**RMIT International University**

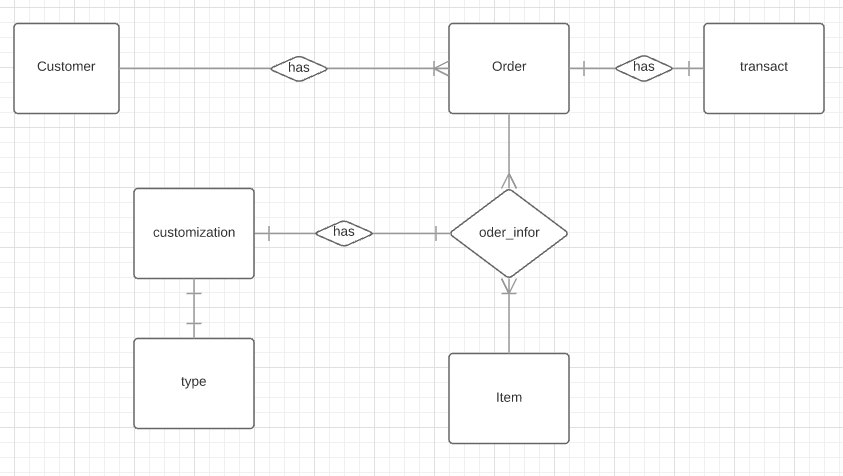
**Bachelor of Business Program**

### **ASSIGNMENT COVER PAGE**

**Your assessment will not be accepted unless all fields below are completed**

|  |  |
| --- | --- |
| **Subject Code:** | **ISYS2421** |
| **Subject Name:** | **Business Database Management And Analytics** |
| **Location where you study:** | **RMIT University** |
| **Title of Assignment:** | **Assignment 2** |
| **Student name:** | **Ly Pham Duy** |
| **Student Number:** | **s3594800** |
| **Student Email Address:** | **s3594800@rmit.edu.vn** |
| **Assignment due date:** | **22 September 2017** |
| **Date of Submission:** | **22 September 2017** |
| **Late Submission Approval** | **NA** |

**Diagram**



**Table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table Customer** | **Order** | **Order\_infor** | **Item** | **customization** | **Type** | **transact** |
| Customer\_id | Order\_id | info\_id | Item\_id | Customization\_id | Type\_info | Transact\_id |
| Surname | Date\_oder | Order\_id | Name | Data | size | Date\_purchase |
| Firstname | quantity | Items\_id | Style |  | price | purchaseType |
| DOB | Total\_price | Customization\_id | Type |  |  | Received |
| Address | Customer\_id | Type\_info | Sleeves\_type |  |  | Cc\_num |
| surburb |  |  | Size |  |  | Cc\_exdate |
| state |  |  | Color |  |  | Paypal\_id |
| postcode |  |  | Price |  |  | Order\_id |
| Email |  |  | Stocked\_quantity |  |  |  |
| Phone |  |  |  |  |  |  |

**Business Rule/Assumption**

* For **`item`** table, the store going to increase the type of their product and will not only selling shirt anymore. So I made a column name so that can add items such as: pants, t-shirt, polo.
* For `**type`** table, I assumed that the the store will have only 3 fixed size for image, logo and text so that it will appear the best on the shirt and also the customer will be easy to choose how large or small the customization (image,logo, text) can be.
  + I created Small, Medium, Large logo, image, text
* For size in `**type`** table, the value is understand as cm2
* However, for `**type`** table, the store can also add more type of customizationt that they can provide for the customer. The store can still scale up and there is no limitation.
* For `**transact`** table. Based on the requirement, I assumed that all the transaction information wil be record into one table because there only a few information needed to be stored as well as all of them need a received tick.
* I assumed that there will be a total price for the `**order`** table.
* For `**customization`** table, there will only a column data to store anything that the customer submit to print on their shirt.
* In `**transact`,** information such as purchase date, type of payment and payment type information will be stored.
* I could have put the id for item table in VARCHAR to better specified the product. However, I think it is better to do it as auto increment int because it is better for business scale up when they will input so many more products.

