanceID-pk* *RoomNumber* *Date* *RepairReason* *FOREIGN KEY RoomNumber* *REFERENCES TABLE* ***Room***	***Maintenance_task*** *TaskID* *MaintenanceID* *FOREIGN KEY TaskID* *REFERENCES TABLE* ***Task*** *FOREIGN KEY MaintenanceID* *REFERENCES TABLE Maintenance*
Task *TaskID-pk* *TaskName* *Contractor* *Cost*	

Part 2: Implement the database

resort.txt script listing

```
DROP TABLE MEMBER cascade constraints;
DROP TABLE BOOKING cascade constraints;
DROP TABLE ROOM cascade constraints;
DROP TABLE ROOMTYPE cascade constraints;
DROP TABLE HOTELBRANCH cascade constraints;
DROP TABLE CONDITIONCHECK cascade constraints;
DROP TABLE MAINTENANCE cascade constraints;
DROP TABLE MAINTENANCE_TASK cascade constraints;
DROP TABLE TASK cascade constraints;
Drop sequence member_ID_seq;
Drop TRIGGER trg_member_id;
Drop sequence BookingID_seq;
Drop trigger trg_BookingID;
Drop sequence MaintenanceID_seq;
Drop trigger trg_MaintenanceID;
Drop sequence TaskID_seq;
Drop trigger trg_TaskID;
Drop trigger trg_check_checkin;

alter session set NLS_DATE_FORMAT='DD/MM/YYYY';
CREATE TABLE MEMBER
(
  MemberID          NUMBER PRIMARY KEY,
  FirstName         VARCHAR2(40) NOT NULL,
  LastName          VARCHAR2(30) NOT NULL,
  Gender            CHAR(1) NOT NULL,
  IC                VARCHAR2(40) UNIQUE,
  Email             VARCHAR2(40),
  ContactNumber     NUMBER(10)
 );

create sequence member_ID_seq;

CREATE TRIGGER trg_member_id
  before insert on MEMBER
  for each row
```

```
begin
  select member_id_seq.nextval
    into :new.MEMBERID
    from dual;
end;
/
CREATE TABLE TASK
(
   TaskID              NUMBER PRIMARY KEY,
   TaskName            VARCHAR(20) UNIQUE,
   Contractor          VARCHAR(50),
   Cost                NUMBER(*,2) CHECK(cost>0)
);
/* auto increment TaskID */
create sequence TaskID_seq start with 1;

CREATE TRIGGER trg_TaskID
  before insert on TASK
  for each row
begin
  select TaskID_seq.nextval
    into :new.TaskID
    from dual;
end;
/
CREATE TABLE ROOMTYPE
(
   TypeID              NUMBER(1) PRIMARY KEY,
   RoomName            VARCHAR2(30),
   RoomRate            NUMBER(*,2) NOT NULL CHECK(RoomRate >0 )
);

CREATE TABLE HOTELBRANCH
(
   BranchID            NUMBER(1) PRIMARY KEY,
   BranchName          VARCHAR2(40) UNIQUE
);


CREATE TABLE ROOM
```

```
(
   RoomNumber              NUMBER(3) PRIMARY KEY,
 BranchID                  NUMBER(1),
TypeID                NUMBER(1),
    FOREIGN KEY (BranchID) references HOTELBRANCH(BranchID),
FOREIGN KEY (TypeID) references ROOMTYPE(TypeID)
);

CREATE TABLE BOOKING
(
  BookingID               NUMBER(4) PRIMARY KEY,
  MemberID                NUMBER(3),
  RoomNumber                    NUMBER(3),
  CheckIn                 DATE NOT NULL,
  CheckOut                DATE NOT NULL,
  BookingDate             DATE NOT NULL,
  SpecialRequest          VARCHAR(100),
  BookingStatus           CHAR(1),
  CancellationDate        DATE,
  Reason                  VARCHAR(100),
FOREIGN KEY (MemberID) references MEMBER(MemberID),
FOREIGN KEY (RoomNumber) references ROOM(RoomNumber)
);

/* checks checkin date to be greater than sysdate */
CREATE OR REPLACE TRIGGER trg_check_checkin
  BEFORE INSERT OR UPDATE ON BOOKING
  FOR EACH ROW
BEGIN
  IF( :new.checkin <= SYSDATE )
  THEN
    RAISE_APPLICATION_ERROR( -20001,
        'Invalid Checkin Date: Checkin date must be greater than the current date - value
= ' ||
        to_char( :new.checkin, 'YYYY-MM-DD HH24:MI:SS' ) );
  END IF;
  IF( :new.checkin > add_months(SYSDATE,12) )
  THEN
    RAISE_APPLICATION_ERROR( -20002,
        'Invalid Checkin date: CheckinDate must be within the next year - value = ' ||
```

```
