OMAR ISMAIL ABDJALEEL ALOMORY

Matric No: S63955

Date: 24/11/2022

Lab: MP3

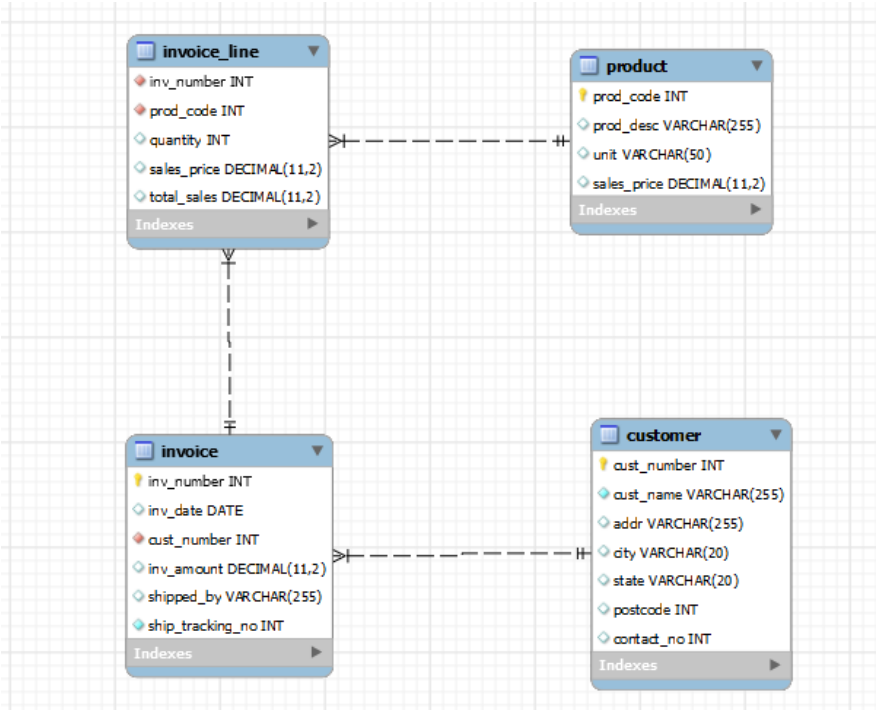
Lecturer: Dr. ROSAIDA ROSLY

MUHAMMAD NASYRUL ADLY BIN MOHD AFANDY

DATABASE (CSF3123) k2

Activity 1

- **Task 1**



- **Task 2**

Creating a database and its tables with define all constraints and referential integrity for these tables.

Customer Table

```
CREATE DATABASE xyz_freight_forwarding;
USE xyz_freight_forwarding;

-- ------------------------------------------------------------------
CREATE TABLE customer (
cust_number CHAR(6) NOT NULL,
cust_name VARCHAR (255) NOT NULL,
addr VARCHAR (255),
city VARCHAR (20),
state VARCHAR (20),
postcode INT(5),
contact_no INT,
CONSTRAINT pk_customer PRIMARY KEY(cust_number)
);
```

Product Table

```
CREATE TABLE product(
prod_code CHAR(7) NOT NULL,
prod_desc VARCHAR (255),
unit VARCHAR(50),
sales_price DECIMAL (11,2) DEFAULT 0.00,
CONSTRAINT pk_product PRIMARY KEY(prod_code)
);
```

Invoice Tabel

```
CREATE TABLE invoice(
inv_number INT NOT NULL,
inv_date DATE DEFAULT (now()),
cust_number CHAR(6) NOT NULL,
inv_amount DECIMAL (11,2) DEFAULT 0.00 ,
shipped_by VARCHAR(255),
ship_tracking_no VARCHAR(50) NOT NULL,
CONSTRAINT pk_invoice PRIMARY KEY(inv_number),
CONSTRAINT fk_invoice FOREIGN KEY(cust_number) REFERENCES customer(cust_number)
);
```

Invoice Line Table

```
CREATE TABLE invoice_line(
inv_number INT NOT NULL,
prod_code CHAR(7) NOT NULL,
quantity INT NOT NULL,
sales_price DECIMAL (11,2) DEFAULT 0.00,
total_sales DECIMAL (11,2) DEFAULT 0.00,
CONSTRAINT pk_invoice_line PRIMARY KEY(inv_number,prod_code),
CONSTRAINT fk_invoice_line_invNumber FOREIGN KEY(inv_number) REFERENCES invoice(inv_number),
CONSTRAINT fk_invoice_line_proCode FOREIGN KEY(prod_code) REFERENCES product(prod_code) ON UPDATE CASCADE
);
```

Another trigger table to calculate the total sales based on quantity and sales_price.

```
-- TRIGGERS TABLES
-- -------------------------------------------------------
DELIMITER $$
CREATE TRIGGER total_price_an_trigger BEFORE INSERT
ON invoice_line
FOR EACH ROW
BEGIN
 SET NEW.total_sales = (NEW.quantity*NEW.sales_price)  ;
END $$
DELIMITER ;
```

- **Task 3**

Create a minimum of **FIVE (5)** numbers of records for each table.

```
-- inserting values to customer table
INSERT INTO customer (cust_number, cust_name, addr, city, state, postcode, contact_no) VALUES
('C10001', 'OMAR', 'UMT', 'Kuala Terengganu', 'Terengganu', 21030, 0182850579),
('C10002', 'AZIM ', 'AH Building, 30st, Johor', 'Johor baru', 'Johor', 32541, 0124587998),
('C10003', 'GARY', 'YET Building, MOH st, ', 'Kuala Terengganu', 'Terengganu', 54245, 0125484787),
('C10004', 'AFIQ', 'UMT', 'Kuala Terengganu', 'Terengganu', 21030, 0177778451),
('C10005', 'HAZIM ', 'UMT', 'Kuala Terengganu', 'Terengganu', 21030, 0178784541);
-- ----------------------------------------------------------------------------------------------------
```

```
SELECT * FROM xyz_freight_forwarding.customer;
```