**Business rule**

* For null and not null, field that have to be inputed (not null) are:
  + Primary key
  + Moreover, only the talbe `**type`**  is all not null because those information is needed to be clear for the customer.
  + Customer also have to input the data in the `**customization`**table.
* Yes and No, as well as (money receive) tick fields in the database are handled with VARCHAR(1) for Y(yes) and N(no) **or** R(received) or N(not)
* For the scenario when customer choose the buying shirt. They will choose, the shirt which fit requirement such as: type, sleeves, style, size, color, and price. When they choose customization, they will submit what they want to print on the shirt **AND** specific type of customization that the shop offer. (type information located in **type** talbe)
* Quantity can only be specified in `**order`** talbe
  + So, if they want to order many different shirt/, different customization. They will need to make many order to fit with their needs.
    - To be more clear, one order can only consist one type of shirt with one customization and the amount of it.
* If the customer cancel order, based on the customer id, they shop can delete all the information, ‘Y’ can be added and the meaning will be order cancelled in the database.
  + These information can be used to calculate the order that was cancelled ( to track scam, bad intention people or just known about the number of total order made and percentage of not canceled and cancelled order)

**SQL commands**

CREATE TABLE IF NOT EXISTS `s3594800`.`customer1` (

`customer\_id` INT(11) NOT NULL AUTO\_INCREMENT,

`surname` VARCHAR(45) NULL DEFAULT NULL,

`lastname` VARCHAR(45) NULL DEFAULT NULL,

`DOB` DATE NULL DEFAULT NULL,

`address` VARCHAR(45) NULL DEFAULT NULL,

`surburb` VARCHAR(45) NULL DEFAULT NULL,

`state` VARCHAR(45) NULL DEFAULT NULL,

`postcode` INT(4) NULL DEFAULT NULL,

PRIMARY KEY (`customer\_id`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `s3594800`.`customization`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `s3594800`.`customization` (

`customization\_id` INT(11) NOT NULL AUTO\_INCREMENT,

`data` VARCHAR(45) NOT NULL,

PRIMARY KEY (`customization\_id`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `s3594800`.`item`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `s3594800`.`item` (

`item\_id` INT(11) NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL,

`style` VARCHAR(255) NULL DEFAULT NULL,

`type` VARCHAR(45) NULL DEFAULT NULL,

`sleeves\_type` VARCHAR(45) NULL DEFAULT NULL,

`size` VARCHAR(3) NULL DEFAULT NULL,

`color` VARCHAR(45) NULL DEFAULT NULL,

`price` INT(4) NULL DEFAULT NULL,

PRIMARY KEY (`item\_id`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `s3594800`.`order`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `s3594800`.`order` (

`order\_id` INT(11) NOT NULL AUTO\_INCREMENT,

`date\_order` DATE NULL DEFAULT NULL,

`quantity` INT(11) NULL DEFAULT NULL,

`total\_price` INT(11) NULL DEFAULT NULL,

`customer\_id` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`order\_id`),

INDEX `fk\_order\_customer11\_idx` (`customer\_id` ASC),

CONSTRAINT `fk\_order\_customer11`

FOREIGN KEY (`customer\_id`)

REFERENCES `s3594800`.`customer1` (`customer\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `s3594800`.`type`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `s3594800`.`type` (

`type\_info` VARCHAR(10) NOT NULL,

`size` VARCHAR(3) NOT NULL,

`price` INT(11) NOT NULL,

PRIMARY KEY (`type\_info`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `s3594800`.`order\_infor`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `s3594800`.`order\_infor` (

`Info\_id` INT(11) NOT NULL AUTO\_INCREMENT,

`order\_id` INT(11) NULL DEFAULT NULL,

`item\_id` INT(11) NULL DEFAULT NULL,

`customization\_id` INT(11) NULL DEFAULT NULL,

`type\_info` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`Info\_id`),

INDEX `fk\_Order\_Infor\_item1` (`item\_id` ASC),

INDEX `fk\_Order\_Infor\_order1` (`order\_id` ASC),

INDEX `fk\_order\_infor\_customization1` (`customization\_id` ASC),

INDEX `fk\_order\_infor\_Type1` (`type\_info` ASC),

CONSTRAINT `fk\_Order\_Infor\_item1`

FOREIGN KEY (`item\_id`)

REFERENCES `s3594800`.`item` (`item\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Order\_Infor\_order1`

FOREIGN KEY (`order\_id`)

REFERENCES `s3594800`.`order` (`order\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_order\_infor\_Type1`

FOREIGN KEY (`type\_info`)

REFERENCES `s3594800`.`type` (`type\_info`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_order\_infor\_customization1`

FOREIGN KEY (`customization\_id`)

REFERENCES `s3594800`.`customization` (`customization\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `s3594800`.`transact`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `s3594800`.`transact` (