        to_char( :new.Checkin, 'YYYY-MM-DD HH24:MI:SS' ) );
  END IF;
END;
/


CREATE TABLE CONDITIONCHECK
(
   CheckDate              DATE default sysdate,
   RoomNumber             NUMBER(3),
   RoomCondition          CHAR(1),
FOREIGN KEY (RoomNumber) references ROOM(RoomNumber),
CONSTRAINT PK_CONDITIONCHECK PRIMARY KEY (CheckDate,RoomNumber)
);

CREATE TABLE MAINTENANCE
(
   MaintenanceID          NUMBER PRIMARY KEY,
   RoomNumber             NUMBER(3),
   MaintenanceDate        DATE NOT NULL,
   RepairReason           VARCHAR(100),
FOREIGN KEY (RoomNumber) references ROOM(RoomNumber)
);

/* auto increment MaintenanceID */
create sequence MaintenanceID_seq start with 1;

CREATE TRIGGER trg_MaintenanceID
  before insert on Maintenance
  for each row
begin
  select MaintenanceID_seq.nextval
    into :new.MaintenanceID
    from dual;
End;
/


CREATE TABLE MAINTENANCE_TASK
(
```

```
  MaintenanceID     NUMBER,
  TaskID            NUMBER,
FOREIGN KEY (MaintenanceID) references MAINTENANCE(MaintenanceID),
FOREIGN KEY (TaskID) references TASK(TaskID)
);
```

```
/* Populate Table Member */
/* Example for member*/
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Tow Wang', 'Chiam', 'M', 880101148888, 'chiamtw95@gmail.com',
013608478);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Michael', 'Chang', 'M', 750519159999, 'mchang1@gmail.com', 015777455);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Joseph', 'Won', 'M', 790519149990, 'jwon3@gmail.com', 015911852);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Beneath', 'Sixfeet', 'M', 421212145420, 'b6feet@gmail.com', 019501658);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Agatan', 'Seek', 'M', 701010121818, 'aseek4@gmail.com', 017144186);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Jera', 'Keri', 'M', 691020052020,'jerake32@gmail.com', 019501658);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Anthony', 'Liu', 'M', 800808080808, 'aliu123@gmail.com', 013805469);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Crack', 'Head', 'M', 900909090909, 'cracklover1@gmail.com', 013576116);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Christian', 'Wood', 'M', 7007070117070, 'chwood1@gmail.com', 011895671);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Alex', 'Tang', 'M', 810409092121, 'atang1@gmail.com', 012705506);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Pepe', 'Hans', 'M', 590921107839, 'pepehands@gmail.com', 018424426);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Kek', 'Wong', 'M', 880808104959, 'kekw@gmail.com', 014779839);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Peanut', 'Butter', 'M', 970320165214, 'peanutbutter@gmail.com',
019267347);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Sonnof', 'Abith', 'M', 920615115839, 'sob300@gmail.com', 013445716);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Ping', 'Pong', 'M', 850509145671, 'pingpong@gmail.com', 012804134);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Gerald', 'Gibraltar', 'M', 900329062341, 'gegibraltar@gmail.com',
013409975);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Holdup', 'Mang', 'M', 861224048327, 'holup1@gmail.com', 017893174);
```

```
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Thyssa', 'Roberi', 'F', 960521057727, 'thisarobery1@gmail.com',
017362669);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Putyo', 'Handsup', 'F', 831015182152, 'putyourhandsup@gmail.com',
017021877);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Kimi', 'Demoney', 'F', 940819107737, 'gimmemoney1@gmail.com',
019031782);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Samhwan', 'Getinhurt', 'F', 820528105692, 'ohno342@gmail.com',
