



SCHOOL OF COMPUTING  
FACULTY OF ENGINEERING  
UNIVERSITY TECHNOLOGY MALAYSIA  
DATABASE  
(SCSD2523)

SEMESTER 1 2018/2019  
GROUP PROJECT  
TITLE: KTDI PHOTOSTAT SHOP PRINTING SYSTEM

By

ONG LE FOO	(A17CS0189) (970401-07-5431)	- GROUP LEADER
TAN SEE JOU	(A17CS0218) (970205-26-5254)	
TAN SHI XUAN	(A17CS0219) (971110-01-7631)	

SECTION 07  
LECTURER:  
DR AHMAD FADHIL YUSOF

## CONTENTS

1.0 PROJECT PROPOSAL .....	3
1.1 PROJECT TITLE .....	3
1.2 INTRODUCTION .....	3
1.3 PROBLEM STATEMENT .....	4
1.4 MISSION STATEMENT .....	4
1.5 MISSION OBJECTIVE .....	4
1.6 SYSTEM BOUNDARY .....	5
1.7 GANTT CHART .....	7
2.0 DATABASE PROJECT .....	9
2.1 TRANSACTION REQUIREMENT .....	9
2.2 DATA REQUIREMENT .....	10
2.3 CROSS REFERENCE ANALYSIS .....	11
3.0 CONCEPTUAL DATABASE DESIGN .....	12
3.1 ERD FOR CONCEPTUAL DESIGN .....	12
3.2. DATA DICTIONARY .....	13
3.2.1 Data dictionary that documents the entities for the system.....	13
3.2.2 Data dictionary for showing the description of relationship.....	13
3.2.3 Data dictionary showing the description of relationship .....	14
3.3 SYSTEM'S FUNCTIONAL REQUIREMENTS .....	15
3.4 LOGICAL DFD .....	16
4.0 LOGICAL DATABASE DESIGN .....	17
4.1 LOGICAL ERD .....	17
4.2 NORMALIZATION .....	18
4.3. DATA DICTIONARY .....	19
4.2.1 Data dictionary that documents the entities for the system.....	19
4.2.2 Data dictionary for showing the description of relationship.....	20
4.2.3 Data dictionary showing the description of relationship .....	22
4.4 SQL STATEMENT .....	22
4.4.1 Data Definition Language .....	23
4.4.2 Data Manipulation Language .....	35

# 1.0 PROJECT PROPOSAL

## 1.1 PROJECT TITLE

Database Planning and System for Kolej Tun Dr Ismail (KTDI) Photostat Shop.

## 1.2 INTRODUCTION

The system that we choose to investigate is the system of the KTDI photostat shop. The reason that we pick this shop is because that their system is not fully digitalized. Besides, KTDI photostat shop is also very convenient and near for us to investigate because we are also living in KTDI. KTDI photostat shop has provided a lot of service, for example, photo stating, printing, laminate, binding, taping, scanning and etc. The system that we choose to investigate are the printing services. The current system of printing service in KTDI photostat shop is customer manually using the computer provided by the owner. Customer need to use their own pen-drive and print out manually. We had interview with the shop owner to discuss about their system. After discussing with the owner, it is found that there are some problem on their system. Besides, the owner had also gave us some idea on how to improve their system in order to achieve a better business.



## 1.3 PROBLEM STATEMENT

We had interviewed the shop owner regarding the problem of their current system.

- Weather problem inconvenience for customer to reach
- Time wasting for customer to queue up
- Customer unfamiliar with the PC provided, for example this may cause customer to print out wrongly
- Owner PC may have high risk of getting virus

## 1.4 MISSION STATEMENT

- To create a new system for the printing service of the KTDI photostat shop.

## 1.5 MISSION OBJECTIVE

- To reduce the risk of owner PC from getting virus.
- To prevent loss of customer due to bad weather and long queue.
- To extend the source of customer
- To provide delivery service for customer.
- To allow customer to have cashless payment by paying digitally.

## 1.6 SYSTEM BOUNDARY

### *Customer*

<i>Customer ID</i>	<i>Name</i>	<i>Password</i>	<i>Contact number</i>	<i>Address</i>	<i>Email Address</i>	<i>Wallet ID</i>	<i>Order ID</i>
--------------------	-------------	-----------------	-----------------------	----------------	----------------------	------------------	-----------------

### *PrintPay*

<i>Wallet ID</i>	<i>Top-Up ID</i>	<i>Payment ID</i>	<i>Balance</i>
------------------	------------------	-------------------	----------------

### *Top-Up*

<i>Top-Up ID</i>	<i>Top-Up Date</i>	<i>Amount</i>
------------------	--------------------	---------------

### *Payment*

<i>Payment ID</i>	<i>Payment Date</i>	<i>Amount</i>
-------------------	---------------------	---------------

### *Order*

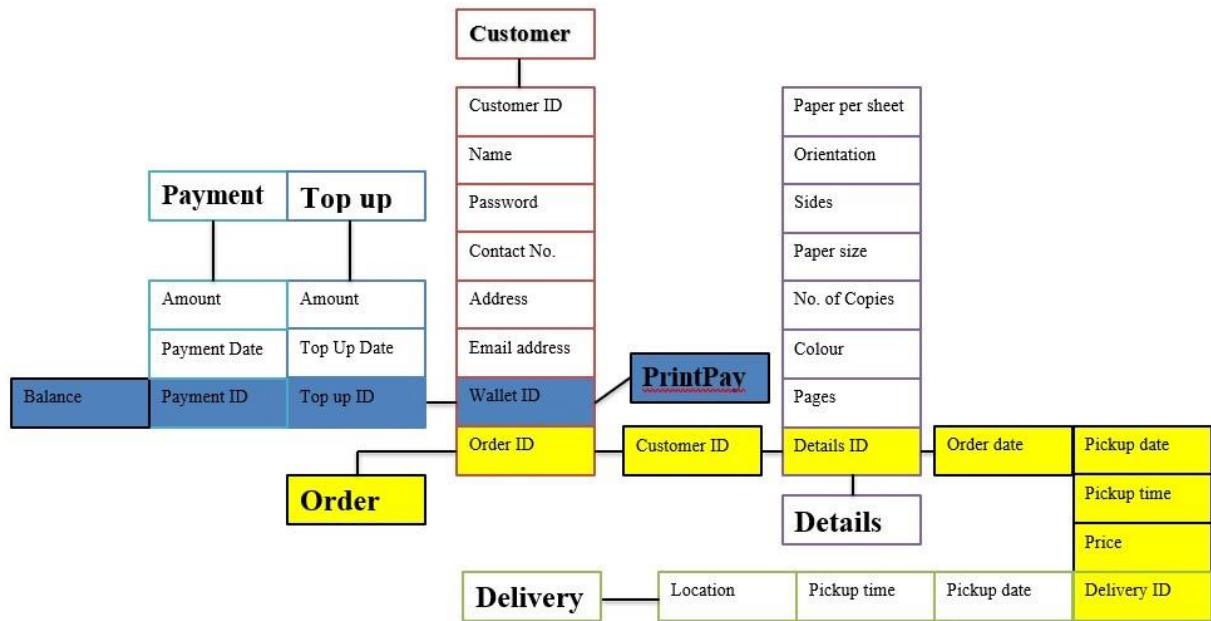
<i>Order ID</i>	<i>Customer ID</i>	<i>Details ID</i>	<i>Order Date</i>	<i>Pickup Date</i>	<i>Pickup Time</i>	<i>Price</i>	<i>Delivery ID</i>
-----------------	--------------------	-------------------	-------------------	--------------------	--------------------	--------------	--------------------