cust_number	cust_name	addr	city	state	postcode	contact_no
C10001	OMAR	UMT	Kuala Terengganu	Terengganu	21030	182850579
C10002	AZIM	AH Building, 30st, Johor	Johor baru	Johor	32541	124587998
C10003	GARY	YET Building, MOH st,	Kuala Terengganu	Terengganu	54245	125484787
C10004	AFIQ	UMT	Kuala Terengganu	Terengganu	21030	177778451
C10005	HAZIM	UMT	Kuala Terengganu	Terengganu	21030	178784541

```
-- inserting vlaues , product table
INSERT INTO product (prod_code, prod_desc, unit, sales_price) VALUES
 ('HG-5451', 'Bosch 10KG Washing Machine', 'Box', 4000),
 ('SM-5289', 'Samsung Galaxy Note 20 Ultra', 'Box', 3500),
 ('HG-6254', 'Bosch 1.6 gallon Vacuum Bare Tool', 'Box', 1338.56),
 ('BS-7548', 'Beats Studio Buds', 'Box', 830.50) ,
 ('LE-8475', 'LEGO Star Wars', 'Box', 4419.52);
-- ------------------------------------------------------------------------
```

```
SELECT * FROM xyz_freight_forwarding.product;
```

prod_code	prod_desc	unit	sales_price
AW-1001	TV LCD 4 inch	Box	4419.52
BS-7548	Beats Studio Buds	Box	830.50
HG-5451	Bosch 10KG Washing Machine	Box	4000.00
HG-6254	Bosch 1.6 gallon Vacuum Bare Tool	Box	1338.56
SM-5289	Samsung Galaxy Note 20 Ultra	Box	3500.00

```sql
-- inserting values to invoice table
INSERT INTO invoice (inv_number, inv_date, cust_number, inv_amount, shipped_by, ship_tracking_no) VALUES
 (10001, '2016-04-28', 'C10004', 1500, 'JTExpress', 'MY021512155'),
 (10002, '2016-08-23', 'C10003', 2000, 'Pos Laju', 'TU154545484'),
 (10003, '2017-09-15', 'C10002', 5000, 'GDex', 'GD021545445'),
 (10004, '2017-12-20', 'C10005', 1000.25, 'JTExpress', 'MY056984515'),
 (10005, '2018-05-15', 'C10001', 1650.50, 'GDex', 'GD115262655');
-- ----------------------------------------------------------------------------------------------
```

```sql
SELECT * FROM xyz_freight_forwarding.invoice;
```

inv_number	inv_date	cust_number	inv_amount	shipped_by	ship_tracking_no
10001	2016-04-28	C10004	1500.00	JTExpress	MY021512155
10002	2016-08-23	C10003	2000.00	Pos Laju	TU154545484
10003	2017-09-15	C10002	5000.00	GDex	GD021545445
10004	2017-12-20	C10005	1000.25	JTExpress	MY056984515
10005	2018-05-15	C10001	1650.50	GDex	GD115262655

```sql
-- Inserting values to invoice_line table
INSERT INTO invoice_line (inv_number, prod_code, quantity, sales_price) VALUES
(10001,'HG-5451',2,4000),
(10002,'BS-7548',3,830.50),
(10003, 'SM-5289',1,3500),
(10004,'LE-8475',2,4419.52),
(10005,'HG-6254',2,1338.56);
-- ----------------------------------------------------------------------------------------
```

```sql
SELECT * FROM xyz_freight_forwarding.invoice_line;
```

inv_number	prod_code	quantity	sales_price	total_sales
10001	HG-5451	2	4000.00	8000.00
10002	BS-7548	3	830.50	2491.50
10003	SM-5289	1	3500.00	3500.00
10004	AW-1001	2	4419.52	8839.04
10005	HG-6254	2	1338.56	2677.12
NULL	NULL	NULL	NULL	NULL

- **<u>Task 4</u>**

Create new record for customer with customer code as *'C10006'*, name as *'Abdul Rauf Osman'* and state as *'Selangor'*.

```sql
-- TASK 3
INSERT INTO customer (cust_number, cust_name, state) VALUES
('C10006','Abdul Rauf Osman','Selangor');
```

cust_number	cust_name	addr	city	state	postcode	contact_no
C10001	OMAR	UMT	Kuala Terengganu	Terengganu	21030	182850579
C10002	AZIM	AH Building, 30st, Johor	Johor baru	Johor	32541	124587998
C10003	GARY	YET Building, MOH st,	Kuala Terengganu	Terengganu	54245	125484787
C10004	AFIQ	UMT	Kuala Terengganu	Terengganu	21030	177778451
C10005	HAZIM	UMT	Kuala Terengganu	Terengganu	21030	178784541
C10006	Abdul Rauf Osman	NULL	NULL	Selangor	NULL	NULL

Activity 2

1. Retrieve customer code, name, state and contact number for all records.

```sql
SELECT cust_number, cust_name, state,contact_no FROM customer;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

cust_number	cust_name	state	contact_no
C10001	OMAR	Terengganu	182850579
C10002	AZIM	Johor	124587998
C10003	GARY	Terengganu	125484787
C10004	AFIQ	Terengganu	177778451
C10005	HAZIM	Terengganu	178784541
C10006	Abdul Rauf Osman	Selangor	NULL
NULL	NULL	NULL	NULL

2. Retrieve customer information who originated from 'Selangor'.

```sql
SELECT * FROM customer WHERE state = 'Selangor';
```

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

cust_number	cust_name	addr	city	state	postcode	contact_no
C10006	Abdul Rauf Osman	NULL	NULL	Selangor	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Retrieve invoice number, customer code, date of invoice and amount where these invoices shipped using operator known as 'GDex'.

```
SELECT inv_number AS 'Invoice Code'
,cust_number AS 'Customer Code'
,inv_date AS 'Date'
,inv_amount AS Amount FROM invoice WHERE shipped_by = 'GDex';
```

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

Invoice Code	Customer Code	Date	Amount
10003	C10002	2017-09-15	5000.00
10005	C10001	2018-05-15	1650.50

4. Modify the description of the product from 'TV LCD 4 inch' to 'TV LCD Plasma 4 inch' for product code 'AW-1001'.

Befor modification

```
SELECT * FROM product;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

prod_code	prod_desc	unit	sales_price
AW-1001	TV LCD 4 inch	Box	4419.52
BS-7548	Beats Studio Buds	Box	830.50
HG-5451	Bosch 10KG Washing Machine	Box	4000.00
HG-6254	Bosch 1.6 gallon Vacuum Bare Tool	Box	1338.56
SM-5289	Samsung Galaxy Note 20 Ultra	Box	3500.00
NULL	NULL	NULL	NULL

Command

```
UPDATE product SET prod_desc = 'TV LCD 4 Plasma 4 inch' WHERE prod_code = 'AW-1001';
```

Output

```
8 •  SELECT * FROM product;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

prod_code	prod_desc	unit	sales_price
AW-1001	TV LCD 4 Plasma 4 inch	Box	4419.52
BS-7548	Beats Studio Buds	Box	830.50
HG-5451	Bosch 10KG Washing Machine	Box	4000.00
HG-6254	Bosch 1.6 gallon Vacuum Bare Tool	Box	1338.56
SM-5289	Samsung Galaxy Note 20 Ultra	Box	3500.00
NULL	NULL	NULL	NULL

5. Retrieve information that consists of invoice number, date, customer number and amount for invoice date from 22 August 2016 to 25 August 2016.

