Name: OMAR ISMAIL ABDJALEEL ALOMORY

Matric No: S63955

Date: 4/11/2022

Lab: MP3

Lecturer: Dr. ROSAIDA ROSLY

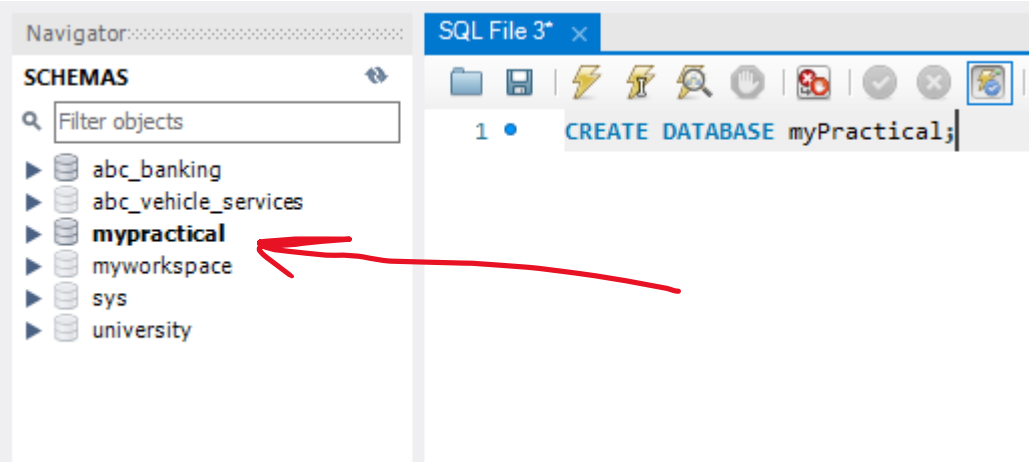
DATABASE (CSF3123) k2

LAB 1 – Installation and Configuration of MySQL, MySQL Workbench and Implement Query By Examples (QBE)

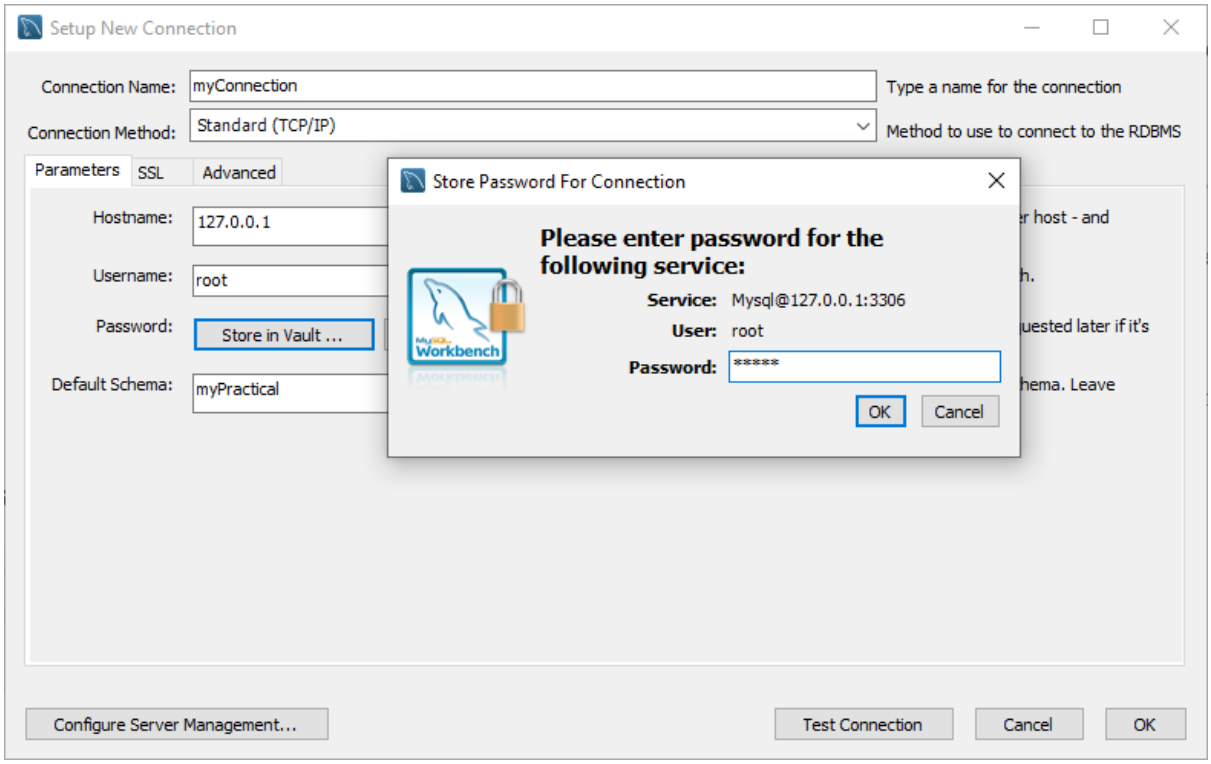
Activity 1

Step-by-step solution:

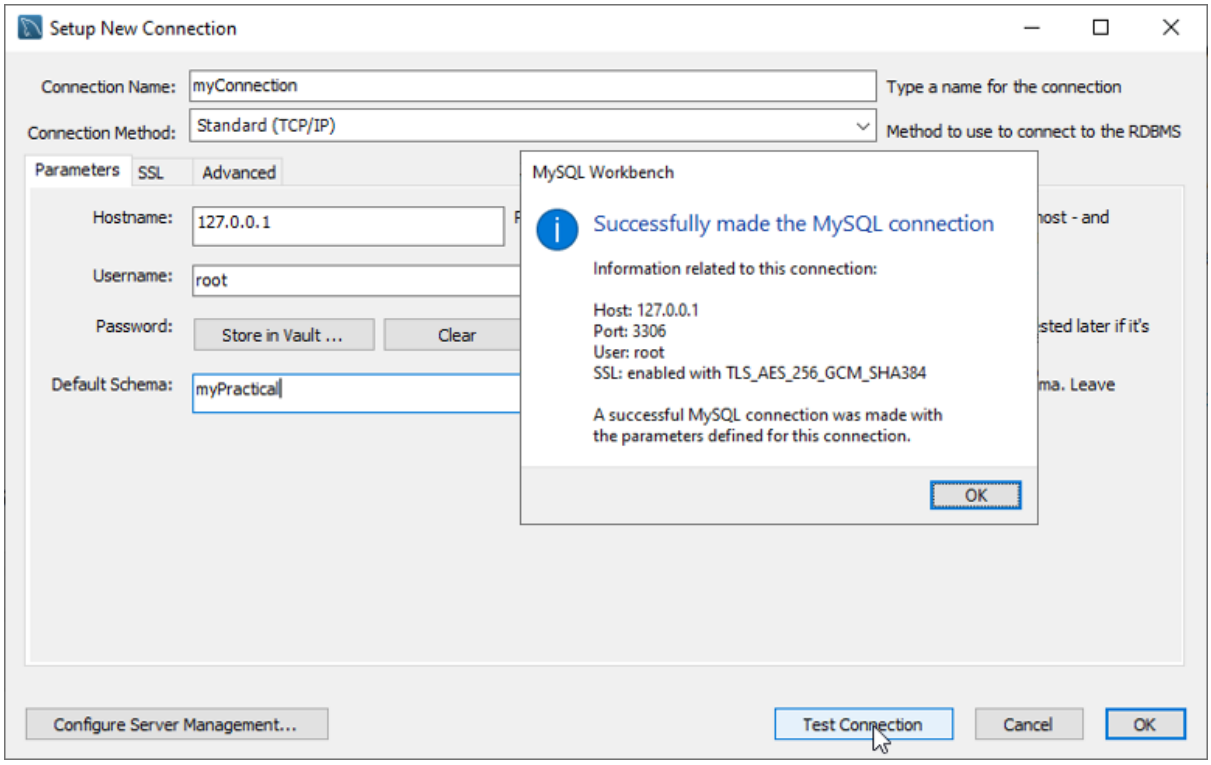
1. Creating database/schema



2. Creating a connection and attaching it to the database schema we created in step 1.



3. Testing the connection.



Activity 2

1. Define data dictionary for Table 1, 2 and 3 respectively. You should properly defined the identifier and data type that representing the attributes for each entity.

Table Name	Attribute Name	Content	Data Type	Format	Key	Required	FK Refernece Table	Notes
customer_table	custID	Customer ID	SMALLINT	#####	PRI			
	custName	Customer Name	varchar(50)	Xxxxx Xxxxx		Y		
	custJob	Customer Job	varchar(50)	Xxxxx Xxxxx OR Xxxxxx		Y		
	custAddr1	Customer Address 1	varchar(50)			Y		
	custAddr2	Customer Address 2	varchar(50)	Xxxxx Xxxxx		Y		
	custState	Customer State	varchar(20)			Y		
	custPostCode	Customer PostCode	SMALLINT	#####		Y		
	custContact	Customer Contact	varchar(12)			Y		
product_table	prdID	Product ID	char(10)	XX####.##	PRI			
	prdName	Product Name	varchar(45)			Y		
	prdCategory	Product Category	varchar(45)	Xxxxxx		Y		ex. Lubricant, Filter
	prdUnit	Product Unit	varchar(45)	Units		Y		ex. Box , Litre
	prdCostPrice	Product CostPrice	Number	####.##		Y		
	prdSalesPrice	Product SalesPrice	Number	####.##		Y		
sales_table	slsID	Sales ID	int	#####	PRI			
	custID	Customer ID	int	#####	FK	Y	customer_table	
	slsType	Sales Type	varchar(45)			Y		ex. Credit , Cash
	slsOrdDate	Sales Order Date	date	DD/MM/YYYY		Y		
	slsShipDate	Sales Ship Date	date	DD/MM/YYYY		Y		

2. By using SQL DDL command, create the structure of these tables (ignore any constraints).

```
-- Creating Customer Table
CREATE TABLE customer_table (
  custID int NOT NULL AUTO_INCREMENT COMMENT 'Customer ID',
  custName varchar(50) DEFAULT NULL COMMENT 'Customer Name',
  custJob varchar(50) DEFAULT NULL COMMENT 'Customer Jop',
  custAddr1 varchar(50) DEFAULT NULL COMMENT 'Customer Address 1',
  custAddr2 varchar(50) DEFAULT NULL COMMENT 'Customer Address 2',
  custState varchar(20) DEFAULT NULL COMMENT 'Customer State',
  custPostCode int DEFAULT NULL COMMENT 'Post Code',
  custContact varchar(12) DEFAULT NULL COMMENT 'Customer Contact',
  PRIMARY KEY (custID));
```

```sql
-- Creating Product Table
CREATE TABLE product_table (
  prdID VARCHAR(200) NOT NULL,
  prdName VARCHAR(45) NULL DEFAULT NULL,
  prdCategory VARCHAR(45) NULL DEFAULT NULL,
  prdUnit VARCHAR(45) NULL DEFAULT NULL,
  prdCostPrice DOUBLE NULL DEFAULT NULL,
  prdSalesPrice DOUBLE NULL DEFAULT NULL,
  PRIMARY KEY (prdID));
```

```sql
-- Creating Sales Table
CREATE TABLE sales_table (
  slsID int NOT NULL,
  custID int DEFAULT NULL,
  slsType varchar(45) DEFAULT NULL,
  slsOrdDate date DEFAULT NULL,
  slsShipDate date DEFAULT NULL,
  PRIMARY KEY (slsID),
  KEY custID_idx (custID),
  CONSTRAINT custID FOREIGN KEY (custID) REFERENCES customer_table (custID));
```

3. Finally, display the structure of table you created using SQL DDL commands.
4. Provide step by step solutions (with a diagram) on how you complete Activity 2.

First creating the database

```
CREATE DATABASE myPractical;
```

costumer_table

```
-- Creating Customer Table
CREATE TABLE customer_table (
  custID int NOT NULL AUTO_INCREMENT COMMENT 'Customer ID',
  custName varchar(50) DEFAULT NULL COMMENT 'Customer Name',
  custJob varchar(50) DEFAULT NULL COMMENT 'Customer Jop',
  custAddr1 varchar(50) DEFAULT NULL COMMENT 'Customer Address 1',
  custAddr2 varchar(50) DEFAULT NULL COMMENT 'Customer Address 2',
  custState varchar(20) DEFAULT NULL COMMENT 'Customer State',
  custPostCode int DEFAULT NULL COMMENT 'Post Code',
  custContact varchar(12) DEFAULT NULL COMMENT 'Customer Contact',
  PRIMARY KEY (custID));
```

```
-- insertion to customer table
insert into customer_table(custName,custJob, custAddr1, custAddr2, custState, custPostCode,custContact)
values ('Ahimd Roslan', 'Insurance Exec','no.15, jln Air Kolam','Kuala Terengganu','Terengganu' , 21060,'0112563030'),
('Chan Liew','Bank Officer','No. 1220, Prima Appartment','Mutiara Damansara','Kuala Lumpur',51200,'0123321919'),
       ('Mohed Sazali','Technician','No. 5, Lrg Melati 5/12','Tmn Sri Gombak', 'Selangor',48500,'0194451517');

-- run the table
```