### *Details*

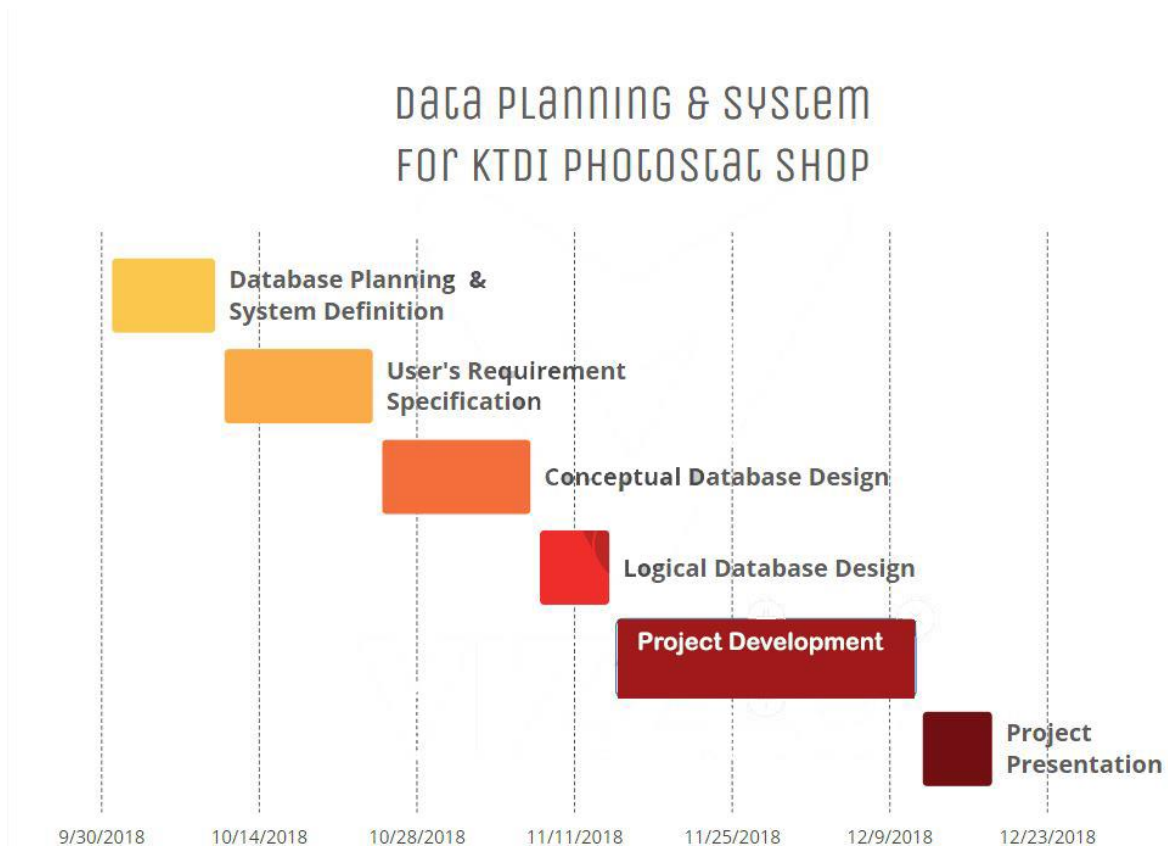
<i>Details ID</i>	<i>Pages</i>	<i>Color</i>	<i>Number of Copies</i>	<i>Paper Size</i>	<i>Sides</i>	<i>Orientation</i>	<i>Page per Sheet</i>
-------------------	--------------	--------------	-------------------------	-------------------	--------------	--------------------	-----------------------

### *Delivery*

<i>Delivery ID</i>	<i>Pickup Date</i>	<i>Pickup Time</i>	<i>Location</i>
--------------------	--------------------	--------------------	-----------------



## 1.7 GANTT CHART



Description	End Date	Start Day	Duration (Days)
P1	1/10/18	10/10/18	10
P2	11/10/18	24/10/18	14
P3	25/10/18	7/11/18	14
P4	8/11/18	14/11/18	7
P5	15/11/18	11/12/18	28
P6	12/12/18	18/12/18	7

#### P1 – Database Planning & System Definition

- ✓ Create the mission statement, mission objectives for the database system.
- ✓ Define the system boundary and major views for the database system.

#### P2 – User's Requirement Specification

- ✓ Gather more details on the user views and any general requirements for the database system
- ✓ Decide on how to manage the user views of the new database system

#### P3 - Conceptual Database Design

- ✓ Based on requirements identified from P2, identify entities and attributes for each entity.
- ✓ Determine all possible relationships between entities and the multiplicities for each relationship.
- ✓ Represent the above in ERD and Include enhance ERD features wherever possible.

#### P4 - Logical Database Design

- ✓ Transform the conceptual ERD produced in P3 into a logical ERD and derive relations schema from the logical ERD.
- ✓ Perform normalization up till BCNF to these relations.
- ✓ Draw the final logical ERD to represent the BCNF relations schema produced and Validate logical ERD with the system's transaction requirements
- ✓ Update the data dictionary based on the normalized relations produced from above

#### P5 – Project Implementation

- ✓ Developed the project by building a simple application.

#### P6- Project Demo

- ✓ Presenting and demo the project.