```
12 •  SELECT inv_number AS 'Invoice Code'
13    ,inv_date AS 'Date'
14    ,cust_number AS 'Customer Code'
15    ,inv_amount AS Amount FROM invoice WHERE inv_date <= '2016-08-25' AND inv_date >='2016-08-22';
```

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

Invoice Code	Date	Customer Code	Amount
10002	2016-08-23	C10003	2000.00

6. Perform insertion to the requirement mentioned in the lab model.

Before inserting the values.

```
SELECT * FROM product;
```

prod_code	prod_desc	unit	sales_price
AW-1001	TV LCD 4 Plasma 4 inch	Box	4419.52
BS-7548	Beats Studio Buds	Box	830.50
HG-5451	Bosch 10KG Washing Machine	Box	4000.00
HG-6254	Bosch 1.6 gallon Vacuum Bare Tool	Box	1338.56
SM-5289	Samsung Galaxy Note 20 Ultra	Box	3500.00

```
SELECT * FROM invoice;
```

inv_number	inv_date	cust_number	inv_amount	shipped_by	ship_tracking_no
10001	2016-04-28	C10004	1500.00	JTExpress	MY021512155
10002	2016-08-23	C10003	2000.00	Pos Laju	TU154545484
10003	2017-09-15	C10002	5000.00	GDex	GD021545445
10004	2017-12-20	C10005	1000.25	JTExpress	MY056984515
10005	2018-05-15	C10001	1650.50	GDex	GD115262655

```
SELECT * FROM invoice_line;
```

inv_number	prod_code	quantity	sales_price	total_sales
10001	HG-5451	2	4000.00	8000.00
10002	BS-7548	3	830.50	2491.50
10003	SM-5289	1	3500.00	3500.00
10004	AW-1001	2	4419.52	8839.04
10005	HG-6254	2	1338.56	2677.12

Command for insertion

```
INSERT INTO product (prod_code, prod_desc, unit, sales_price) VALUES -- A2_T1_6
('AC-5510','Panasonic air-cond 1.5 Hose Power ECO','Box',2300.00);

INSERT INTO invoice (inv_number,inv_date, cust_number,inv_amount,shipped_by, ship_tracking_no) VALUES
(10006,'2016-8-28','C10001',2300.00,'DHL','DHL/2145-004');

INSERT INTO invoice_line (inv_number, prod_code, quantity, sales_price) VALUES
(10006,'AC-5510',1,2300,2300);
```

After insertion

```
SELECT * FROM product;
```

prod_code	prod_desc	unit	sales_price
AC-5510	Panasonic air-cond 1.5 Hose Power E...	Box	2300.00
AW-1001	TV LCD 4 Plasma 4 inch	Box	4419.52
BS-7548	Beats Studio Buds	Box	830.50
HG-5451	Bosch 10KG Washing Machine	Box	4000.00
HG-6254	Bosch 1.6 gallon Vacuum Bare Tool	Box	1338.56
SM-5289	Samsung Galaxy Note 20 Ultra	Box	3500.00

```
SELECT * FROM invoice;
```

inv_number	inv_date	cust_number	inv_amount	shipped_by	ship_tracking_no
10001	2016-04-28	C10004	1500.00	JTExpress	MY021512155
10002	2016-08-23	C10003	2000.00	Pos Laju	TU154545484
10003	2017-09-15	C10002	5000.00	GDex	GD021545445
10004	2017-12-20	C10005	1000.25	JTExpress	MY056984515
10005	2018-05-15	C10001	1650.50	GDex	GD115262655
10006	2016-08-28	C10001	2300.00	DHL	DHL/2145-004

```
SELECT * FROM invoice_line;
```

inv_number	prod_code	quantity	sales_price	total_sales
10001	HG-5451	2	4000.00	8000.00
10002	BS-7548	3	830.50	2491.50
10003	SM-5289	1	3500.00	3500.00
10004	AW-1001	2	4419.52	8839.04
10005	HG-6254	2	1338.56	2677.12
10006	AC-5510	1	2300.00	2300.00

Update product code for product AC-5510 to AC-5587

Before modification

```
SELECT * FROM product;
```

prod_code	prod_desc	unit	sales_price
AC-5510	Panasonic air-cond 1.5 Hose Power E...	Box	2300.00
AW-1001	TV LCD 4 Plasma 4 inch	Box	4419.52
BS-7548	Beats Studio Buds	Box	830.50
HG-5451	Bosch 10KG Washing Machine	Box	4000.00
HG-6254	Bosch 1.6 gallon Vacuum Bare Tool	Box	1338.56
SM-5289	Samsung Galaxy Note 20 Ultra	Box	3500.00

Command

```
27 •  UPDATE product SET prod_code = 'AC-5587' WHERE prod_code = 'AC-5510';
```

After modifications: Display the changes in product code.

```
SELECT * FROM product;
```

prod_code	prod_desc	unit	sales_price
AC-5587	Panasonic air-cond 1.5 Hose Power E...	Box	2300.00
AW-1001	TV LCD 4 Plasma 4 inch	Box	4419.52
BS-7548	Beats Studio Buds	Box	830.50
HG-5451	Bosch 10KG Washing Machine	Box	4000.00
HG-6254	Bosch 1.6 gallon Vacuum Bare Tool	Box	1338.56
SM-5289	Samsung Galaxy Note 20 Ultra	Box	3500.00

Display the invoice number, product code, sales price, total price.

```
SELECT inv_number,prod_code,sales_price,total_sales FROM invoice_line;
```

inv_number	prod_code	sales_price	total_sales
10001	HG-5451	4000.00	8000.00
10002	BS-7548	830.50	2491.50
10003	SM-5289	3500.00	3500.00
10004	AW-1001	4419.52	8839.04
10005	HG-6254	1338.56	2677.12
10006	AC-5587	2300.00	2300.00

<u>Is there any changes occur in the *invoice_line* table? Explain your answer.</u>

Yes, the modification occurred because we declared the product code 'ON UPDATE CASCADE' which means that in any update that happens to the product table it automatically changes in any other tables which have a relation to it. Alternative meaning, update the matching records from the child table automatically when we update the rows in the parent table.

7. Display the description of product and price for product that having sales price is less than RM2000.00.