Field	Type	Null	Key	Default	Extra
custID	int	NO	PRI	NULL	auto_increment
custName	varchar(50)	YES		NULL	
custJob	varchar(50)	YES		NULL	
custAddr1	varchar(50)	YES		NULL	
custAddr2	varchar(50)	YES		NULL	
custState	varchar(20)	YES		NULL	
custPostCode	int	YES		NULL	
custContact	varchar(12)	YES		NULL	

custID	custName	custJob	custAddr1	custAddr2	custState	custPostCode	custContact
10001	Ahimd Roslan	Insurance Exec	no.15, jln Air Kolam	Kuala Terengganu	Terengganu	21060	0112563030
10002	Chan Liew	Bank Officer	No. 1220, Prima Appartment	Mutiara Damansara	Kuala Lumpur	51200	0123321919
10003	Mohed Sazali	Technician	No. 5, Lrg Melati 5/12	Tmn Sri Gombak	Selangor	48500	0194451517

product_table

```
-- Creating Product Table
CREATE TABLE product_table (
  prdID VARCHAR(200) NOT NULL,
  prdName VARCHAR(45) NULL DEFAULT NULL,
  prdCategory VARCHAR(45) NULL DEFAULT NULL,
  prdUnit VARCHAR(45) NULL DEFAULT NULL,
  prdCostPrice DOUBLE NULL DEFAULT NULL,
  prdSalesPrice DOUBLE NULL DEFAULT NULL,
  PRIMARY KEY (prdID));

-- insertion to product table
insert into   product_table(prdID,prdName,prdCategory,prdUnit,prdCostPrice,prdSalesPrice)
values ('PW1001-01','Mach 20/50 Cylinder Oil','Lubricant','Litre',45.00,76.90),
       ('PW2001-01','Vios Oil Filter', 'Filter',   'box',  15.35,28.95),
       ('PW1001-45','Gear Oil 10/30','Lubricant','Litre',28.89,44.9);
```

Field	Type	Null	Key	Default
prdID	char(10)	NO	PRI	NULL
prdName	varchar(45)	YES		NULL
prdCategory	varchar(45)	YES		NULL
prdUnit	varchar(45)	YES		NULL
prdCostPrice	double	YES		NULL
prdSalesPrice	double	YES		NULL

prdID	prdName	prdCategory	prdUnit	prdCostPrice	prdSalesPrice
PW1001-01	Mach 20/50 Cylinder Oil	Lubricant	Litre	45	76.9
PW1001-45	Gear Oil 10/30	Lubricant	Litre	28.89	44.9
PW2001-01	Vios Oil Filter	Filter	box	15.35	28.95

sales_table

```
-- Creating Sales Table
CREATE TABLE sales_table (
  slsID int NOT NULL,
  custID int DEFAULT NULL,
  slsType varchar(45) DEFAULT NULL,
  slsOrdDate date DEFAULT NULL,
  slsShipDate date DEFAULT NULL,
  PRIMARY KEY (slsID),
  KEY custID_idx (custID),
  CONSTRAINT custID FOREIGN KEY (custID) REFERENCES customer_table (custID));

  -- insertion to sales table
  insert into sales_table (slsID, custID, slsType, slsOrdDate, slsShipDate)
  values (21001,10002,'Credit','12/8/2016','8/16/2016'),
         (21002,10001,'Credit','13/08/2016', '8/22/2016'),
         (31001,10003,'Cash','16/08/2016','8/23/2016');
```

Field	Type	Null	Key	Default	Extra
slsID	int	NO	PRI	NULL	
custID	int	YES	MUL	NULL	
slsType	varchar(45)	YES		NULL	
slsOrdDate	date	YES		NULL	
slsShipDate	date	YES		NULL	

slsID	custID	slsType	slsOrdDate	slsShipDate
21001	10002	Credit	12/08/2016	16/08/2016
21002	10001	Credit	13/08/2016	22/08/2016
31001	10003	Cash	16/08/2016	23/08/2016
NULL	NULL	NULL	NULL	NULL

Activity 3

- Using SQL DDL command, create all tables based on the ERD shown in Figure 1. You SQL script should include;
    1. Entity integrity constraint that act as a primary key for table.
    2. Apply domain constraint for service's table to validate service type is either *"Normal", "Major"* or *"Warranty"*.
    3. Apply referential integrity constraints for table that have a relationships based on specific attribute.
    4. Ensure the following field is mandatory;
        - vecOwner
        - tchNo
        - srvNo
        - partNo

Solution

All tables have a primary key and foreign key to apply entity integrity constraint and referential in referential integrity constraints with no null values constraints

Creating the database and use it.

```
CREATE DATABASE ABC_Vehicle_Services;

USE ABC_Vehicle_Services;
```

Creating required tables

```
-- CREATING srvCustomer table ---------------
CREATE TABLE srvCustomer(
custCode VARCHAR(15) NOT NULL,
custName VARCHAR(60) NOT NULL ,
custICNo VARCHAR(15) NOT NULL,
custType VARCHAR(15) NOT NULL,
custAddr VARCHAR(60) DEFAULT NULL,
custTown VARCHAR(30) DEFAULT NULL,
custPostCode INT(5) DEFAULT NULL,
custRegister DATE DEFAULT NULL,
PRIMARY KEY (custCode));
```

Step 4 vecOwner field is mandatory

```
-- creating srvVehicle table
CREATE TABLE srvVehicle (
vecNo VARCHAR(10) NOT NULL,
vecOwner VARCHAR(15) NOT NULL,
vecChassisNo VARCHAR(30) DEFAULT NULL,
vecModel VARCHAR(60) NOT NULL,
vehManufactured VARCHAR(60) DEFAULT NULL,
vehYearProduced INT(4) DEFAULT NULL,
vehLastOdometer BIGINT(20) DEFAULT NULL,
PRIMARY KEY (vecNo),
FOREIGN KEY (vecOwner) REFERENCES srvcustomer(custCode));
```

Step 4 partNo field is mandatory

```
-- creating  srvParts table-----------------------------------------

CREATE TABLE srvParts(
partNo VARCHAR(45) NOT NULL,
partDescription VARCHAR (60) NOT NULL,
partCategory VARCHAR(45) DEFAULT NULL,
partUnit VARCHAR (35) DEFAULT NULL,
partCostPrice DECIMAL(9,2) DEFAULT NULL,
partSalesPrice DECIMAL(9,2) DEFAULT NULL,
partQtyOnHand INT (11) DEFAULT NULL,
PRIMARY KEY (partNo));
```

Step 4 tchNo field is mandatory

```
-- creating srvTechnician table
CREATE TABLE srvTechnician(
tchNo INT(12) NOT NULL,
tchName VARCHAR (60) NOT NULL,
tchHireDate DATE DEFAULT NULL,
PRIMARY KEY (tchNo));
```

For step 2 Apply domain constraint for service's table in srvType (*"Normal", "Major"* or *"Warranty"*)