## 2.0 DATABASE PROJECT

### 2.1 TRANSACTION REQUIREMENT

Entity	Data	Data Entry	Data Update	Data Deletion	Data Query
Staff	staff_id	Enter personal details by staff	Update personal details by staff	Delete staff's details by owner	List staff's details by owner
	staff_name				
	tel_no				
Customer	customer_id	Enter personal details by customer	Update customer's details by customer	Delete customer's details by customer	Query on customer's data by owner
	customer_name				
	tel_no				
	email				
	address				
E-Wallet	customer_id	Enter customer ID by customer.	Update wallet balance by customer.	E-wallet details delete by customers.	Display balance of customer's e-wallet
	balance				
Order	order_ID	Enter order information by customer.	Update order information by customer	Delete order details by customer	List order details by owner
	customer_ID				
	order_date				
	price				
Top-up	top_up_id	Enter top up details by customer.	Update top up details by customer	Delete top up details by customers	List top up details by customer.
	date				
	amount				

## 2.2 DATA REQUIREMENT

ENTITY	DESCRIPTION
Staff	The staff include the data store of staff_ID, staff_Name, and tel_No. The staff_ID are unique for each staff.
Customer	The staff include the data store of staff_ID, staff_Name, and tel_No. The staff_ID are unique for each staff.
E-wallet	When customer register an account from the system, customer needs to enter the name, telephone, email and address. The customer ID is automatically generated by the system and it is an unique identifier.
Order	When customers place order, they need to enter order details for each order. The data stored of order includes order ID, which is generated by the system, customer ID, order date and total price.
Top-Up	Once a customer made a top-up into the E-wallet. The data information of top_up_ID, date, and amount will be stored. top_up_ID is unique.

## 2.3 CROSS REFERENCE ANALYSIS

Transaction / Relation	Deliver				Place				Make				Has				Pay			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Staff		X																X		
Customer	X		X	X	X				X		X	X			X		X			
E-Wallet									X		X	X					X			
Order	X		X	X																

I: INSERT

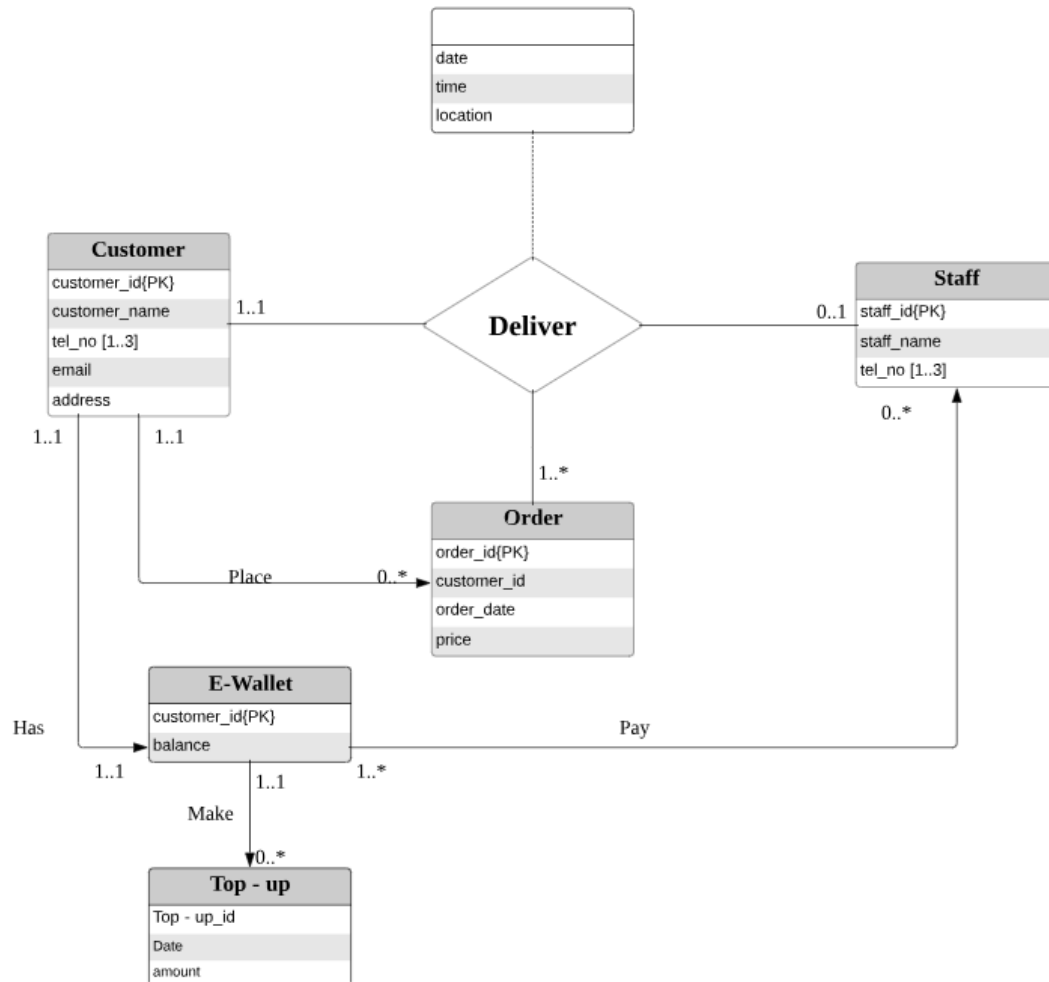
R: READ

U: UPDATE

D: DELETE

## 3.0 CONCEPTUAL DATABASE DESIGN

### 3.1 ERD FOR CONCEPTUAL DESIGN



## 3.2. DATA DICTIONARY

### 3.2.1 Data dictionary that documents the entities for the system

ENTITY	DESCRIPTION	ALIASES	OCCURRENCE
Staff	General term describing all staff employed by KTDI printing shop.	staff	Staff who are managing the system and provided the printing services
Customer	General term describing all users for the KTDI Printing System	customer	Each customer has their own account and can make order through it.
E-wallet	General term describing the function of top-up system.	e_wallet	Each E-wallet has only one owner and is used to make payment.
Order	General term describing the order or request place by customer.	order	Order is made by customer and take note by staff to provide service.
Top Up	General term describing the top up of customer	top_up	Top up is made by customer to top up money into the e-wallet

### 3.2.2 Data dictionary for showing the description of relationship.

Entity	Attribute	Data type	Not Null?	Constraints	Description
Staff	Staff_id	Varchar2	No	Primary key	A unique identification number for each staff
	staff_name	Varchar2	No	-	Name of the staff
	tel_no	Number	Yes	-	Staff contact number

Customer	Customer_id	Varchar2	No	Primary key	A unique identification number for each customer
	customer_name	Varchar2	No	-	Name of the customer
	tel_no	Number	No	-	Contact number of the customer
	email	Varchar2	No	-	Email address of the customer
	address	Varchar2	Yes	-	Customer's address
E-wallet	customer_id	Varchar2	No	Foreign key	A unique identification number for each item
	balance	Number	No	-	Name of the item
Order	order_id	Varchar2	No	Primary key	Order Unique identification
	customer_id	Varchar2	No	Foreign key	Customer Unique Identification
	order_date	date	No	-	Date of the order made
	price	number	No	-	Total price of the order.
Top up	top_up_id	Varchar2	No	Primary Key	Top up id unique identification
	date	date	No	-	Date of the top up made
	amount	number	No	-	Amount of top up made