```
SELECT prod_desc AS 'Product Description'
,sales_price AS 'Sales Price'
 FROM product WHERE sales_price < 2000;
```

Result Grid | Filter Rows: | Export: | Wrap

Product Description	Sales Price
Beats Studio Buds	830.50
Bosch 1.6 gallon Vacuum Bare Tool	1338.56

8. Change the contact number for customer 'C10003' to 016-5585050.
Before modification

```
SELECT * FROM customer;
```

cust_number	cust_name	addr	city	state	postcode	contact_no
C10001	OMAR	UMT	Kuala Terengganu	Terengganu	21030	182850579
C10002	AZIM	AH Building, 30st, Johor	Johor baru	Johor	32541	124587998
C10003	GARY	YET Building, MOH st,	Kuala Terengganu	Terengganu	54245	125484787
C10004	AFIQ	UMT	Kuala Terengganu	Terengganu	21030	177778451
C10005	HAZIM	UMT	Kuala Terengganu	Terengganu	21030	178784541
C10006	Abdul Rauf Osman	NULL	NULL	Selangor	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Command

```
UPDATE customer SET contact_no = 0165585050 WHERE cust_number = 'C10003';
```

After modifications

```
SELECT * FROM customer;
```

cust_number	cust_name	addr	city	state	postcode	contact_no
C10001	OMAR	UMT	Kuala Terengganu	Terengganu	21030	182850579
C10002	AZIM	AH Building, 30st, Johor	Johor baru	Johor	32541	124587998
C10003	GARY	YET Building, MOH st,	Kuala Terengganu	Terengganu	54245	165585050
C10004	AFIQ	UMT	Kuala Terengganu	Terengganu	21030	177778451
C10005	HAZIM	UMT	Kuala Terengganu	Terengganu	21030	178784541
C10006	Abdul Rauf Osman	NULL	NULL	Selangor	NULL	NULL

9. Retrieve customer information and sort the record by state in ascending order.

Before

```
SELECT * FROM customer;
```

cust_number	cust_name	addr	city	state	postcode	contact_no
C10001	OMAR	UMT	Kuala Terengganu	Terengganu	21030	182850579
C10002	AZIM	AH Building, 30st, Johor	Johor baru	Johor	32541	124587998
C10003	GARY	YET Building, MOH st,	Kuala Terengganu	Terengganu	54245	165585050
C10004	AFIQ	UMT	Kuala Terengganu	Terengganu	21030	177778451
C10005	HAZIM	UMT	Kuala Terengganu	Terengganu	21030	178784541
C10006	Abdul Rauf Osman	NULL	NULL	Selangor	NULL	NULL

Command

```
41 •    SELECT * FROM customer ORDER BY state ASC;
```

After

cust_number	cust_name	addr	city	state	postcode	contact_no
C10002	AZIM	AH Building, 30st, Johor	Johor baru	Johor	32541	124587998
C10006	Abdul Rauf Osman	NULL	NULL	Selangor	NULL	NULL
C10001	OMAR	UMT	Kuala Terengganu	Terengganu	21030	182850579
C10003	GARY	YET Building, MOH st,	Kuala Terengganu	Terengganu	54245	165585050
C10004	AFIQ	UMT	Kuala Terengganu	Terengganu	21030	177778451
C10005	HAZIM	UMT	Kuala Terengganu	Terengganu	21030	178784541

10. Retrieve invoice number, customer code, date of invoice, amount and shipped by for invoice date after 22 August 2016 and sort the records by invoice date in descending order.

```
47 •    SELECT inv_number , cust_number, inv_date, inv_amount, shipped_by
48      FROM invoice WHERE inv_date > '2016-8-22' ORDER BY inv_date DESC;
```

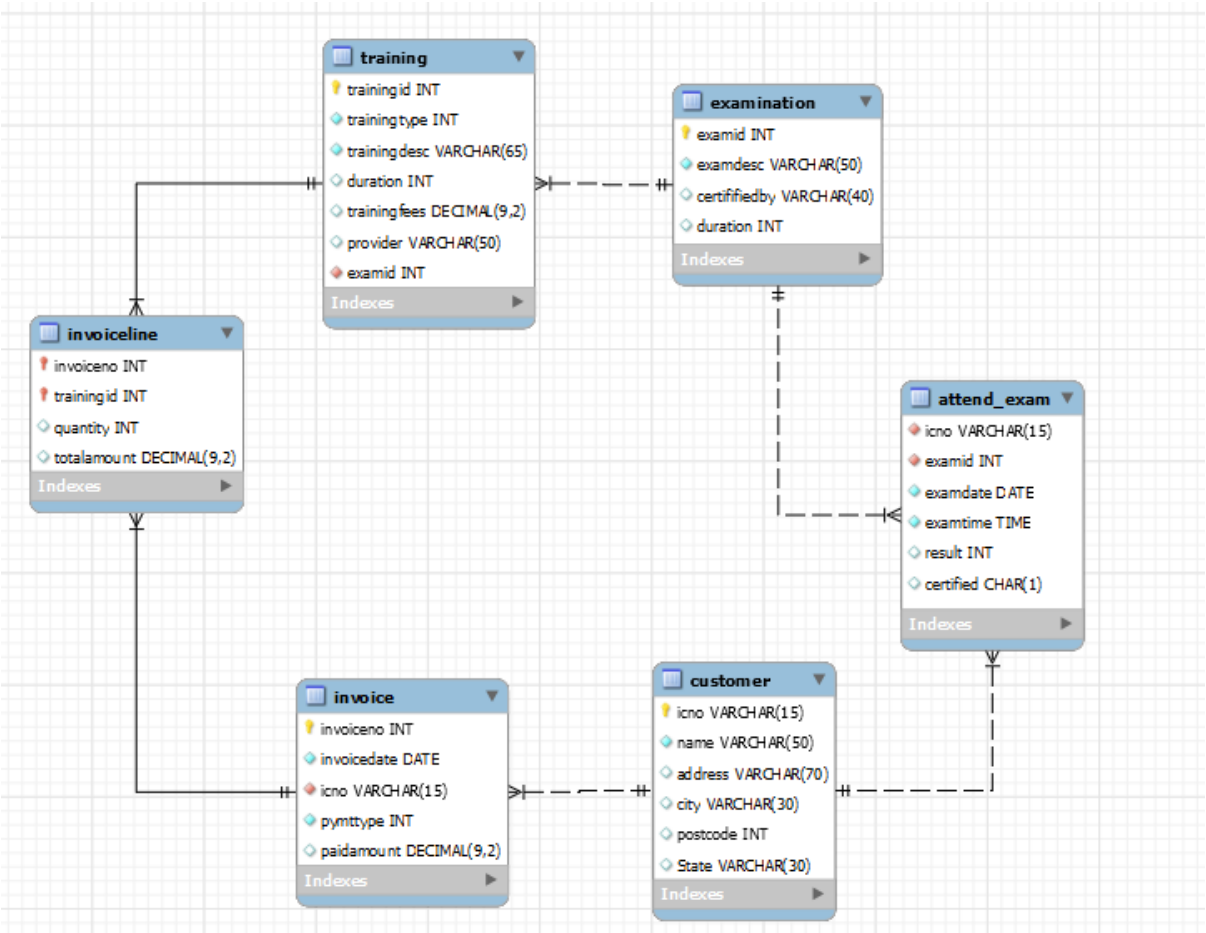
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Con

inv_number	cust_number	inv_date	inv_amount	shipped_by
10005	C10001	2018-05-15	1650.50	GDex
10004	C10005	2017-12-20	1000.25	JTExpress
10003	C10002	2017-09-15	5000.00	GDex
10006	C10001	2016-08-28	2300.00	DHL
10002	C10003	2016-08-23	2000.00	Pos Laju

Lab Exercise:

- **Task 1**

Entity Relationship Diagram



- **Task 2**

Creating the database and tables

Examination table