016744917);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Hannah', 'Jo', 'F', 740521057372, 'hannahj@gmail.com', 011875492);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Serena', 'Williams', 'F', 900101145929, 'swilliams@gmail.com', 019424295);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Amana', 'Whoa', 'F', 650415205030, 'amanaw@gmail.com', 019178684);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Simp', 'King', 'F', 600624126215, 'simpking@gmail.com', 012401560);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Tera', 'Byte', 'F', 700225196349, 'terabyte5@gmail.com', 012512469);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Tera', 'Ble', 'F', 960315106340, 'terrible14@gmail.com', 013065324);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Puchong', 'Uptown', 'F', 921220129191, 'puchongcity@gmail.com',
1234567);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Loraine', 'Cheese', 'F', 900601102953, 'lecheese@gmail.com', 013357881);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Mike', 'Hawk', 'F', 800718106969, 'mikehawk1@gmail.com', 012821068);
INSERT INTO MEMBER(FirstName, LastName, Gender, IC, Email, ContactNumber)
VALUES ( 'Celine', 'Tam', 'F', 990509145671, 'ctam99@gmail.com', 0126871068);
```

```
/* Populate Table Task, 9 tasks created */
INSERT INTO TASK(TaskName, Contractor, Cost) VALUES ('Cleaning', 'Best Cleaning Company Sdn. Bhd', 300);
INSERT INTO TASK(TaskName, Contractor, Cost) VALUES ('ToiletRepair', 'Toilet Company Sdn. Bhd', 200);
INSERT INTO TASK(TaskName, Contractor, Cost) VALUES ('Aircon', 'Aircon Company Sdn. Bhd', 600);
INSERT INTO TASK(TaskName, Contractor, Cost) VALUES ('Window', 'Window Company Sdn. Bhd', 700);
INSERT INTO TASK(TaskName, Contractor, Cost) VALUES ('Lighting', 'Lighting Company Sdn. Bhd', 600);
INSERT INTO TASK(TaskName, Contractor, Cost) VALUES ('Building', 'Construction Company Sdn. Bhd', 1000);
INSERT INTO TASK(TaskName, Contractor, Cost) VALUES ('Pipe', 'Pipe Company Sdn. Bhd', 110);
INSERT INTO TASK(TaskName, Contractor, Cost) VALUES ('Door', 'Door Systems Company Sdn. Bhd', 2000);
INSERT INTO TASK(TaskName, Contractor, Cost) VALUES ('Pest', 'Pest Control Company Sdn. Bhd', 650);
```

```
/* Populate Table ROOMTYPE */

INSERT INTO ROOMTYPE VALUES( 1, 'Suite', 300);
INSERT INTO ROOMTYPE VALUES( 2, 'Cabana', 400);
INSERT INTO ROOMTYPE VALUES( 3, 'Studio', 500);
INSERT INTO ROOMTYPE VALUES( 4, 'Lanai', 600);
```

```
/* Populate Table HOTELBRANCH */
/*
The resort has five branches,
situated in Ipoh, Redang, Royal Belum and Endau Rompin, Langkawi
*/
INSERT INTO HOTELBRANCH VALUES( 1, 'Ipoh');
INSERT INTO HOTELBRANCH VALUES( 2, 'Redang');
INSERT INTO HOTELBRANCH VALUES( 3, 'Royal Belum');
INSERT INTO HOTELBRANCH VALUES( 4, 'Endau Rompin');
INSERT INTO HOTELBRANCH VALUES( 5, 'Langkawi');
```

```
/*Populate Room table
roomid 1-125 rooomtype:1-4 branchID:1-5
*/
/* branch 1  */
INSERT INTO ROOM VALUES (1,1,1);
INSERT INTO ROOM VALUES (2,1,1);
INSERT INTO ROOM VALUES (3,1,1);
INSERT INTO ROOM VALUES (4,1,1);
INSERT INTO ROOM VALUES (5,1,1);
INSERT INTO ROOM VALUES (6,1,1);
INSERT INTO ROOM VALUES (7,1,1);
INSERT INTO ROOM VALUES (8,1,1);
INSERT INTO ROOM VALUES (9,1,1);
INSERT INTO ROOM VALUES (10,1,1);
INSERT INTO ROOM VALUES (11,1,2);
INSERT INTO ROOM VALUES (12,1,2);
INSERT INTO ROOM VALUES (13,1,2);
INSERT INTO ROOM VALUES (14,1,2);
INSERT INTO ROOM VALUES (15,1,2);
INSERT INTO ROOM VALUES (16,1,2);
INSERT INTO ROOM VALUES (17,1,2);
INSERT INTO ROOM VALUES (18,1,2);
INSERT INTO ROOM VALUES (19,1,2);
INSERT INTO ROOM VALUES (20,1,3);
INSERT INTO ROOM VALUES (21,1,3);