Step 4 srvNo field is mandatory

```
-- creating srvService table
CREATE TABLE srvService (
srvNo INT(12) NOT NULL,
custCode VARCHAR(15) NOT NULL,
vecNo VARCHAR (10) NOT NULL,
srvType VARCHAR (25)  CHECK (srvType IN ('Normal','Major','Warranty')),
srvDate DATE DEFAULT NULL,
tchNo INT (5) NOT NULL,
PRIMARY KEY(srvNo),
FOREIGN KEY (custCode) REFERENCES srvCustomer (custCode),
FOREIGN KEY (vecNo) REFERENCES srvVehicle(vecNo),
FOREIGN KEY (tchNo) REFERENCES srvTechnician(tchNo));
```

```sql
-- creating srvServiceDetails table
CREATE TABLE srvServiceDetails(
srvNo INT (12) NOT NULL,
partNo VARCHAR(45) NOT NULL,
orderQty INT(11) NOT NULL,
slsPrice DECIMAL (9,2) DEFAULT NULL,
FOREIGN KEY (srvNo) REFERENCES srvService(srvNo),
FOREIGN KEY (partNo) REFERENCES srvParts(partNo));
```

Lets try to insert values srvService table

```sql
INSERT INTO srvservice (srvNo, custCode, vecNo, srvType, tchNo) VALUES ('1', '2', '3', 'Warranty', '4');
```

We got this error

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`abc_vehicle_services`.`srvservice`, CONSTRAINT `srvservice_ibfk_1` FOREIGN KEY (`custCode`) REFERENCES `srvcustomer` (`custCode`))

This is because the primary and foreign key constraints, let's try to fill the required tables first

```sql
103 •  INSERT INTO srvservice (srvNo, custCode, vecNo, srvType, tchNo) VALUES ('1', '1', '1', 'Testing', 1);
104
```

Output

Action Output

#	Time	Action	Message
126	15:39:36	INSERT INTO srvcustomer (custCode, custName, custICNo, custType) VALUES ('1', 'OMA...	1 row(s) affected
127	15:39:38	INSERT INTO srvvehicle (vecNo, vecOwner, vecModel) VALUES ('1', '1', '2000')	1 row(s) affected
128	15:39:40	INSERT INTO srvtechnician (tchNo, tchName) VALUES (1, 'ali')	1 row(s) affected
129	15:39:49	SELECT * FROM abc_vehicle_services.srvservice LIMIT 0, 1000	0 row(s) returned
130	15:40:54	INSERT INTO srvservice (srvNo, custCode, vecNo, srvType, tchNo) VALUES ('1', '1', '1', 'T...	Error Code: 3819. Check constraint 'srvservice_chk_1' is violated.

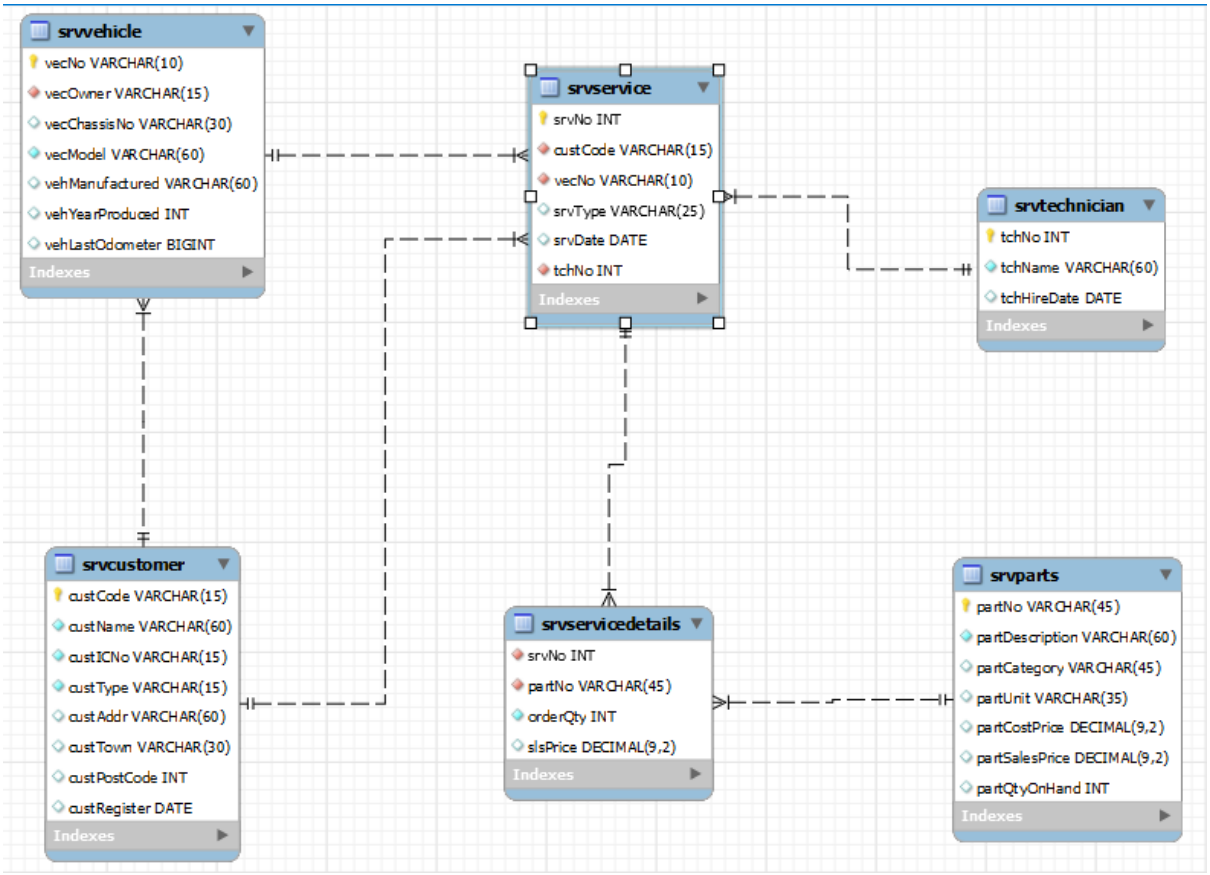
Didn't accept because the srvType must be within (*"Normal", "Major"* or *"Warranty"*)

```sql
103 •  INSERT INTO srvservice (srvNo, custCode, vecNo, srvType, tchNo) VALUES ('1', '1', '1', 'Warranty', 1);
104
```