### 3.2.3 Data dictionary showing the description of relationship

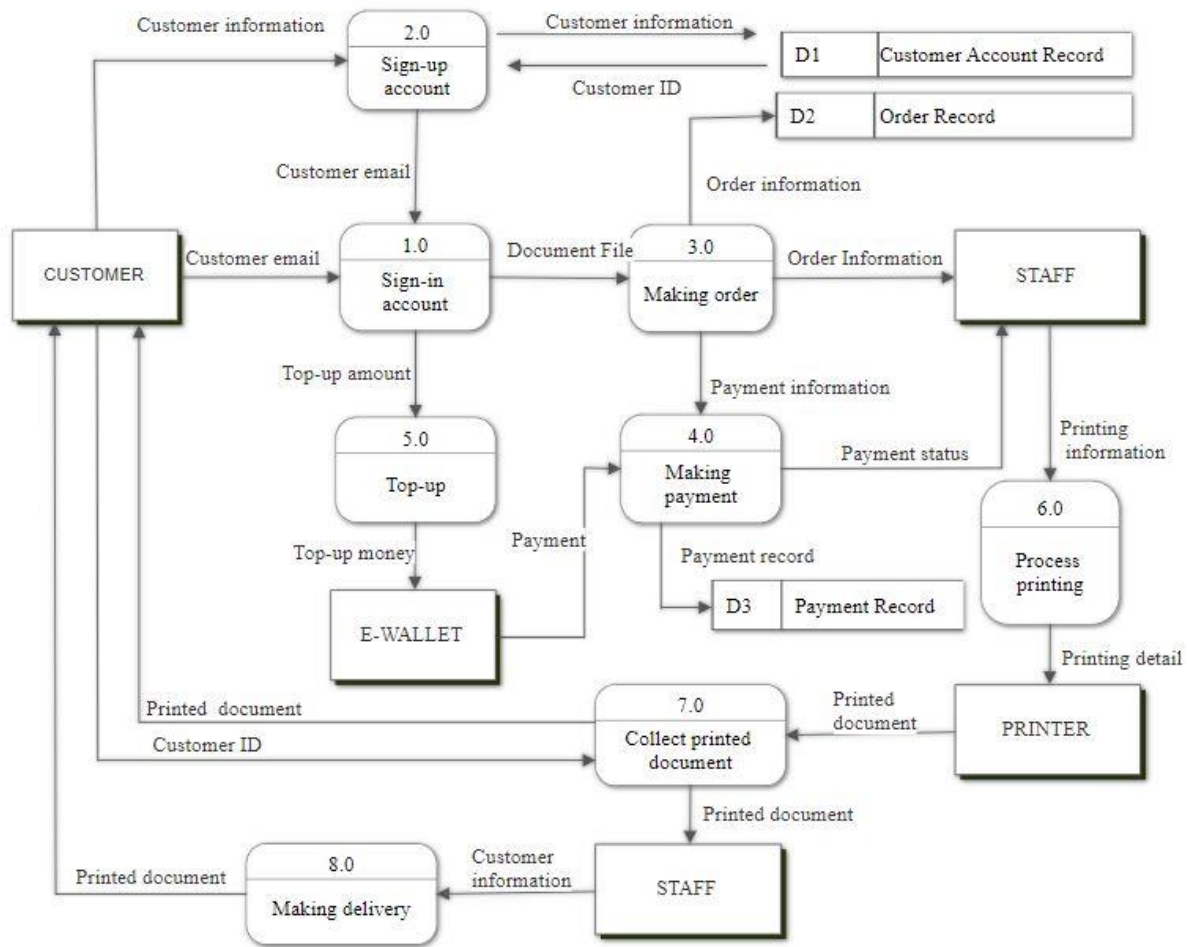
ENTITY NAME	MULTIPLICITY	RELATIONSHIP	ENTITY NAME	MULTIPLICITY
Staff	0..1	deliver	Customer	1..1

Staff	0..1	deliver	Order	1..*
Customer	1..1	has	E-wallet	1..1
Customer	1..1	place	Order	0..*
E-wallet	1..*	pay	Staff	0..1
E-wallet	1..1	make	Top Up	0..*

### 3.3 SYSTEM'S FUNCTIONAL REQUIREMENTS

The system will require internet connection in order to run. User have to download and install the applications before they can use it. The system user has to register an account before they can start using the system. The user will need to insert their personal information and all of the data information will be stored in the entity of Customer. User have to upload their document before making their order, and order cannot be placed without paying first. User have to use their E-wallet to pay for their order. User can modify their order before paying it but cannot cancel after, but the Staff can manage the order after the order has been made. User can top-up and view their own E-wallet but cannot withdraw it, and the staff member won't have access to it. The E-wallet will limit the user/customer from placing the order if they have insufficient balance to pay for their order.

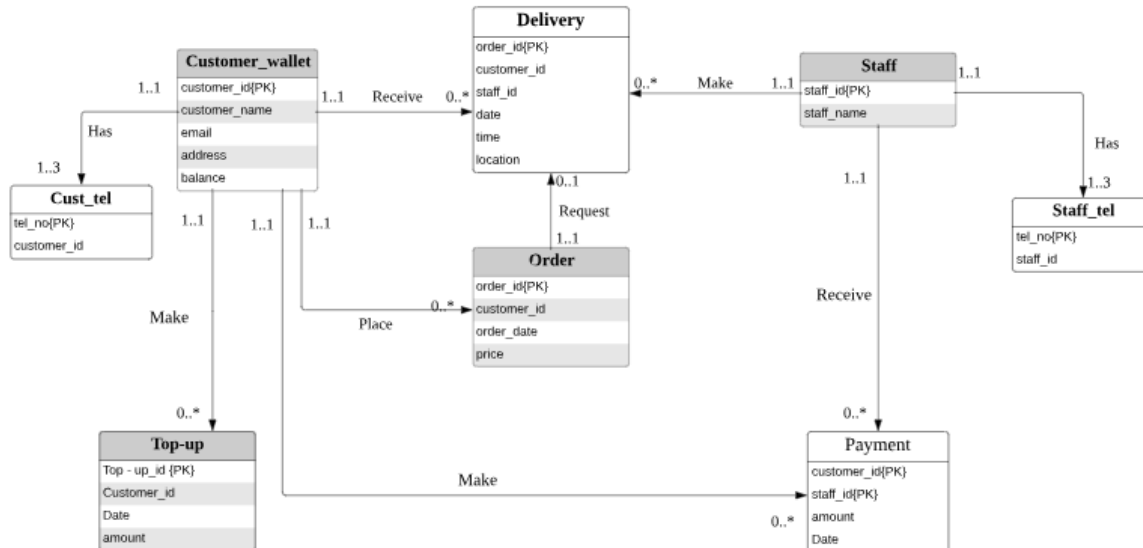
### 3.4 LOGICAL DFD





## 4.0 LOGICAL DATABASE DESIGN

### 4.1 LOGICAL ERD



#### Relations Schema from logical ERD

Customers (customer\_id, customer\_name, email, address)

Cust\_tel (tel\_no, customer\_id)

FK: customer\_id reference to Customers table(customer\_id).