INSERT INTO ROOM VALUES (22,1,3);
INSERT INTO ROOM VALUES (23,1,4);
INSERT INTO ROOM VALUES (24,1,4);
INSERT INTO ROOM VALUES (25,1,4);

/* branch 2  */
INSERT INTO ROOM VALUES (26,2,1);
INSERT INTO ROOM VALUES (27,2,2);
INSERT INTO ROOM VALUES (28,2,2);
INSERT INTO ROOM VALUES (29,2,2);
INSERT INTO ROOM VALUES (30,2,2);
INSERT INTO ROOM VALUES (31,2,2);
INSERT INTO ROOM VALUES (32,2,2);
INSERT INTO ROOM VALUES (33,2,2);
```

```
INSERT INTO ROOM VALUES (34,2,2);
INSERT INTO ROOM VALUES (35,2,2);
INSERT INTO ROOM VALUES (36,2,2);
INSERT INTO ROOM VALUES (37,2,2);
INSERT INTO ROOM VALUES (38,2,2);
INSERT INTO ROOM VALUES (39,2,2);
INSERT INTO ROOM VALUES (40,2,2);
INSERT INTO ROOM VALUES (41,2,2);
INSERT INTO ROOM VALUES (42,2,2);
INSERT INTO ROOM VALUES (43,2,3);
INSERT INTO ROOM VALUES (44,2,3);
INSERT INTO ROOM VALUES (45,2,4);
INSERT INTO ROOM VALUES (46,2,4);
INSERT INTO ROOM VALUES (47,2,4);
INSERT INTO ROOM VALUES (48,2,4);
INSERT INTO ROOM VALUES (49,2,4);
INSERT INTO ROOM VALUES (50,2,4);


/* Branch 3*/
INSERT INTO ROOM VALUES (51,3,1);
INSERT INTO ROOM VALUES (52,3,1);
INSERT INTO ROOM VALUES (53 ,3,1);
INSERT INTO ROOM VALUES (54,3,1);
INSERT INTO ROOM VALUES (55,3,1);
INSERT INTO ROOM VALUES (56,3,1);
INSERT INTO ROOM VALUES (57,3,1);
INSERT INTO ROOM VALUES (58,3,1);
INSERT INTO ROOM VALUES (59,3,1);
INSERT INTO ROOM VALUES (60,3,1);
INSERT INTO ROOM VALUES (61,3,2);
INSERT INTO ROOM VALUES (62,3,2);
INSERT INTO ROOM VALUES (63,3,2);
INSERT INTO ROOM VALUES (64,3,2);
INSERT INTO ROOM VALUES (65,3,3);
INSERT INTO ROOM VALUES (66,3,3);
INSERT INTO ROOM VALUES (67,3,3);
INSERT INTO ROOM VALUES (68,3,3);
INSERT INTO ROOM VALUES (69,3,3);
```

```
INSERT INTO ROOM VALUES (70,3,3);
INSERT INTO ROOM VALUES (71,3,4);
INSERT INTO ROOM VALUES (72,3,4);
INSERT INTO ROOM VALUES (73,3,4);
INSERT INTO ROOM VALUES (74,3,4);
INSERT INTO ROOM VALUES (75,3,4);

/* Branch 4*/
INSERT INTO ROOM VALUES (76,4,1);
INSERT INTO ROOM VALUES (77,4,1);
INSERT INTO ROOM VALUES (78,4,1);
INSERT INTO ROOM VALUES (79,4,1);
INSERT INTO ROOM VALUES (80,4,1);
INSERT INTO ROOM VALUES (81,4,1);
INSERT INTO ROOM VALUES (82,4,1);
INSERT INTO ROOM VALUES (83,4,1);
INSERT INTO ROOM VALUES (84,4,1);
INSERT INTO ROOM VALUES (85,4,1);
INSERT INTO ROOM VALUES (86,4,2);
INSERT INTO ROOM VALUES (87,4,2);
INSERT INTO ROOM VALUES (88,4,2);
INSERT INTO ROOM VALUES (89,4,2);
INSERT INTO ROOM VALUES (90,4,3);
INSERT INTO ROOM VALUES (91,4,3);
INSERT INTO ROOM VALUES (92,4,3);
INSERT INTO ROOM VALUES (93,4,3);
INSERT INTO ROOM VALUES (94,4,3);
INSERT INTO ROOM VALUES (94,4,3);
INSERT INTO ROOM VALUES (96,4,4);
INSERT INTO ROOM VALUES (97,4,4);
INSERT INTO ROOM VALUES (98,4,4);
INSERT INTO ROOM VALUES (99,4,4);
INSERT INTO ROOM VALUES (100,4,4);

/* Branch 5*/
INSERT INTO ROOM VALUES (101,5,1);
INSERT INTO ROOM VALUES (102,5,1);
INSERT INTO ROOM VALUES (103,5,1);
INSERT INTO ROOM VALUES (104,5,1);
INSERT INTO ROOM VALUES (105,5,1);
```

```
INSERT INTO ROOM VALUES (106,5,1);
INSERT INTO ROOM VALUES (107,5,1);
INSERT INTO ROOM VALUES (108,5,1);
INSERT INTO ROOM VALUES (109,5,1);
INSERT INTO ROOM VALUES (110,5,1);
INSERT INTO ROOM VALUES (111,5,2);
INSERT INTO ROOM VALUES (112,5,2);
INSERT INTO ROOM VALUES (113,5,2);
INSERT INTO ROOM VALUES (114,5,2);
INSERT INTO ROOM VALUES (115,5,3);
INSERT INTO ROOM VALUES (116,5,3);
INSERT INTO ROOM VALUES (117,5,3);
INSERT INTO ROOM VALUES (118,5,3);
INSERT INTO ROOM VALUES (119,5,3);
INSERT INTO ROOM VALUES (120,5,3);
INSERT INTO ROOM VALUES (121,5,4);
INSERT INTO ROOM VALUES (122,5,4);
INSERT INTO ROOM VALUES (123,5,4);
INSERT INTO ROOM VALUES (124,5,4);
INSERT INTO ROOM VALUES (125,5,4);
```