Output

Action Output

#	Time	Action	Message
126	15:39:36	INSERT INTO srvcustomer (custCode, custName, custICNo, custType) VALUES ('1', 'OMA...	1 row(s) affected
127	15:39:38	INSERT INTO srvvehicle (vecNo, vecOwner, vecModel) VALUES ('1', '1', '2000')	1 row(s) affected
128	15:39:40	INSERT INTO srvtechnician (tchNo, tchName) VALUES (1, 'ali')	1 row(s) affected
129	15:39:49	SELECT * FROM abc_vehicle_services.srvservice LIMIT 0, 1000	0 row(s) returned
130	15:40:54	INSERT INTO srvservice (srvNo, custCode, vecNo, srvType, tchNo) VALUES ('1', '1', '1', 'T...	Error Code: 3819. Check constraint 'srvservice_chk_1' is violated.
131	15:41:57	INSERT INTO srvservice (srvNo, custCode, vecNo, srvType, tchNo) VALUES ('1', '1', '1', 'W...	1 row(s) affected


srvvehicle
vecNo VARCHAR(10)
vecOwner VARCHAR(15)
vecChassisNo VARCHAR(30)
vecModel VARCHAR(60)
vehManufactured VARCHAR(60)
vehYearProduced INT
vehLastOdometer BIGINT
Indexes
srvservice
srvNo INT
custCode VARCHAR(15)
vecNo VARCHAR(10)
srvType VARCHAR(25)
srvDate DATE
tchNo INT
Indexes
srvtechnician
tchNo INT
tchName VARCHAR(60)
tchHireDate DATE
Indexes
srvcustomer
custCode VARCHAR(15)
custName VARCHAR(60)
custICNo VARCHAR(15)
custType VARCHAR(15)
custAddr VARCHAR(60)
custTown VARCHAR(30)
custPostCode INT
custRegister DATE
Indexes
srvservicedetails
srvNo INT
partNo VARCHAR(45)
orderQty INT
slsPrice DECIMAL(9,2)
Indexes
srvparts
partNo VARCHAR(45)
partDescription VARCHAR(60)
partCategory VARCHAR(45)
partUnit VARCHAR(35)
partCostPrice DECIMAL(9,2)
partSalesPrice DECIMAL(9,2)
partQtyOnHand INT
Indexes

Activity 4

Task 1

Solution

This is the structure of srvService table look like before the modifications.

	Field	Type	Null	Key	Default
▶	srvNo	int	NO	PRI	NULL
	custCode	varchar(15)	NO	MUL	NULL
	vecNo	varchar(10)	NO	MUL	NULL
	srvType	varchar(25)	YES		NULL
	srvDate	date	YES		NULL
	tchNo	int	NO	MUL	NULL

After

```
1 •  USE abc_vehicle_services;
2 •  DESC srvService;
3 •  ALTER TABLE srvService MODIFY COLUMN custCode VARCHAR(15) DEFAULT NULL;
4 •  ALTER TABLE srvService MODIFY COLUMN vecNo VARCHAR(10) DEFAULT NULL;
5 •  ALTER TABLE srvService MODIFY COLUMN srvType VARCHAR(25) Not NULL
6    CHECK (srvType IN ('Normal','Major','Warranty'));
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	srvNo	int	NO	PRI	NULL	
	custCode	varchar(15)	YES	MUL	NULL	
	vecNo	varchar(10)	YES	MUL	NULL	
	srvType	varchar(25)	NO		NULL	
	srvDate	date	YES		NULL	
	tchNo	int	NO	MUL	NULL	

Task 2

- Create a *student's* table based on structure in Figure 2. You need to ensure attribute gender will only accept character *'M'* or *'F'*.

  Although I have declared intake to be int and size is 4 but when I show the sturctuer of the table it disappears

```
11      -- Task 2
12      -- creating new database for student table we will call it University
13      CREATE DATABASE University;
14      USE university;
15
16      CREATE TABLE student(
17      StudentID VARCHAR(6) NOT NULL,
18      StuName VARCHAR (50) DEFAULT NULL,
19      Gender CHAR(1) CHECK(Gender IN ('M','F')) DEFAULT NULL,
20      Program VARCHAR (70) DEFAULT NULL,
21      Intake INT(4) DEFAULT NULL,
22      PRIMARY KEY (StudentID));
23
24      DESCRIBE student;
25      DESC student;
```

Field	Type	Null	Key	Default
StudentID	varchar(6)	NO	PRI	NULL
StuName	varchar(50)	YES		NULL
Gender	char(1)	YES		NULL
Program	varchar(70)	YES		NULL
Intake	int	YES		NULL

- Then, insert a sample of records for student information based on the data shown in Table 5.

```
27      ---------------------------------------------
28      -- adding the values from the figure given
29      INSERT INTO student (StudentID, StuName, Gender, Program, Intake)
30      VALUES ('S10023', 'Amran Hamzah', 'M', 'BSc.Comp.Science (SE)', '2014'),
31      ('S20106', 'Goh Seng Huat', 'M', 'BSc.Comp.Science with IM', '2015'),
32      ('S30078', 'Nur Hafizah Rahman', 'F', 'BSc.Comp.Science (Networking)', '2016');
33
```

- Retrieve the records from *student*'s table.

```
--
34 •  SELECT * FROM student;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

StudentID	StuName	Gender	Program	Intake
S10023	Amran Hamzah	M	BSc.Comp.Science (SE)	2014
S20106	Goh Seng Huat	M	BSc.Comp.Science with IM	2015
S30078	Nur Hafizah Rahman	F	BSc.Comp.Science (Networking)	2016
NULL	NULL	NULL	NULL	NULL

- Modify the size of *student name* attribute from *50* to *10* characters. What is the output appear when you execute SQL DDL command for this modification? Explain your answer with justification.

```
182  17:59:44  ALTER TABLE student MODIFY StuName VARCHAR (10) DEFAULT NULL   Error Code: 1265. Data truncated for column 'StuName' at row 1
```

I tried to modify but it didn't work I think because the String length in student name attribute is more than 10, to check if my solution is at least acceptable i tried to modify it to VARCHAR(20) and it works fine.

```
187  18:02:58  ALTER TABLE student MODIFY StuName VARCHAR (20) DEFAULT NULL   0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

Activity 5

1. Create *course* table based on the following attributes;
- ✓ Courseid – characters with size 10 and mandatory.
- ✓ Coursename = characters with size 50.

You should include SQL script and print screen

```
CREATE TABLE course (
CourseID CHARACTER(10) NOT NULL,
CourseName CHARACTER (50) DEFAULT NULL);
```

2. Insert the following records into *course* table.

Course Id	Course Name
CS1001	Programming
CS2001	Advanced Programming
CS3201	Neural Network

You should include SQL script and print screen records already created in the table in your answer.

```
INSERT INTO course (CourseID , CourseName)
VALUES ('CS1001','Programming'),
       ('CS2001','Advanced Programming'),
       ('CS3001','Neural Network');

SELECT * FROM course;
```

	CourseID	CourseName
▶	CS1001	Programming
	CS2001	Advanced Programming
	CS3001	Neural Network

3. Delete all records from *course* table and display the structure of *course* table. You should include SQL script and print screen for displaying the structure of table in your answer.