Top-up (customer\_id, Date, amount)

Delivery (order\_id, customer\_id, staff\_id, date, time, location)

FK: customer\_id reference to Customers table(customer\_id).

FK: staff\_id reference to Staff table(staff\_id).

Order (order\_id, customer\_id, order\_date, price)

FK: customer\_id reference to Customers table(customer\_id).

Staff (staff\_id, staff\_name)

Staff\_tel (tel\_no, staff\_id)

FK: staff\_id reference to Staff table(staff\_id)

Payment (customer\_id, staff\_id, amount, Date)

## 4.2 NORMALIZATION

### Functional Dependency

Customers table

FD1: customer\_id -> customer\_name, tel\_no, email, address

Top-Up table

FD1: top\_up\_id -> amount, date,

Delivery table

FD1: order\_id->customer\_id, staff\_id, date, time, location

Order table

FD1: order\_id -> customer\_id, order\_date, price

Staff table

Staff\_id -> staff\_name, tel\_no

Payment table

customer\_id, staff\_id, amount, Date

### 1st Normal Form

Customers (customer\_id, customer\_name, tel\_no, email, address)

Cust\_tel (tel\_no, customer\_id)

FK: customer\_id reference to Customers table(customer\_id).

Top-up (customer\_id, Date, amount)

Delivery (order\_id, customer\_id, staff\_id, date, time, location)

FK: customer\_id reference to Customers table(customer\_id).

FK: staff\_id reference to Staff table(staff\_id).

Order (order\_id, customer\_id, order\_date, price)

FK: customer\_id reference to Customers table(customer\_id).

Staff (staff\_id, staff\_name)

Staff\_tel (tel\_no, staff\_id)

FK: staff\_id reference to Staff table(staff\_id)

Payment (customer\_id, staff\_id, amount, Date)

### 2nd Normal Form

There is no second normal form as there is no partial dependency in all of the relations

### 3rd Normal Form

There is no third normal form as there is no transitive dependency in all of the relations

## 4.3. DATA DICTIONARY

### 4.2.1 Data dictionary that documents the entities for the system

ENTITY	DESCRIPTION	ALIASES	OCCURRENCE
Staff	General term describing all staff employed by KTDI printing shop.	staff	Staff who are managing the system and provided the printing services
Customer	General term describing all users for the KTDI Printing System	customer	Each customer has their own account and can make order through it.
Order	General term describing the order or request place by customer.	order	Order is made by customer and take note by staff to provide service.
Top-Up	General term that describe the top-up made by customer into the E-wallet	top_up	Customer can make many top-up into their e-wallet.

Delivery	General term describing delivery of order from staff to customer	delivery	Staff make delivery of an order to a customer.
Payment	General term describing the payment detail made by customer to staff	payment	Customer will use their E-wallet to make their payment of the order to staff

#### 4.2.2 Data dictionary for showing the description of relationship.

Entity	Attribute	Data type	Not Null?	Constraints	Description
Staff	Staff_id	Varchar2	No	Primary key	A unique identification number for each staff
	staff_name	Varchar2	No	-	Name of the staff
Staff_tel	tel_no	Number	No	Primary key	Contact number of the staff
	staff_id	Varchar2	No	Foreign key	A unique identification number for each staff
Customer	Customer_id	Varchar2	No	Primary key	A unique identification number for each customer
	customer_name	Varchar2	No	-	Name of the customer
	Tel_no	Number	No	-	Contact number of the customer
	email	Varchar2	No	-	Email address of the customer
	address	Varchar2	Yes	-	Customer's address
	balance	number	No	-	Customer's balance

Cust_tel	tel_no	Number	No	Primary key	Contact number of the customer
	Customer_id	Varchar2	No	Foreign key	A unique identification number for each customer
Order	order_id	Varchar2	No	Primary key	Order Unique identification
	customer_id	Varchar2	No	Foreign key	Customer Unique Identification
	order_date	date	No	-	Date of the order made
	price	number	No	-	Total price of the order.
Top-up	top_up_id	Varchar2	No	Primary Key	A unique identification number for each customer
	customer_id	Varchar2	No	Foreign key	A unique identification number for each customer
	date	date	No	-	Date of top-up made
	amount	number	No	-	Amount of top-up
Delivery	order_id	Varchar2	No	Primary key	Order Unique identification
	customer_id	Varchar2	No	Foreign key	A unique identification number for each customer
	staff_id	Varchar2	No	Foreign key	A unique identification number for each staff
	date	date	No	-	Date of delivery
	time	Varchar2	No	-	Time of delivery made
	location	Varchar2	No	-	Location of delivery made
Payment	customer_id	Varchar2	No	Primary key	A unique identification number for each customer
	Staff_id	Varchar2	No	Primary Key	A unique identification number for each staff

	amount	number	No	-	Amount of payment
	date	date	No	-	Date of payment

#### 4.2.3 Data dictionary showing the description of relationship

ENTITY NAME	MULTIPLICITY	RELATIONSHIP	ENTITY NAME	MULTIPLICITY
Customer	1..1	Make	Top-up	0..*
Customer	1..1	Place	Order	0..*
Customer	0..1	Receive	Delivery	0..*
Customer	1..1	Make	Payment	0..*
Customer	1..1	Has	Cust_tel	1..3
Order	1..1	Request	Delivery	0..1
Staff	1..1	Make	Delivery	0..*
Staff	1..1	Receive	Payment	0..*
Staff	1..1	Has	Staff_tel	1..3

## 4.4 SQL STATEMENT

This will be included 2 sections, section one for DDL and section 2 for DML for each transaction requirement.