```sql
CREATE DATABASE XYZ_Professional_Training_Centre;
USE XYZ_Professional_Training_Centre;

CREATE TABLE Examination (
examid INT(5) PRIMARY KEY AUTO_INCREMENT,
examdesc VARCHAR(50) NOT NULL ,
certifiedby VARCHAR (40) DEFAULT NULL,
duration INT DEFAULT NULL
);
ALTER TABLE Examination AUTO_INCREMENT=10000;
```

Output

```sql
SELECT * FROM Examination;
```

examid	examdesc	certifiedby	duration
NULL	NULL	NULL	NULL

81	00:47:08	CREATE TABLE Examination (examid INT(5) PRIMARY KEY AUTO_INCREMENT, examde...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed ...	0.031 sec
82	00:47:11	ALTER TABLE Examination AUTO_INCREMENT=10000	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec

---

Customer table

```sql
CREATE TABLE Customer(
icno VARCHAR (15) PRIMARY KEY ,
name VARCHAR (50) NOT NULL ,
address VARCHAR (70) DEFAULT NULL,
city VARCHAR (30) DEFAULT NULL,
postcode INT(5) DEFAULT NULL,
State VARCHAR (30) DEFAULT NULL
);
```

Output

```sql
SELECT * FROM Customer;
```

icno	name	address	city	postcode	State
NULL	NULL	NULL	NULL	NULL	NULL

84	00:53:47	CREATE TABLE Customer(icno VARCHAR (15) PRIMARY KEY , name VARCHAR (50) NO...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed ...	0.015 sec
85	00:54:10	SELECT * FROM xyz_professional_training_centre.customer LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

---

AttendExam table

```sql
CREATE TABLE AttendExam(
icno VARCHAR (15) NOT NULL,  -- FOREIGN KEY ON UPDATE CASCADE
examid INT(5) NOT NULL , -- FOREIGN KEY ON UPDATE CASCADE
examdate DATE NOT NULL ,
examtime TIME NOT NULL ,
result INT DEFAULT NULL,
certified CHAR(1) , -- CONSTRAINT CHECK ('Y','N')
CONSTRAINT fk_Attand_Exam_icno FOREIGN KEY (icno) REFERENCES Customer (icno) ON UPDATE CASCADE, --
CONSTRAINT fk_Attand_Exam_examid FOREIGN KEY (examid) REFERENCES Examination (examid) ON UPDATE CASCADE,
CONSTRAINT chk_Attand_Exam_certified CHECK(certified IN ('Y','N'))
);
```

Output

```sql
SELECT * FROM AttendExam;
```

icno	examid	examdate	examtime	result	certified

88	01:00:14	CREATE TABLE AttendExam(icno VARCHAR (15) NOT NULL, -- FOREIGN KEY ON UPD...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed ...	0.047 sec
89	01:00:20	SELECT * FROM xyz_professional_training_centre.attendexam LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

---

Invoice table

```sql
CREATE TABLE Invoice (
invoiceno INT (7) PRIMARY KEY AUTO_INCREMENT,
invoicedate DATE NOT NULL,
icno  VARCHAR (15) NOT NULL , -- FOREIGN KEY ON UPDATE CASCADE CUSTOMER TABLE
pymttype INT(1) NOT NULL DEFAULT 1,  -- CHECK IF (1,2,3)  1 = CASH , 2 = CREDIT CARD  3 = MISC DEFULT IS 1
paidamount DECIMAL (9,2),
CONSTRAINT fk_Invoice_icno FOREIGN KEY (icno) REFERENCES Customer (icno) ON UPDATE CASCADE,
CONSTRAINT chk_Invoice_pymttype CHECK (pymttype IN (1,2,3))
);
```

Output

```sql
SELECT * FROM Invoice;
```

invoiceno	invoicedate	icno	pymttype	paidamount
NULL	NULL	NULL	NULL	NULL

90	01:06:15	CREATE TABLE Invoice (invoiceno INT (7) PRIMARY KEY AUTO_INCREMENT, invoiced...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed ...	0.047 sec
91	01:06:55	SELECT * FROM xyz_professional_training_centre.invoice LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

---

Training table

```sql
CREATE TABLE Training(
trainingid INT(5) PRIMARY KEY AUTO_INCREMENT,
trainingtype INT (1) NOT NULL, -- CHECK CONSTRAINT IF IN (1,2,3) 1 = Management  2 = Technical  3 = Support  DEFAULT 1
trainingdesc VARCHAR (65) NOT NULL ,
duration INT (4) DEFAULT NULL,
trainingfees DECIMAL (9,2) DEFAULT NULL ,
provider VARCHAR (50) DEFAULT NULL,
examid INT (5) NOT NULL, -- FOREIGN KEY ON UPDATE CASCADE FROM Examination TABLE
CONSTRAINT chk_Training_trainingtype CHECK (trainingtype IN(1,2,3)),
CONSTRAINT fk_Training_examid FOREIGN KEY(examid) REFERENCES Examination (examid) ON UPDATE CASCADE
);
```

Output