/* Populate Table BOOKING */
/* update syntax if there is cancellation to be updated

```
UPDATE
    parts
SET
    cost = 130
WHERE
    part_id = 1; */
```

INSERT INTO BOOKING VALUES
(60,31,5,'1/12/2020','6/12/2020','3/10/2020',NULL,NULL,NULL,NULL);
INSERT INTO BOOKING VALUES
(61,31,5,'10/12/2020','12/12/2020','20/2/2020',NULL,NULL,NULL,NULL);
INSERT INTO BOOKING VALUES
(62,31,5,'25/11/2020','27/11/2020','16/7/2021',NULL,'F','1/10/2020','Travel Restrictions');
INSERT INTO BOOKING VALUES
(63,31,5,'20/12/2020','22/12/2020','17/8/2020',NULL,'F','1/11/2020','Travel Restrictions');

SELECT BookingID,BookingDate,BookingStatus,CancellationDate,Reason
FROM BOOKING
INNER JOIN MEMBER
ON BOOKING.memberid = MEMBER.memberid
WHERE MEMBER.FirstName ='Celine' AND MEMBER.LastName = 'Tam'
AND BOOKING.BookingDate > '1/3/2020'
ORDER BY BookingDate DESC

/*Not Celine, same dates as celine*/
INSERT INTO BOOKING VALUES
(64,20,100,'20/12/2021','22/12/2021','17/8/2021',NULL,NULL,NULL,NULL);
(65,21,101,'20/12/2021','22/12/2021','17/8/2021',NULL,NULL,NULL,NULL);
(66,22,102,'20/12/2021','22/12/2021','17/8/2021',NULL,NULL,NULL,NULL);
(67,23,103,'20/12/2021','22/12/2021','17/8/2021',NULL,NULL,NULL,NULL);
(68,24,104,'20/12/2021','22/12/2021','17/8/2021',NULL,NULL,NULL,NULL);

```
/* Populate Table CONDITIONCHECK */
/* S for superb, E for excellent, G for good, P for poor */
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (1, 'S');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (2, 'E');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (3, 'G');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (4, 'P');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (5, 'S');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (6, 'S');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (7, 'S');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (8, 'S');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (9, 'E');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (10, 'G');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (11, 'P');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (12, 'S');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (13, 'S');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (14, 'S');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (15, 'G');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (16, 'P');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (17, 'S');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (18, 'S');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (19, 'S');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (20, 'G');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (21, 'P');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (22, 'S');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (23, 'S');
INSERT INTO  CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (24, 'S');
INSERT INTO CONDITIONCHECK (RoomNumber,RoomCondition) VALUES (25, 'S');
```

```
/* Populate Table MAINTENANCE
TaskID ranges from 1-9
RoomNumber from 1-125
*/
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (1, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (1, '3/1/2020', 'Leaking Pipe');
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (1, '5/1/2020', 'Aircon Faulty');
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (4, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (5, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (6, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (7, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (8, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (9, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (10, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (25, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (26, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (27, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (28, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (29, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (51, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (52, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (53, '1/1/2020', NULL);
```