`transact\_id` INT(11) NOT NULL AUTO\_INCREMENT,

`date\_purchase` DATE NULL DEFAULT NULL,

`received` VARCHAR(1) NULL DEFAULT NULL,

`purchaseType` VARCHAR(45) NULL DEFAULT NULL,

`cc\_num` VARCHAR(16) NULL DEFAULT NULL,

`paypal\_id` VARCHAR(45) NULL DEFAULT NULL,

`order\_order\_id` INT(11) NOT NULL,

`cc\_exdate` VARCHAR(5) NULL DEFAULT NULL,

PRIMARY KEY (`transact\_id`),

INDEX `fk\_transact\_order1\_idx` (`order\_order\_id` ASC),

CONSTRAINT `fk\_transact\_order1`

FOREIGN KEY (`order\_order\_id`)

REFERENCES `s3594800`.`order` (`order\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

**INSERT DATA**

**Customer**

INSERT INTO customer1 (surname, lastname, DOB, address, surburb, state, postcode)

VALUES ('Cardinal', 'Tom B. Erichsen', '1997/08/07', '2 Kingsley Pl', 'Delahey', 'VIC', '3011');

INSERT INTO customer1 (surname, lastname, DOB, address, surburb, state, postcode)

VALUES ('Truc', 'Mike', '1989/01/01', '2 Kingsley Pl', 'Delahey', 'VIC', '3011');

INSERT INTO customer1 (surname, lastname, DOB, address, surburb, state, postcode)

VALUES ('Mike', 'Chris', '1989/01/01', '2 Kingsley Pl', 'St Albans', 'VIC', '3037');

INSERT INTO customer1 (surname, lastname, DOB, address, surburb, state, postcode)

VALUES ('sthopier', 'Mike', '1989/01/01', '2 Kingsley Pl', 'Delahey', 'VIC', '3021');

INSERT INTO customer1 (surname, lastname, DOB, address, surburb, state, postcode)

VALUES ('Mike', 'Tim', '1989/01/01', '2 Kingsley Pl', 'Delahey', 'VIC', '3037');

INSERT INTO customer1 (surname, lastname, DOB, address, surburb, state, postcode)

VALUES ('Gleen', 'Mike', '1989/01/01', '2 Kingsley Pl', 'Melbourne', 'VIC', '3000');

INSERT INTO customer1 (surname, lastname, DOB, address, surburb, state, postcode)

VALUES ('Mike', 'Ly', '1989/01/01', '2 Kingsley Pl', 'Fooscray', 'VIC', '3059');

INSERT INTO customer1 (surname, lastname, DOB, address, surburb, state, postcode)

VALUES ('Truc', 'Mike', '1989/01/01', '2 Kingsley Pl', 'Delahey', 'VIC', '3037');

Insert into type (type\_info, size, price) values ('LogoS','100','1');

Insert into type values ('LogoM','200','1');

Insert into type values ('LogoL','300','1');

Insert into type values ('TextS','100','1');

Insert into type values ('TextM','200','2');

Insert into type values ('TextL','300','3');

Insert into type values ('ImgS','100','1');

Insert into type values ('ImgM','200','2');

Insert into type values ('ImgL','300','3');

**customization**

insert into customization (data) values ('test');

insert into customization (data) values ('school');

insert into customization (data) values ('database');

insert into customization (data) values ('data management');

insert into customization (data) values ('funny');

insert into customization (data) values ('funnies');

insert into customization (data) values ('rock');

**item**

insert into item (name, style, type, sleeves\_type, size, color, price) values ('shirt', 'long-tee', 't-shirt', 'short', 'S', 'red', '7');

insert into item (name, style, type, sleeves\_type, size, color, price) values ('shirt', 'T-bar', 't-shirt', 'long', 'M', 'red', '8');

insert into item (name, style, type, sleeves\_type, size, color, price) values ('shirt', 'normal', 't-shirt', 'middle', 'L', 'red', '9');

insert into item (name, style, type, sleeves\_type, size, color, price) values ('shirt', 'T-bar', 't-shirt', 'long', 'S', 'red', '7');

insert into item (name, style, type, sleeves\_type, size, color, price) values ('shirt', 'oversize', 't-shirt', 'short', 'M', 'red', '8');

insert into item (name, style, type, sleeves\_type, size, color, price) values ('shirt', 'T-bar', 't-shirt', 'short', 'L', 'red', '9');