```sql
SELECT * FROM Training;
```

trainingid	trainingtype	trainingdesc	duration	trainingfees	provider	examid
NULL	NULL	NULL	NULL	NULL	NULL	NULL

92	01:08:09	CREATE TABLE Training(trainingid INT(5) PRIMARY KEY AUTO_INCREMENT, trainingtyp...	0 row(s) affected, 4 warning(s): 1681 Integer display width is deprecated and will be removed ...	0.047 sec
93	01:09:59	SELECT * FROM xyz_professional_training_centre.training LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

InvoiceLine table

```sql
CREATE TABLE InvoiceLine(
invoiceno INT(7) NOT NULL UNIQUE , -- FOREIGN KEY ON UPDATE CASCADE FROM INVOICE TABLE  BOTH ARE PRIMARY KEY
trainingid  INT (5) NOT NULL UNIQUE , -- FOREIGN KEY ON UPDATE CASCADE FROM TRAINING TABLE BOTH ARE PRIMARY KEY
quantity INT DEFAULT NULL,
totalamount DECIMAL (9,2) DEFAULT NULL,
CONSTRAINT pk_InvoiceLine_conbination PRIMARY KEY(invoiceno,trainingid) ,
CONSTRAINT fk_InvoiceLine_invoiceno FOREIGN KEY (invoiceno) REFERENCES Invoice (invoiceno) ON UPDATE CASCADE,
CONSTRAINT fk_InvoiceLine_trainingid FOREIGN KEY (trainingid) REFERENCES Training (trainingid) ON UPDATE CASCADE
);
```

Output

```sql
SELECT * FROM Invoiceline;
```

invoiceno	trainingid	quantity	totalamount
NULL	NULL	NULL	NULL

94	01:12:18	CREATE TABLE InvoiceLine(invoiceno INT(7) NOT NULL UNIQUE , -- FOREIGN KEY ON ...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed ...	0.047 sec
95	01:12:24	SELECT * FROM xyz_professional_training_centre.invoiceline LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

---

*This trigger will calculate the total amount in the invoiceLine table the moment of insertion and will automatically set the value of total amount = (trainingFee * quantity)*

```
-- ==========================================================================================
-- This triiger will calculate the total amount in InvoiceLine table the moment of insertion.
DELIMITER $$
CREATE TRIGGER bi_Invoiceline_trigger BEFORE INSERT
ON Invoiceline
FOR EACH ROW
BEGIN
 DECLARE fee_of_training double ;
 SET fee_of_training = (SELECT trainingfees FROM Training WHERE trainingid = NEW.trainingid);
    BEGIN
        SET NEW.totalamount = (NEW.quantity * fee_of_training);
        END ;
END $$
DELIMITER ;
-- ==========================================================================================
```

97 01:27:37 CREATE TRIGGER bi_Invoiceline_trigger BEFORE INSERT ON Invoiceline FOR EACH R... 0 row(s) affected 0.016 sec

- **Task 3**

```
-- inserting data to Examination table
INSERT INTO Examination (examid, examdesc, certififiedby, duration) VALUES
(default,'Data Structure and Algorithm', 'MYSQL Organazations', 3),
(default,'Data Manipulation Language', 'Oracle Corporation', 5),
(default,'Web Based Design', 'W3C Corporation', 2),
(default,'MYSQL Fundamentals', 'MYSQL Organazations', 4),
(default,'Database Management', 'Oracle Corporation', 3);
-- ==========================================================================================
-- inserting data to Customer table
INSERT INTO Customer (icno, name, address, city, postcode, State) VALUE
('890315-10-3278','Ruhaimie Osman','No. 1254 Taman Teluk','Paka',21300,'Terengganu'),
('845152-05-5412','Omar Alomroy','No. 2546 Taman Prajurit','Jeli',52300,'Putrajaya'),
('784958-12-7485','MD Akash Ali','No. 7845 Jalan Sepakat Indah 2 ','Bachok',32400,'Kelantan'),
('548524-13-2254','Shahdad Ahsan','University Malaysia Terengganu','Kuala Terengganu',21030,'Terengganu'),
('778458-02-7291','Moad Mohammed','No. 127 Jalan Kelawai Belle Vue','George Town',10350,'Pulau Pinang');
-- ==========================================================================================

-- inserting data to AttendExam table
INSERT INTO AttendExam (icno, examid, examdate, examtime, result, certified) VALUES
('778458-02-7291',10001,'2022-10-20','09:30',85,'Y'),
('845152-05-5412',10004,'2022-09-15','11:00',55,'N'),
('890315-10-3278',10002,'2022-08-31','08:30',90,'Y'),
('784958-12-7485',10000,'2022-10-13','10:00',89,'Y'),
('548524-13-2254',10003,'2022-10-16','09:30',52,'N');
-- ==========================================================================================
-- inserting data to Invoice table
INSERT INTO Invoice (invoiceno, invoicedate, icno, pymttype, paidamount) VALUES
(default,'2022-10-10' ,'778458-02-7291',2,3600),
(default,'2022-08-21' ,'890315-10-3278',1,3000),
(default,'2022-10-06' ,'548524-13-2254',3,5000),
(default,'2022-10-03' ,'784958-12-7485',1,4050),
(default,'2022-09-05' ,'845152-05-5412',2,4100);
-- ==========================================================================================
```

```sql
INSERT INTO Training (trainingid, trainingtype, trainingdesc, duration, trainingfees, provider,examid) VALUES
(default, 1, 'MySQL 8.0 for Database Administrators',3,2400, 'MYSQL',10000),
(default, 3, 'Data Manipulation, Organizing, Analyzing and Designing',7,3600, 'MYSQL',10003),
(default, 2, 'Creating Web Services Using Java(TM) Technology',3,3500, 'Oracle',10001),
(default, 3, 'Programming in HTML5 with JavaScript and CSS3',3,3800, 'HTML',10002),
(default, 2, 'Oracle Database : Performance Tuning',3,4500, 'Oracle',10004);
-- ==========================================================================================
 -- inserting data to Invoiceline table
 INSERT INTO Invoiceline (invoiceno, trainingid, quantity) VALUES
 (1,1,3),
 (2,3,2),
 (3,5,5),
 (4,2,3),
 (5,4,3);
 -- ==========================================================================================
```

Output for insertion

115	01:32:31	CREATE TRIGGER bi_Invoiceline_trigger BEFORE INSERT ON Invoiceline FOR EACH R...	0 row(s) affected		0.000 sec
116	01:32:42	INSERT INTO Examination (examid, examdesc, certifiedby, duration) VALUES (default,'Dat...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0		0.015 sec
117	01:32:46	INSERT INTO Customer (icno, name, address, city, postcode, State) VALUE ('890315-10-32...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0		0.016 sec
118	01:32:49	INSERT INTO AttendExam (icno, examid, examdate, examtime, result, certified) VALUES ('7...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0		0.000 sec
119	01:32:52	INSERT INTO Invoice (invoiceno, invoicedate, icno, pymttype, paidamount) VALUES (defau...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0		0.000 sec
120	01:32:56	INSERT INTO Training (trainingid, trainingtype, trainingdesc, duration, trainingfees, provider,...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0		0.000 sec
121	01:33:06	INSERT INTO Invoiceline (invoiceno, trainingid, quantity) VALUES (1,1,3), (2,3,2), (3,5,5), (4,...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0		0.000 sec