## 4.4.1 Data Definition Language

```
CREATE TABLE Staff(  
    staff_id VARCHAR2(10) PRIMARY KEY,  
    staff_name VARCHAR2(100));  
  
--FIRST QUERY  
INSERT INTO Staff  
    (staff_id, staff_name)  
VALUES  
    ('S0001', 'Jacky');  
  
INSERT INTO STAFF (staff_id, staff_name)  
VALUES ('S0002', 'Mary');  
INSERT INTO STAFF (staff_id, staff_name)  
VALUES ('S0003', 'Chian');  
INSERT INTO STAFF (staff_id, staff_name)  
VALUES ('S0004', 'Denny');  
  
CREATE TABLE Staff_tel(  
    tel_no NUMBER PRIMARY KEY,  
    staff_id VARCHAR2(10) REFERENCES Staff (staff_id));  
  
INSERT INTO Staff_tel(tel_no, staff_id)  
VALUES (0111122334, 'S0001');  
INSERT INTO Staff_tel(tel_no, staff_id)  
VALUES (0125566987, 'S0002');  
INSERT INTO Staff_tel(tel_no, staff_id)  
VALUES (0178896523, 'S0003');  
INSERT INTO Staff_tel(tel_no, staff_id)  
VALUES (0195685324, 'S0004');  
  
CREATE TABLE Customer(  
    customer_id VARCHAR2(10) PRIMARY KEY,  
    customer_name VARCHAR2(100),  
    email VARCHAR2(40),  
    address VARCHAR2(300),  
    balance NUMBER);  
  
CREATE TABLE Cust_tel(  
    tel_no NUMBER PRIMARY KEY,  
    customer_id VARCHAR2(10) REFERENCES Customer (customer_id));  
  
--2nd Query  
INSERT INTO Customer  
    (customer_id, customer_name, email, address, balance)  
VALUES  
    ('A0001', 'Tan Shi Xuan', 'tzzszzxzzz@gmail.com', '154 MA1, KTDI, UTM, 81310,  
Johor, Skudai', '10');  
  
INSERT INTO Customer  
VALUES ('A0002', 'Soh Jia Jun', 'jiajun012@gmail.com', '147 MA1, KTDI, UTM,  
81310, Johor, Skudai', 5.8);  
INSERT INTO Customer
```

```

VALUES ('A0003','Tan See Jou','seejou666@gmail.com','412 MA7, KTDI, UTM,
81310, Johor, Skudai',3.3);
INSERT INTO Customer
VALUES ('A0004','Ong Le Foo','nodenong@gmail.com','247 MA1, KTDI, UTM, 81310,
Johor, Skudai',2);
INSERT INTO Customer
VALUES ('A0005','Goh Chiang Cheng','kentkk@gmail.com','250 MA1, KTDI, UTM,
81310, Johor, Skudai',1.2);
INSERT INTO Customer
VALUES ('A0006','Muhamad JohanShah','johan87@gmail.com','222 MA1, KTDI, UTM,
81310, Johor, Skudai',0);
INSERT INTO Customer
VALUES ('A0007','Mellisa Lim','mellisa1111@gmail.com','211 MA4, KTDI, UTM,
81310, Johor, Skudai',7);
INSERT INTO Customer
VALUES ('A0008','Ibrahim Mohamed','ibrabra@hotmail.com','108 MA4, KTDI, UTM,
81310, Johor, Skudai',5.2);
INSERT INTO Customer
VALUES ('A0009','Ali Razak Zul Fikri','alirazak@yahoo.com','222 M16, KTDI,
UTM, 81310, Johor, Skudai',8.9);
INSERT INTO Customer
VALUES ('A0010','Siti Nuraliyah Zaina','aliyah9797@gmail.com','233 M27, KTDI,
UTM, 81310, Johor, Skudai',7);
INSERT INTO Customer
VALUES ('A0011','Mohamed Qatadah','QQQQ@hotmail.com','111, M15, KTDI, UTM,
81310, Johor, Skudai',7);
INSERT INTO Customer
VALUES ('A0012','Ahmad Fadil','Fadil@yahoo.com','402 M17, KTDI, UTM, 81310,
Johor, Skudai',4.3);
INSERT INTO Customer
VALUES ('A0013','Leo Tan','Leo7496@gmail.com','204 M15. KTDI, UTM, 81310,
Johor, Skudai',4.6);
INSERT INTO Customer
VALUES ('A0014','Mohammad Gala','GalaGaga@gmail.com','100 M17, KTDI, UTM,
81310, Johor, Skudai',7.8);
INSERT INTO Customer
VALUES ('A0015','Lin Jun Jie','JJLin@gmail.com','108, M16, KTDI, UTM, 81310,
Johor, Skudai',8);
INSERT INTO Customer
VALUES ('A0016','Evelyn Lim','EveeeeL@gmail.com','G01, M11, KTDI, UTM, 81310,
Johor, Skudai',0);
INSERT INTO Customer
VALUES ('A0017','Lim Hooi Hooi','Lhh@hotmail.om','112, M11, KTDI, UTM, 81310,
Johor, Skudai',0);
INSERT INTO Customer
VALUES ('A0018','Ng Sern Wei','N992laks@yahoo.com','G12, M17, KTDI, UTM,
81310, Johor, Skudai',0);
INSERT INTO Customer
VALUES ('A0019','Lee Joshua','joshualee@gmail.com','269 M15. KTDI, UTM,
81310, Johor, Skudai',50);
INSERT INTO Customer
VALUES ('A0020','Jimmy Leong','JmmyL122@gmail.com','101 M17, KTDI, UTM,
81310, Johor, Skudai',69.2);
INSERT INTO Customer
VALUES ('A0021','Ong Thien Ming','MinTea22@gmail.com','165 M16, KTDI, UTM,
81310, Johor, Skudai',10);
INSERT INTO Customer

```



```

VALUES ('A0022','Siti Azalina','12Siti@gmail.com','G11 MA7,KTDI, UTM, 81310,
Johor, Skudai',1.2);
INSERT INTO Customer
VALUES ('A0023','Puan Naliyah','naliyah1002@gmail.com','G01 MA4 KTDI,UTM,
81310, Johor, Skudai',1.6);
INSERT INTO Customer
VALUES ('A0024','Nur Zaniah','zaniah@gmail.com','209 MA6, KTDI, UTM, 81310,
Johor, Skudai',7.8);
INSERT INTO Customer
VALUES ('A0025','Noor Siti','noorsiti7@gmail.com','105 M14, KTDI, UTM, 81310,
Johor, Skudai',3);
INSERT INTO Customer
VALUES ('A0026','Ahmad Fitri','afitri8@gmail.com','122 MA1, KTDI, UTM, 81310,
Johor, Skudai',2);
INSERT INTO Customer
VALUES ('A0027','Tan Min MIng','tmm988@gmail.com','353 MA4, KTDI, UTM, 81310,
Johor, Skudai',5);
INSERT INTO Customer
VALUES ('A0028','Liew Chia Jia','lcj09@gmail.com','229, MA7, KTDI, UTM,
81310, Johor, Skudai',5.1);
INSERT INTO Customer
VALUES ('A0029','Michael Jackson','mjstar@yahoo.com','357, MA1, KTDI, UTM,
81310, Johor, Skudai',13.6);
INSERT INTO Customer
VALUES ('A0030','Charlie Puth','charlie91@hotmail.com','450 MA1, KTDI, UTM,
81310, Johor, Skudai',11.1);

INSERT INTO Cust_tel
VALUES (0116587493,'A0001');
INSERT INTO Cust_tel
VALUES (0124458796,'A0002');
INSERT INTO Cust_tel
VALUES (0129966555,'A0003');
INSERT INTO Cust_tel
VALUES (0172369874,'A0004');
INSERT INTO Cust_tel
VALUES (0198588888,'A0005');
INSERT INTO Cust_tel
VALUES (0113632333,'A0006');
INSERT INTO Cust_tel
VALUES (0117796774,'A0007');
INSERT INTO Cust_tel
VALUES (0125347859,'A0008');
INSERT INTO Cust_tel
VALUES (0125658963,'A0009');
INSERT INTO Cust_tel
VALUES (0198697456,'A0010');
INSERT INTO Cust_tel
VALUES (0111205030,'A0011');
INSERT INTO Cust_tel
VALUES (0127878564,'A0012');
INSERT INTO Cust_tel
VALUES (0178523697,'A0013');
INSERT INTO Cust_tel
VALUES (0175458963,'A0014');
INSERT INTO Cust_tel
VALUES (0112223456,'A0015');