```
20 •    DELETE FROM course;
```

	CourseID	CourseName

course table structure

	Field	Type	Null	Key	Default
▶	CourseID	char(10)	NO		NULL
	CourseName	char(50)	YES		NULL

4. Remove *course* table from *myPractical* database schema and display the structure of *course* table. What is the output you get when execute SQL command for displaying this structure? You should include SQL script and the output print screen in your answer.

```
DROP TABLE IF EXISTS course;
```

213 21:52:03 DROP TABLE IF EXISTS course 0 row(s) affected

When I tried to execute the query to show the table structure it showed me this error because the table no longer exists.

214 21:52:08 DESC course Error Code: 1146. Table 'university.course' doesn't exist

215 21:53:36 SELECT * FROM course LIMIT 0, 1000 Error Code: 1146. Table 'university.course' doesn't exist

5. What is the different between deleting records and removing the *course* table?

Deleting recodes means that we are just deleting data inside the table, for example when we write 'DELETE FROM course WHERE CourseID = "CS1001" ' we will be deleting one record only, if we want to delete all records in a table we will use the same way in step 3 but we still have the table and we can reinsert values to it, we still have the structure of that particular table. Therefore, when we talk about removing the whole table and whatever it contains and its structure, if we want to reinsert any values we need to recreate the table from the beginning.

Lab exercise

1. System must be able to register a few bank branches across Malaysia. The information need for register branches are branch id, name, city, and manager.

Branches Table:

```
-- BRANCHE TABLE
CREATE TABLE branch (
  brID int NOT NULL,
  brName varchar(50) NOT NULL,
  brCity varchar(50) NOT NULL,
  brManager varchar(255) DEFAULT NULL,
  PRIMARY KEY (brID)
);
```

	Field	Type	Null	Key	Default	Extra
▶	brID	int	NO	PRI	NULL	
	brName	varchar(50)	NO		NULL	
	brCity	varchar(50)	NO		NULL	
	brManager	varchar(255)	YES		NULL	

Inserting values to branch table:

```
INSERT INTO branch (brID, brName, brCity, brManager) VALUES
(1001, 'Terengganu branch', 'Terengganu', 'Ali Omar'),
(1002, 'Kuala Lampur branch', 'Kuala Lampur', 'MD Akash Ali'),
(1003, 'Kelantan branch', 'Kelantan', 'MD Shahdad');
```

	brID	brName	brCity	brManager
▶	1001	Terengganu branch	Terengganu	Ali Omar
	1002	Kuala Lampur branch	Kuala Lampur	MD Akash Ali
	1003	Kelantan branch	Kelantan	MD Shahdad

2. Customer is eligible to open only one bank account. The bank account must consists of account number, ic number, branch id, balance, register date. System must ensure to validate the branch id, ic number is mandatory and account number is a unique id.

Customer Account Table:

```
-- ACCOUNT TABLE
CREATE TABLE cust_account (
  accNo INT NOT NULL,
  custICNo INT NOT NULL ,
  brID int NOT NULL,
  balance DECIMAL(11,2) NOT NULL CHECK(balance > 200),
  registerDate DATE DEFAULT (now()),
  PRIMARY KEY (accNo,custICNo),
  CONSTRAINT fk_brID  FOREIGN KEY (brID) REFERENCES branch (brID),
  CONSTRAINT fk_custICNo FOREIGN KEY (custICNo) REFERENCES customer (custICNo)
);
```

Field	Type	Null	Key	Default	Extra
accNo	int	NO	PRI	NULL	
custICNo	int	NO	PRI	NULL	
brID	int	NO	MUL	NULL	
balance	decimal(11,2)	NO		NULL	
registerDate	date	YES		now()	DEFAULT_GENERATED

3. Customer can register their profile by entering information such as ic number, name, address, town, state, postcode, HP number, email id and status. System will default the customer status as '*New*' when upon completing the registration process.

Customer table:

```
-- CUSTOMERS TABLE
CREATE TABLE customer (
  custICNo int NOT NULL,
  custName varchar(255) NOT NULL,
  custAddr varchar(255) NOT NULL,
  custTown varchar(50) DEFAULT NULL,
  custState varchar(50) DEFAULT NULL,
  custPostCode int DEFAULT NULL,
  hpNo char(10) NOT NULL,
  custEmail varchar(255) NOT NULL,
  custStatus varchar(6) DEFAULT 'New',
  PRIMARY KEY (custICNo),
  CONSTRAINT customer CHECK ((custStatus ='New' OR custStatus = 'Active'))
) ;
```

Field	Type	Null	Key	Default	Extra
custICNo	int	NO	PRI	NULL	
custName	varchar(255)	NO		NULL	
custAddr	varchar(255)	NO		NULL	
custTown	varchar(50)	YES		NULL	
custState	varchar(50)	YES		NULL	
custPostCode	int	YES		NULL	
hpNo	char(10)	NO		NULL	
custEmail	varchar(255)	NO		NULL	
custStatus	varchar(6)	YES		New	

Inserting values to customer Table

```
INSERT INTO customer (custICNo, custName, custAddr, custTown, custState, custPostCode, hpNo, custEmail) VALUES
(12001, 'Omar Alomory', 'Terengganu,UMT B2 201 5 1', 'Kuala Nerus', 'Terengganu', 21300, 0182850579, 'Komar112011@gmail.com'),
(12002, 'Ahmed Mohemmed', 'Jalan Melor, Bandar Bukit Beruntung', 'Selangor', 'Kuala Lampur', 48300, 0164957822, 'Ahmed98@gmail.com'),
(12003, 'Gary Lim', 'Jalan Jelatek, Kementah', 'Kuala Lumpur', 'Kuala Lampur', 54200, 0124876245, 'GLim_ary@gmail.com'),
('12004', 'Hazim Hafizuldin', 'Taman Desa Skudai', 'Johor Bahru', 'Johor', '81300', '0144785685', 'HafizHaz2000@gmail.com');
```

Default values to customer status is '***New***' till he/she register account in bank then it will be '***Active***'

custICNo	custName	custAddr	custTown	custState	custPostCode	hpNo	custEmail	custStatus
12001	Omar Alomory	Terengganu,UMT B2 201 5 1	Kuala Nerus	Terengganu	21300	0182850579	Komar112011@gmail.com	New
12002	Ahmed Mohemmed	Jalan Melor, Bandar Bukit Be...	Selangor	Kuala Lampur	48300	0164957822	Ahmed98@gmail.com	New
12003	Gary Lim	Jalan Jelatek, Kementah	Kuala Lumpur	Kuala Lampur	54200	0124876245	GLim_ary@gmail.com	New
12004	Hazim Hafizuldin	Taman Desa Skudai	Johor Bahru	Johor	81300	0144785685	HafizHaz2000@gmail.com	New
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Customer who has status as *'Active'* is permitted to make apply various loan application; personal loan, housing loan and others. Information need for loan application is loan number, ic number, type of loan, branch id, loan amount and loan status which defaulted to *'Apply'*.

Creating loan Table:

```
-- LOANS TABLE

CREATE TABLE loan (
  loanID int NOT NULL,
  custICNo int NOT NULL,
  loanType varchar(255) DEFAULT NULL,
  brID int NOT NULL,
  loanAmount decimal(9,2) DEFAULT NULL,
  loanStatus varchar(10) DEFAULT Null,
  PRIMARY KEY (loanID),
  CONSTRAINT fk_loan_brID  FOREIGN KEY (brID) REFERENCES branch (brID),
  CONSTRAINT fk_loan_custICNo  FOREIGN KEY (custICNo) REFERENCES customer (custICNo)
);
```

Field	Type	Null	Key	Default	Extra
loanID	int	NO	PRI	NULL	
custICNo	int	NO	MUL	NULL	
loanType	varchar(255)	YES		NULL	
brID	int	NO	MUL	NULL	
loanAmount	decimal(9,2)	YES		NULL	
loanStatus	varchar(10)	YES		NULL	

5. Customers also can withdraw or deposit their money frequently. System must be able to record information such as transaction id, account number, ic number, date, time, transaction type, and amount. The amount of balance with automatically updated upon customer perform transactions either deposit or withdraw the money. The transaction type can be either *'Deposit'* or *'Withdrawal'* only.

Creating customer Transaction Table:

```
-- TRANSACTIONS TABLE

CREATE TABLE cust_transaction (
    tranID INT AUTO_INCREMENT,
    accNo INT NOT NULL,
    custICNo INT  NOT NULL  ,
    tranDate DATE DEFAULT (NOW()),
    tranTime TIME DEFAULT (NOW()),
    tranType VARCHAR(10) NOT NULL,
    tranAmount DECIMAL(11,2) NOT NULL,
    PRIMARY KEY (tranID),
    CONSTRAINT fk_tran_accNo FOREIGN KEY (accNo,custICNo) REFERENCES cust_account (accNo,custICNo),
    CONSTRAINT ck_tran_Type CHECK ((tranType in ('Deposit','Withdrawal')))
);
```

Field	Type	Null	Key	Default	Extra
tranID	int	NO	PRI	NULL	auto_increment
accNo	int	NO	MUL	NULL	
custICNo	int	NO		NULL	
tranDate	date	YES		now()	DEFAULT_GENERATED
tranTime	time	YES		now()	DEFAULT_GENERATED
tranType	varchar(10)	NO		NULL	
tranAmount	decimal(11,2)	NO		NULL	

In a short description of the banking system:

We will fill in some banks information in the branch table:

Inserting values to branch table:

```
INSERT INTO branch (brID, brName, brCity, brManager) VALUES
(1001, 'Terengganu branch', 'Terengganu', 'Ali Omar'),
(1002, 'Kuala Lampur branch', 'Kuala Lampur', 'MD Akash Ali'),
(1003, 'Kelantan branch', 'Kelantan', 'MD Shahdad');
```

brID	brName	brCity	brManager
1001	Terengganu branch	Terengganu	Ali Omar
1002	Kuala Lampur branch	Kuala Lampur	MD Akash Ali
1003	Kelantan branch	Kelantan	MD Shahdad

After that

Customers who wish to register to the bank, they have to fill up their personal information, these values would be saved in customer table.

Inserting values to customer Table

```
INSERT INTO customer (custICNo, custName, custAddr, custTown, custState, custPostCode, hpNo, custEmail) VALUES
(12001, 'Omar Alomory', 'Terengganu,UMT B2 201 5 1', 'Kuala Nerus', 'Terengganu', 21300, 0182850579, 'Komar112011@gmail.com'),
(12002, 'Ahmed Mohemmed', 'Jalan Melor, Bandar Bukit Beruntung', 'Selangor', 'Kuala Lampur', 48300, 0164957822, 'Ahmed98@gmail.com'),
(12003, 'Gary Lim', 'Jalan Jelatek, Kementah', 'Kuala Lumpur', 'Kuala Lampur', 54200, 0124876245, 'GLim_ary@gmail.com'),
('12004', 'Hazim Hafizuldin', 'Taman Desa Skudai', 'Johor Bahru', 'Johor', '81300', '0144785685', 'HafizHaz2000@gmail.com');
```

Default values to customer status is '***New***' till he/she register account in bank then it will be '***Active***'

custICNo	custName	custAddr	custTown	custState	custPostCode	hpNo	custEmail	custStatus
12001	Omar Alomory	Terengganu,UMT B2 201 5 1	Kuala Nerus	Terengganu	21300	0182850579	Komar112011@gmail.com	New
12002	Ahmed Mohemmed	Jalan Melor, Bandar Bukit Be...	Selangor	Kuala Lampur	48300	0164957822	Ahmed98@gmail.com	New
12003	Gary Lim	Jalan Jelatek, Kementah	Kuala Lumpur	Kuala Lampur	54200	0124876245	GLim_ary@gmail.com	New
12004	Hazim Hafizuldin	Taman Desa Skudai	Johor Bahru	Johor	81300	0144785685	HafizHaz2000@gmail.com	New
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Lets register some customers to customers account (cust_account) table and see what will happen

```
INSERT INTO cust_account (accNo, custICNo, brID, balance) VALUES
(123001, 12001, 1001, 15000),
(123002, 12004, 1003, 10000);
```

accNo	custICNo	brID	balance	registerDate
123001	12001	1001	15000.00	2022-11-19
123002	12004	1003	10000.00	2022-11-19
NULL	NULL	NULL	NULL	NULL

After insertion to cust_account table it will modify the values of customer status to Active.

custICNo	custName	custAddr	custTown	custState	custPostCode	hpNo	custEmail	custStatus
12001	Omar Alomory	Terengganu,UMT B2 201 5 1	Kuala Nerus	Terengganu	21300	0182850579	Komar112011@gmail.com	Active
12002	Ahmed Mohemmed	Jalan Melor, Bandar Bukit Be...	Selangor	Kuala Lampur	48300	0164957822	Ahmed98@gmail.com	New
12003	Gary Lim	Jalan Jelatek, Kementah	Kuala Lumpur	Kuala Lampur	54200	0124876245	GLim_ary@gmail.com	New
12004	Hazim Hafizuldin	Taman Desa Skudai	Johor Bahru	Johor	81300	0144785685	HafizHaz2000@gmail.com	Active
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

we put also CONSTRAINT to customers table not accepting any values but Active or New in the custStatus

```
29 •  INSERT INTO customer (custICNo, custName, custAddr, custTown, custState, custPostCode, hpNo, custEmail, custStatus) VALUES
30     (12005, 'ali', 'testing', 'testing ', 'test', 1215, 15454, 'random@eamil.com','test');
```

Output

Action Output

#	Time	Action	Message
87	02:16:30	INSERT INTO `abc_banking`.`customer` (`custICNo`, `custName`, `custAddr`, `custTown`, `c...	Error Code: 3819. Check constraint 'customer' is violated.
88	02:24:52	INSERT INTO customer (custICNo, custName, custAddr, custTown, custState, custPostCod...	Error Code: 3819. Check constraint 'customer' is violated.

And since customer IC No is primary key, customers can register only to one account as it was mentioned in one of the requirements.