**order**

insert into `order` (date\_order, quantity, total\_price, customer\_id) values ('2017/10/20','10','100','1');

insert into `order` (date\_order, quantity, total\_price, customer\_id) values ('2017/10/20','20','200','6');

insert into `order` (date\_order, quantity, total\_price, customer\_id) values ('2017/10/20','30','300','4');

insert into `order` (date\_order, quantity, total\_price, customer\_id) values ('2017/10/20','40','400','5');

insert into `order` (date\_order, quantity, total\_price, customer\_id) values ('2017/10/20','50','500','1');

insert into `order` (date\_order, quantity, total\_price, customer\_id) values ('2017/10/20','60','600','2');

**order\_infor**

insert into order\_infor (order\_id, item\_id, customization\_id, type\_info) values ('1','1','1','LogoS');

insert into order\_infor (order\_id, item\_id, customization\_id, type\_info) values ('2','5','2','LogoS');

insert into order\_infor (order\_id, item\_id, customization\_id, type\_info) values ('3','2','3','LogoS');

insert into order\_infor (order\_id, item\_id, customization\_id, type\_info) values ('5','4','4','TextL');

insert into order\_infor (order\_id, item\_id, customization\_id, type\_info) values ('7','6','5','ImgM');

insert into order\_infor (order\_id, item\_id, customization\_id, type\_info) values ('4','7','6','LogoS');

insert into order\_infor (order\_id, item\_id, customization\_id, type\_info) values ('3','2','7','LogoS');

insert into order\_infor (order\_id, item\_id, customization\_id, type\_info) values ('10','1','8','TextL');

insert into order\_infor (order\_id, item\_id, customization\_id, type\_info) values ('13','1','7','ImgM');

**Transact**

insert into transact (date\_purchase, received, purchaseType, cc\_num, paypal\_id, order\_order\_id, cc\_exdate) values ('2017/10/20','Y','bank','','','1','');

insert into transact (date\_purchase, received, purchaseType, cc\_num, paypal\_id, order\_order\_id, cc\_exdate) values ('2017/10/20','Y','creditcard','0234567894561235','','1','07/20');

insert into transact (date\_purchase, received, purchaseType, cc\_num, paypal\_id, order\_order\_id, cc\_exdate) values ('2017/10/20','N','paypal','','test@gmail.com','3','');

insert into transact (date\_purchase, received, purchaseType, cc\_num, paypal\_id, order\_order\_id, cc\_exdate) values ('2017/10/20','N','bank','','','1','');

insert into transact (date\_purchase, received, purchaseType, cc\_num, paypal\_id, order\_order\_id, cc\_exdate) values ('2017/10/20','Y','bank','','','1','');

**Aler table**

After created the table and test. I noticed a few of shortage. Therefore, I used these commands to fix the table so it is correct with the models as well as makes sense. I used alter because it is only minor problem and it is not worth to drop and re-create everything.

**I noticed that it should have a column to save the stocked shirt so I used these command to add another column and update its information.**

alter table item

add column stocked\_quantity INT;

update item

set stocked\_quantity = ‘100’

where color = ‘red’;

**I forgot to add these column and mistakenly put the wrong name when created the table in customer1 table so I add them into using these command.**

alter table customer1

add column email varchar(45);

alter table customer1

add column phone varchar(11);

UPDATE customer1

SET email = 'test@gmail.com'

WHERE address = '2 Kingsley Pl';

UPDATE customer1

SET phone = '0392183492'

WHERE address = '2 Kingsley Pl';

UPDATE `transact`

SET date\_purchase = Null

WHERE received = 'N';

alter table `s3594800`.`customer1`

change column `lastname` `firstname` varchar(45);

**Add a column to order table to handle order cancelization.**

alter table `order`

add column cancel VARCHAR(1);

update `order`

set cancel = 'Y'

where customer\_id = '6';

**Query:**

1. //List customer transaction and their payment type

select concat(c.surname,' ',c.firstname) as CustomerName, tr.purchaseType

from customer1 c JOIN `order` o ON c.customer\_id = o.customer\_id JOIN transact tr on o.order\_id = tr.order\_order\_id

1. //Quantity ordered

select concat(c.surname,' ',c.firstname) as CustomerName, o.quantity, o.total\_price

from customer1 c JOIN `order` o ON c.customer\_id = o.customer\_id

1. //List cancelation order

select concat(c.surname,' ',c.firstname) as CustomerName

from customer1 c JOIN `order` o ON c.customer\_id = o.order\_id

where o.cancel = 'Y';