```

```

INSERT INTO Cust_tel
VALUES (0135268974,'A0016');
INSERT INTO Cust_tel
VALUES (0114472563,'A0017');
INSERT INTO Cust_tel
VALUES (0196666325,'A0018');
INSERT INTO Cust_tel
VALUES (0125566447,'A0019');
INSERT INTO Cust_tel
VALUES (0113698521,'A0020');
INSERT INTO Cust_tel
VALUES (0147878456,'A0021');
INSERT INTO Cust_tel
VALUES (0147878457,'A0021');
INSERT INTO Cust_tel
VALUES (0173369524,'A0022');
INSERT INTO Cust_tel
VALUES (0112236020,'A0023');
INSERT INTO Cust_tel
VALUES (0114458779,'A0024');
INSERT INTO Cust_tel
VALUES (0133749856,'A0025');
INSERT INTO Cust_tel
VALUES (0147455211,'A0026');
INSERT INTO Cust_tel
VALUES (0123365632,'A0027');
INSERT INTO Cust_tel
VALUES (0123363332,'A0027');
INSERT INTO Cust_tel
VALUES (0123364432,'A0027');
INSERT INTO Cust_tel
VALUES (0179689552,'A0028');
INSERT INTO Cust_tel
VALUES (0174525444,'A0029');
INSERT INTO Cust_tel
VALUES (0112589635,'A0030');

```

```

CREATE TABLE Top_up(
    top_up_id VARCHAR2(10) PRIMARY KEY,
    top_up_date DATE,
    amount NUMBER,
    customer_id VARCHAR(10) REFERENCES Customer (customer_id));

```

--3rd Query

```

INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0028', '12-NOV-2018', 6.00, 'A0003');

INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0030', '09-NOV-2018', 10.00, 'A0002');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0029', '12-NOV-2018', 8.00, 'A0003');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0027', '16-NOV-2018', 20.00, 'A0005');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0026', '20-NOV-2018', 3.00, 'A0006');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)

```

```

VALUES ('t0025', '25-NOV-2018', 5.00, 'A0006');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0024', '3-DEC-2018', 20.00, 'A0008');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0023', '1-NOV-2018', 30.00, 'A0009');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0022', '10-NOV-2018', 20.00, 'A0009');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0021', '6-DEC-2018', 5.00, 'A0011');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0020', '20-DEC-2018', 5.00, 'A0012');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0019', '23-NOV-2018', 10.00, 'A0012');

```

```

INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0018', '03-NOV-2018', 9.00, 'A0014');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0017', '1-DEC-2018', 20.00, 'A0015');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0016', '5-NOV-2018', 10.00, 'A0015');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0015', '18-NOV-2018', 15.00, 'A0017');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0014', '16-DEC-2018', 5.00, 'A0018');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0013', '30-NOV-2018', 15.00, 'A0018');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0012', '8-DEC-2018', 3.00, 'A0020');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0011', '12-NOV-2018', 9.00, 'A0021');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0010', '8-NOV-2018', 12.00, 'A0021');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0009', '6-DEC-2018', 30.00, 'A0023');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0008', '28-NOV-2018', 5.00, 'A0024');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0007', '10-NOV-2018', 10.00, 'A0024');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0006', '26-NOV-2018', 2.00, 'A0026');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0005', '4-DEC-2018', 17.00, 'A0027');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0004', '15-DEC-2018', 10.00, 'A0027');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0003', '10-DEC-2018', 20.00, 'A0029');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0002', '21-NOV-2018', 15.00, 'A0030');
INSERT INTO Top_up (top_up_id, top_up_date, amount, customer_id)
VALUES ('t0001', '14-DEC-2018', 24.00, 'A0030');

```

```

CREATE TABLE Delivery(
    order_id VARCHAR2(10) PRIMARY KEY,
    customer_id VARCHAR(10) REFERENCES Customer (customer_id),
    staff_id VARCHAR(10) REFERENCES Staff (staff_id),
    delivery_date DATE,
    time VARCHAR(10),

```

location VARCHAR(50));

```
INSERT INTO Delivery
VALUES ('OD0001', 'A0001', 'S0001', '06-DEC-2018', '1200', 'Faculty
Computing');
INSERT INTO Delivery
VALUES ('OD0003', 'A0008', 'S0002', '06-DEC-2018', '1700', 'Faculty
Computing');
INSERT INTO Delivery
VALUES ('OD0004', 'A0014', 'S0001', '06-DEC-2018', '1900', 'Arked Angkasa');
INSERT INTO Delivery
VALUES ('OD0005', 'A0007', 'S0004', '06-DEC-2018', '1630', 'Arked Angkasa');
INSERT INTO Delivery
VALUES ('OD0006', 'A0010', 'S0004', '06-DEC-2018', '1730', 'Arked Cengal');
INSERT INTO Delivery
VALUES ('OD0007', 'A0002', 'S0003', '07-DEC-2018', '1200', 'Arked Meranti');
INSERT INTO Delivery
VALUES ('OD0008', 'A0010', 'S0002', '07-DEC-2018', '1530', 'Arked Angkasa');
INSERT INTO Delivery
VALUES ('OD0009', 'A0011', 'S0001', '07-DEC-2018', '1400', 'Faculty
Computing');
INSERT INTO Delivery
VALUES ('OD0010', 'A0007', 'S0002', '07-DEC-2018', '1500', 'Arked Meranti');
INSERT INTO Delivery
VALUES ('OD0011', 'A0005', 'S0002', '07-DEC-2018', '1130', 'Arked Angkasa');
INSERT INTO Delivery
VALUES ('OD0012', 'A0003', 'S0004', '08-DEC-2018', '1800', 'Faculty
Computing');
INSERT INTO Delivery
VALUES ('OD0013', 'A0016', 'S0004', '08-DEC-2018', '1740', 'Faculty
Computing');
INSERT INTO Delivery
VALUES ('OD0023', 'A0015', 'S0001', '10-DEC-2018', '2000', 'MA1, KTDI');
INSERT INTO Delivery
VALUES ('OD0024', 'A0022', 'S0001', '10-DEC-2018', '2000', 'MA1, KTDI');

INSERT INTO Delivery
VALUES ('OD0027', 'A0008', 'S0004', '10-DEC-2018', '1720', 'Mak Ngah, KTDI,
UTM');
INSERT INTO Delivery
VALUES ('OD0028', 'A0003', 'S0001', '10-DEC-2018', '1630', 'M27, KTDI, UTM');
INSERT INTO Delivery
VALUES ('OD0036', 'A0009', 'S0003', '11-DEC-2018', '1900', 'G18, KRP, UTM');
INSERT INTO Delivery
VALUES ('OD0038', 'A0023', 'S0002', '11-DEC-2018', '2000', 'S19, KTC, UTM');
INSERT INTO Delivery
VALUES ('OD0040', 'A0001', 'S0001', '11-DEC-2018', '1500', 'Faculty
Computing');
INSERT INTO Delivery
VALUES ('OD0042', 'A0005', 'S0001', '12-DEC-2018', '1330', 'H04, KTF, UTM');
INSERT INTO Delivery
VALUES ('OD0043', 'A0028', 'S0004', '12-DEC-2018', '1400', 'K11, KRP, UTM');
INSERT INTO Delivery
VALUES ('OD0049', 'A0013', 'S0002', '12-DEC-2018', '1230', 'FABU');
INSERT INTO Delivery
VALUES ('OD0050', 'A0019', 'S0002', '12-DEC-2018', '1200', 'FABU');
```

```
CREATE TABLE Ordering(  
    order_id VARCHAR2(10) PRIMARY KEY,  
    customer_id VARCHAR2(10) REFERENCES Customer (customer_id),  
    order_date DATE,  
    price NUMBER);
```