```sql
SELECT * FROM Examination;
```

examid	examdesc	certifiedby	duration
10000	Data Structure and Algorithm	MYSQL Organazations	3
10001	Data Manipulation Language	Oracle Corporation	5
10002	Web Based Design	W3C Corporation	2
10003	MYSQL Fundamentals	MYSQL Organazations	4
10004	Database Management	Oracle Corporation	3
NULL	NULL	NULL	NULL

```sql
SELECT * FROM Customer;
```

icno	name	address	city	postcode	State
548524-13-2254	Shahdad Ahsan	University Malaysia Terengganu	Kuala Terengganu	21030	Terengganu
778458-02-7291	Moad Mohammed	No. 127 Jalan Kelawai Belle Vue	George Town	10350	Pulau Pinang
784958-12-7485	MD Akash Ali	No. 7845 Jalan Sepakat Indah 2	Bachok	32400	Kelantan
845152-05-5412	Omar Alomroy	No. 2546 Taman Prajurit	Jeli	52300	Putrajaya
890315-10-3278	Ruhaimie Osman	No. 1254 Taman Teluk	Paka	21300	Terengganu
NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT * FROM AttendExam;
```

icno	examid	examdate	examtime	result	certified
778458-02-7291	10001	2022-10-20	09:30:00	85	Y
845152-05-5412	10004	2022-09-15	11:00:00	55	N
890315-10-3278	10002	2022-08-31	08:30:00	90	Y
784958-12-7485	10000	2022-10-13	10:00:00	89	Y
548524-13-2254	10003	2022-10-16	09:30:00	52	N

```
SELECT * FROM Invoice;
```

invoiceno	invoicedate	icno	pymttype	paidamount
1	2022-10-10	778458-02-7291	2	3600.00
2	2022-08-21	890315-10-3278	1	3000.00
3	2022-10-06	548524-13-2254	3	5000.00
4	2022-10-03	784958-12-7485	1	4050.00
5	2022-09-05	845152-05-5412	2	4100.00
NULL	NULL	NULL	NULL	NULL

```
SELECT * FROM Training;
```

trainingid	trainingtype	trainingdesc	duration	trainingfees	provider	examid
1	1	MySQL 8.0 for Database Administrators	3	2400.00	MYSQL	10000
2	3	Data Manipulation, Organizing, Analyzing and D...	7	3600.00	MYSQL	10003
3	2	Creating Web Services Using Java(TM) Technol...	3	3500.00	Oracle	10001
4	3	Programming in HTML5 with JavaScript and CSS3	3	3800.00	HTML	10002
5	2	Oracle Database : Performance Tuning	3	4500.00	Oracle	10004
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT * FROM Invoiceline;
```

invoiceno	trainingid	quantity	totalamount
1	1	3	7200.00
2	3	2	7000.00
3	5	5	22500.00
4	2	3	10800.00
5	4	3	11400.00
NULL	NULL	NULL	NULL

- **Task 4**

```
-- 1. Retrieve a list of training code, training description, training duration and fees.
SELECT trainingid, trainingdesc, duration, trainingfees FROM Training;
```

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

trainingid	trainingdesc	duration	trainingfees
1	MySQL 8.0 for Database Administrators	3	2400.00
2	Data Manipulation, Organizing, Analyzing and D...	7	3600.00
3	Creating Web Services Using Java(TM) Technol...	3	3500.00
4	Programming in HTML5 with JavaScript and CSS3	3	3800.00
5	Oracle Database : Performance Tuning	3	4500.00
NULL	NULL	NULL	NULL

```
-- 2. Retrieve customer ic number, name and city they live for customer from 'Terengganu' or 'Kelantan'.
SELECT icno, name, city FROM Customer WHERE State IN ('Terengganu','Kelantan');
```

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

icno	name	city
548524-13-2254	Shahdad Ahsan	Kuala Terengganu
784958-12-7485	MD Akash Ali	Bachok
890315-10-3278	Ruhaimie Osman	Paka

Is this question is asking for this

```
-- 3. Display the candidate ic number and the certification status who taking the examination.
SELECT icno, certified FROM AttendExam;
```

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

icno	certified
778458-02-7291	Y
845152-05-5412	N
890315-10-3278	Y
784958-12-7485	Y
548524-13-2254	N

Or asking who only certified and his status is 'Y'

```
SELECT icno, certified, examdesc FROM AttendExam INNER JOIN Examination ON Examination.examid = AttendExam.examid
WHERE certified = 'Y';
```

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

icno	certified	examdesc
778458-02-7291	Y	Data Manipulation Language
890315-10-3278	Y	Web Based Design
784958-12-7485	Y	Data Structure and Algorithm

```
-- ==========================================================================================
-- 4. Retrieve examination code, description and duration for examination which the certification is comes from 'Oracle Corporation'.
SELECT examid, examdesc, duration FROM Examination WHERE certifiedby = 'Oracle Corporation';
```

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

examid	examdesc	duration
10001	Data Manipulation Language	5
10004	Database Management	3
NULL	NULL	NULL

```
-- 5. Display invoice information where the customer pay the invoice amount greater than RM4000.00.
SELECT
invoiceno AS 'Invoice No',
invoicedate AS 'Date',
icno AS 'IC No',
CASE pymttype WHEN 1 THEN 'Cash' WHEN 2 THEN 'Credit Card' ELSE 'Misc' END AS 'Payment Type',
paidamount AS 'Paid Amount'
FROM Invoice WHERE paidamount > 4000;
```

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

Invoice No	Date	IC No	Payment Type	Paid Amount
3	2022-10-06	548524-13-2254	Misc	5000.00
4	2022-10-03	784958-12-7485	Cash	4050.00
5	2022-09-05	845152-05-5412	Credit Card	4100.00

Before modification