```
INSERT INTO  MAINTENANCE(RoomNumber, MaintenanceDate, RepairReason)
VALUES (54, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (55, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (56, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (57, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (58, '1/1/2020', NULL);
INSERT INTO  MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (59, '1/1/2020', NULL);
INSERT INTO MAINTENANCE (RoomNumber, MaintenanceDate, RepairReason)
VALUES (60, '1/1/2020', NULL);
```

Part 3: Query the database

No	Query Description
i)	List the bookings made by guest *Celine Tam* since March 2020. Include also her cancelled bookings. Sort the report by booking date in descending order.
Insert **a) Sql query** **b) a legible screenshot with WHITE background and with at least font size of 10**	
ii)	Produce a report to show the room which has the highest maintenance cost for each branch. Write this query with no table joins (i.e., use subqueries).
Insert **a) Sql query** **b) a legible screenshot with WHITE background and with at least font size of 10**	
iii)	***Insert your question here***
Insert **a) Sql query** **b) a legible screenshot with WHITE background and with at least font size of 10**	

Part 4: Presentation

List each student's tasks here. Please change the questions accordingly.

No	Name	Questions to present
i)		Run batch script, Run the SQL codes Q??, etc

ii)		Run the SQL codes Q??, ad-hoc queries, etc
iii)		Explain the various keys in each table. Alternate keys are expected to be identified, etc
iv)		Explain anomalies
v)		Explain normalisation eg: which table has flaws, etc

Part 1: Design the database (30 marks)	
attributes and data types, and justification.	/5
ERD	/10
RDM in 3NF	/15
Part 2: Implement the database (25 marks)	
entity and referential integrities are correctly established.	/10
populate the database with quality data	/15
Part 3: Query the database (35 marks)	
i) List all the pet handler deals/activities from July 2019 to September 2019. Group them by the type of handler and sort the records by the handler's name in ascending order.	/3

ii) Using subqueries (without join of any kind), list the names of the pet owner who have booked and later cancelled their bookings two weeks prior to the booked date.	/3
iii) List the pet-sitters' name, age and their average rating for all the cats under their care. Sort the result based on the ratings with five-point rating to be displayed on the first row.	/3
iv) List the percentage of abuse incidences reported in year 2018.	/3
v) List the details of dog owner who has sent his/her dogs for either grooming or boarding during the month of June or August 2019.	/3
vi) Which type of insurance is the most popular amongst the pet owners?	/3
vii) Produce a report on the amount of money made by PetStreet through successful deals in year 2018. A deal is considered successful when the owner has made full payment and has used the service as requested. Where no deals were transacted for certain months, you need to display "NIL" under the profit column. The report needs to have the following heading:	/3
viii) Please replace this question with your question. Your question should be that of a scenario that warrants a SQL statement which requires 3 sub-queries (no join is expected in this question). Please do not create a scenario that uses the sub-queries for the sake of answering the question. In addition, ensure that your query yields at least two records.	/7
ix) Please replace this question with your question. Your question should be that of a scenario that warrants a SQL statement which requires 5-table join. Ensure that your question reflects the reason for such join and not join those tables for the sake of joining. In addition, ensure that your query yields at least two records	/7
Part 4: Presentation (10 marks)	
i) Run batch script ii) Run the SQL codes for Part 3 and explain the use of the command/keyword iii) Explain the various keys in each table. Alternate keys are expected to be identified iv) Explain Part 3 viii and ix. v) Explain how a view can be implemented in your database.	