```
29 •  INSERT INTO customer (custICNo, custName, custAddr, custTown, custState, custPostCode, hpNo, custEmail, custStatus) VALUES
30     (12001, 'Omar Alomory', 'testing', 'testing ', 'test', 1215, 15454, 'random@eamil.com',DEFAULT);
```

Output

Action Output

#	Time	Action	Message
89	02:25:57	INSERT INTO customer (custICNo, custName, custAddr, custTown, custState, custPostCod...	Error Code: 3819. Check constraint 'customer' is violated.
90	02:26:29	INSERT INTO customer (custICNo, custName, custAddr, custTown, custState, custPostCod...	Error Code: 1062. Duplicate entry '12001' for key 'customer.PRIMARY'

Now let's try to do transactions:

Withdrawal on cust_transaction table

```
INSERT INTO cust_transaction (accNo, custICNo, tranType, tranAmount) VALUES
 (123002, 12004, 'Withdrawal', 2500);
```

tranID	accNo	custICNo	tranDate	tranTime	tranType	tranAmount
1	123002	12004	2022-11...	02:31:21	Withdra...	2500.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Balance would change in cust_account table

accNo	custICNo	brID	balance	registerDate
123001	12001	1001	15000.00	2022-11-19
123002	12004	1003	7500.00	2022-11-19
NULL	NULL	NULL	NULL	NULL

Deposit on cust_transaction table

```
INSERT INTO cust_transaction (accNo, custICNo, tranType, tranAmount) VALUES
(123001, 12001, 'Deposit', 2500);
```

tranID	accNo	custICNo	tranDate	tranTime	tranType	tranAmount
1	123002	12004	2022-11-19	02:31:21	Withdrawal	2500.00
2	123001	12001	2022-11-19	02:39:22	Deposit	2500.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Balance would change in cust_account table too

accNo	custICNo	brID	balance	registerDate
123001	12001	1001	17500.00	2022-11-19
123002	12004	1003	7500.00	2022-11-19
NULL	NULL	NULL	NULL	NULL

Last thing we want to check is the loan table and the condition given in the requirement of this lab exercise too which 'Customer who has status as 'Active' is permitted to make apply various loan application and loanStatus will be the default 'Apply' ' if customer did not finish registration process the default value would be 'Invalid' refers to invalid Apply.

Inserting values to loan table

```
INSERT INTO loan (loanID, custICNo, loanType, brID, loanAmount) VALUES
(12341, 12001, 'Personal loan', 1001, 150000);
INSERT INTO loan (loanID, custICNo, loanType, brID, loanAmount) VALUES
(12342, 12003, 'Housing loan', 1001, 10000);
```

If we checked the customer table to see who did register and who didn't, and which will accept his/her application and who will not.

custICNo	custName	custAddr	custTown	custState	custPostCode	hpNo	custEmail	custStatus
12001	Omar Alomory	Terengganu,UMT B2 201 5 1	Kuala Nerus	Terengganu	21300	0182850579	Komar112011@gmail.com	Active
12002	Ahmed Mohemmed	Jalan Melor, Bandar Bukit Be...	Selangor	Kuala Lampur	48300	0164957822	Ahmed98@gmail.com	New
12003	Gary Lim	Jalan Jelatek, Kementah	Kuala Lumpur	Kuala Lumpur	54200	0124876245	GLim_ary@gmail.com	New
12004	Hazim Hafizuldin	Taman Desa Skudai	Johor Bahru	Johor	81300	0144785685	HafizHaz2000@gmail.com	Active
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Omar Alomory is a customer and his status is Active so his application would be accepted but Gary Lim did not register yet so his application is not accepted.

loanID	custICNo	loanType	brID	loanAmount	loanStatus
12341	12001	Personal l...	1001	150000.00	Apply
12342	12003	Housing l...	1001	10000.00	Invalid

These are the trigger we made to achieve the modifications on the other tables like subtracting or adding to customers account balance, modifying the customers status from New to Active apon finishing the registration processes, and last one is to check if the customer statue is New and intend to apply a loan from the bank the loan status would be invalid, in the other hand is if it Active then the loan status is Apply.

```
-- The amount of balance with automatically updated upon customer perform transactions
-- ( Withdrawal, Deposit )
delimiter $$
CREATE TRIGGER tran_trigger AFTER INSERT
 ON ABC_Banking.cust_transaction
for each row
 BEGIN
    IF (new.tranType = 'Deposit') THEN
        UPDATE cust_Account SET balance = balance + new.tranAmount WHERE ABC_Banking.cust_account.accNo = new.accNo;
    ELSE IF (new.tranType = 'Withdrawal') THEN
        UPDATE cust_Account SET balance = balance - new.tranAmount WHERE abc_banking.cust_account.accNo = new.accNo;
    ELSE
        UPDATE cust_Transaction SET new.tranAmount = 0;
    END IF;
    END IF;
 END $$
 delimiter ;

 -- customer status as 'New' when upon completing the registration process.
 -- this trigger will change it to active
 delimiter $$
CREATE TRIGGER customer_status_trigger AFTER INSERT
 ON ABC_Banking.cust_account
for each row
 BEGIN
    UPDATE  customer set custStatus = 'Active' WHERE ABC_Banking.customer.custICNo = NEW.custICNo;
 END $$
 delimiter ;

-- CHECK CUSTOMER STATUS TO APPLY for A LOAN: if customer already have bank account their loan
-- status will be default apply otherwise default invalid which indicates that they did not open
-- bank account yet
 delimiter $$
CREATE TRIGGER customer_Apply_trigger BEFORE INSERT
 ON ABC_Banking.loan
for each row
 BEGIN
    IF (NEW.custICNo = (SELECT custICNo FROM abc_banking.customer WHERE custStatus = 'Active' LIMIT 1)) THEN
        set NEW.loanStatus = 'Apply' ;
    ELSE
         set NEW.loanStatus = 'Invalid'  ;
    END IF;
 END $$
 delimiter ;
```

That's all from me and thank you.