```
-- 6. Modify the duration for examination code 10004 from 3 hours to 4 hours.
SELECT * FROM Examination;
```

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

examid	examdesc	certifiedby	duration
10000	Data Structure and Algorithm	MYSQL Organazations	3
10001	Data Manipulation Language	Oracle Corporation	5
10002	Web Based Design	W3C Corporation	2
10003	MYSQL Fundamentals	MYSQL Organazations	4
10004	Database Management	Oracle Corporation	3
NULL	NULL	NULL	NULL

After modifications

```
-- 6. Modify the duration for examination code 10004 from 3 hours to 4 hours.
SELECT * FROM Examination;
UPDATE Examination SET  duration = 4 WHERE examid = 10004;
```

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

examid	examdesc	certifiedby	duration
10000	Data Structure and Algorithm	MYSQL Organazations	3
10001	Data Manipulation Language	Oracle Corporation	5
10002	Web Based Design	W3C Corporation	2
10003	MYSQL Fundamentals	MYSQL Organazations	4
10004	Database Management	Oracle Corporation	4
NULL	NULL	NULL	NULL

```
-- 7. Display a record for ic number, examination code, date of examination and time of examination.
--    Your output must be sort out based on the latest date of examination.
    SELECT icno,examid, examdate , examtime FROM attendExam ORDER BY examdate DESC;
-- ==============================================================
```

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

icno	examid	examdate	examtime
778458-02-7291	10001	2022-10-20	09:30:00
548524-13-2254	10003	2022-10-16	09:30:00
784958-12-7485	10000	2022-10-13	10:00:00
845152-05-5412	10004	2022-09-15	11:00:00
890315-10-3278	10002	2022-08-31	08:30:00

```
-- 8. The ic number for Ruhaimie Osman is wrongly key-in as 890315-10-3278. Perform the query to retreive
--    ic number and name from customer. Subsequently, and retrieve ic number from invoice table.
--    Verify the ic number for both tables. Write a SQL statement to change the ic number to 890315-10-3288.
--    Verify whether change occurs in invoice table. Justify your answer.
   SELECT icno, name FROM customer;
```

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

icno	name
548524-13-2254	Shahdad Ahsan
778458-02-7291	Moad Mohammed
784958-12-7485	MD Akash Ali
845152-05-5412	Omar Alomroy
890315-10-3278	Ruhaimie Osman
NULL	NULL

This is the two tables (customer, invoice) joined together to show if we change the icon in the customer table will it affect the icon in the invoice table

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

name	Customer ICNo	Invoice ICNo
Shahdad Ahsan	548524-13-2254	548524-13-2254
Moad Mohammed	778458-02-7291	778458-02-7291
MD Akash Ali	784958-12-7485	784958-12-7485
Omar Alomroy	845152-05-5412	845152-05-5412
Ruhaimie Osman	890315-10-3278	890315-10-3278

After Update

```
173 •    UPDATE customer  SET customer.icno = '890315-10-3288' WHERE name = 'Ruhaimie Osman';
```

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

name	Customer ICNo	Invoice ICNo
Shahdad Ahsan	548524-13-2254	548524-13-2254
Moad Mohammed	778458-02-7291	778458-02-7291
MD Akash Ali	784958-12-7485	784958-12-7485
Omar Alomroy	845152-05-5412	845152-05-5412
Ruhaimie Osman	890315-10-3288	890315-10-3288

As we can see the ICNo have changed in both tables since we already initialise it to whenever there is an update in the primary key do the update too on the other related tables to it.

Verifying before modification

```
176    -- 9. Perform the query to verify the payment type for inoive 4 is 1 (Cash). Then, make a changes for type of payment for
177    --    this invoice from Cash to Misc. Your answer should include original record, SQL statement to modify payment type and,
178    --    finally display invoice information in order to ascertain that the modification has been done accordingly.
179
180 •    SELECT * FROM invoice WHERE invoiceno = 4;
```

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

invoiceno	invoicedate	icno	pymttype	paidamount
4	2022-10-03	784958-12-7485	1	4050.00
NULL	NULL	NULL	NULL	NULL

```
180 •    SELECT * FROM invoice ;
```

Result Grid | Filter Rows: | Edit:

invoiceno	invoicedate	icno	pymttype	paidamount
1	2022-10-10	778458-02-7291	2	3600.00
2	2022-08-21	890315-10-3288	1	3000.00
3	2022-10-06	548524-13-2254	3	5000.00
4	2022-10-03	784958-12-7485	1	4050.00
5	2022-09-05	845152-05-5412	2	4100.00
NULL	NULL	NULL	NULL	NULL

After modifications

```
183 •    SELECT * FROM invoice WHERE invoiceno = 4; -- after modification
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wra

invoiceno	invoicedate	icno	pymttype	paidamount
4	2022-10-03	784958-12-7485	3	4050.00
NULL	NULL	NULL	NULL	NULL

```
184 •    SELECT * FROM invoice;  --  records after modification
```

Result Grid | Filter Rows: | Edit: | Export/Import:

invoiceno	invoicedate	icno	pymttype	paidamount
1	2022-10-10	778458-02-7291	2	3600.00
2	2022-08-21	890315-10-3288	1	3000.00
3	2022-10-06	548524-13-2254	3	5000.00
4	2022-10-03	784958-12-7485	3	4050.00
5	2022-09-05	845152-05-5412	2	4100.00
NULL	NULL	NULL	NULL	NULL

```
-- 10. Retrieve details about the examination information where the records sorted by certification
--     provider as an ascending order, followed by duration as descending order.
        SELECT * FROM examination ORDER BY certifiedby ASC  , duration DESC;
```

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

examid	examdesc	certifiedby	duration
10003	MYSQL Fundamentals	MYSQL Organazations	4
10000	Data Structure and Algorithm	MYSQL Organazations	3
10001	Data Manipulation Language	Oracle Corporation	5
10004	Database Management	Oracle Corporation	4
10002	Web Based Design	W3C Corporation	2
NULL	NULL	NULL	NULL

```
SELECT
Customer.name,
Invoice.paidamount,
InvoiceLine.quantity,
InvoiceLine.totalamount ,
(totalamount - paidamount ) as remaining
FROM Invoice
INNER JOIN Customer ON Customer.icno = Invoice.icno
INNER JOIN InvoiceLine ON InvoiceLine.invoiceno = Invoice.invoiceno;
```

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

name	paidamount	quantity	totalamount	remaining
Moad Mohammed	3600.00	3	7200.00	3600.00
Ruhaimie Osman	3000.00	2	7000.00	4000.00
Shahdad Ahsan	5000.00	5	22500.00	17500.00
MD Akash Ali	4050.00	3	10800.00	6750.00
Omar Alomroy	4100.00	3	11400.00	7300.00