--4th Query

```
INSERT INTO Ordering  
(order_id, customer_id, order_date, price)  
VALUES  
( 'OD0004', 'A0014', '06-DEC-2018', 4.1);  
  
INSERT INTO Ordering  
VALUES ('OD0001', 'A0001', '06-Dec-2018', 3.2);  
INSERT INTO Ordering  
VALUES ('OD0002', 'A0002', '06-Dec-2018', 6.0);  
INSERT INTO Ordering  
VALUES ('OD0003', 'A0008', '06-Dec-2018', 1.5);  
INSERT INTO Ordering  
VALUES ('OD0005', 'A0007', '06-Dec-2018', 1.5);  
INSERT INTO Ordering  
VALUES ('OD0006', 'A0010', '06-Dec-2018', 1.3);  
INSERT INTO Ordering  
VALUES ('OD0007', 'A0002', '07-Dec-2018', 2);  
INSERT INTO Ordering  
VALUES ('OD0008', 'A0010', '07-Dec-2018', 11.5);  
INSERT INTO Ordering  
VALUES ('OD0009', 'A0011', '07-Dec-2018', 6.5);  
INSERT INTO Ordering  
VALUES ('OD0010', 'A0007', '07-Dec-2018', 17);  
INSERT INTO Ordering  
VALUES ('OD0011', 'A0005', '07-Dec-2018', 1.4);  
INSERT INTO Ordering  
VALUES ('OD0012', 'A0003', '08-Dec-2018', 2.2);  
INSERT INTO Ordering  
VALUES ('OD0013', 'A0016', '08-Dec-2018', 1.5);  
INSERT INTO Ordering  
VALUES ('OD0014', 'A0016', '08-Dec-2018', 2.0);  
INSERT INTO Ordering  
VALUES ('OD0015', 'A0002', '08-Dec-2018', 0.5);  
INSERT INTO Ordering  
VALUES ('OD0016', 'A0015', '08-Dec-2018', 3.5);  
INSERT INTO Ordering  
VALUES ('OD0017', 'A0023', '09-Dec-2018', 2.3);  
INSERT INTO Ordering  
VALUES ('OD0018', 'A0030', '09-Dec-2018', 1.2);  
INSERT INTO Ordering  
VALUES ('OD0019', 'A0028', '09-Dec-2018', 1.0);  
INSERT INTO Ordering  
VALUES ('OD0020', 'A0019', '09-Dec-2018', 0.5);  
INSERT INTO Ordering  
VALUES ('OD0021', 'A0013', '09-Dec-2018', 2.1);  
INSERT INTO Ordering  
VALUES ('OD0022', 'A0003', '09-Dec-2018', 1.7);
```

```

INSERT INTO Ordering
VALUES ('OD0023', 'A0015', '10-Dec-2018', 1.9);
INSERT INTO Ordering
VALUES ('OD0024', 'A0022', '10-Dec-2018', 5.0);
INSERT INTO Ordering
VALUES ('OD0025', 'A0011', '10-Dec-2018', 4.5);
INSERT INTO Ordering
VALUES ('OD0026', 'A0026', '10-Dec-2018', 3.0);
INSERT INTO Ordering
VALUES ('OD0027', 'A0008', '10-Dec-2018', 4.0);
INSERT INTO Ordering
VALUES ('OD0028', 'A0003', '10-Dec-2018', 2.2);
INSERT INTO Ordering
VALUES ('OD0029', 'A0001', '10-Dec-2018', 0.3);
INSERT INTO Ordering
VALUES ('OD0030', 'A0017', '10-Dec-2018', 0.3);
INSERT INTO Ordering
VALUES ('OD0031', 'A0021', '10-Dec-2018', 1.0);
INSERT INTO Ordering
VALUES ('OD0032', 'A0013', '10-Dec-2018', 0.5);
INSERT INTO Ordering
VALUES ('OD0033', 'A0005', '11-Dec-2018', 1.0);
INSERT INTO Ordering
VALUES ('OD0034', 'A0002', '11-Dec-2018', 0.3);
INSERT INTO Ordering
VALUES ('OD0035', 'A0017', '11-Dec-2018', 1.0);
INSERT INTO Ordering
VALUES ('OD0036', 'A0009', '11-Dec-2018', 6.5);
INSERT INTO Ordering
VALUES ('OD0037', 'A0014', '11-Dec-2018', 1.5);
INSERT INTO Ordering
VALUES ('OD0038', 'A0023', '11-Dec-2018', 1.9);
INSERT INTO Ordering
VALUES ('OD0039', 'A0012', '11-Dec-2018', 2.3);
INSERT INTO Ordering
VALUES ('OD0040', 'A0001', '11-Dec-2018', 11.5);
INSERT INTO Ordering
VALUES ('OD0041', 'A0012', '11-Dec-2018', 0.5);
INSERT INTO Ordering
VALUES ('OD0042', 'A0005', '12-Dec-2018', 4.0);
INSERT INTO Ordering
VALUES ('OD0043', 'A0028', '12-Dec-2018', 4.2);
INSERT INTO Ordering
VALUES ('OD0044', 'A0026', '12-Dec-2018', 0.3);
INSERT INTO Ordering
VALUES ('OD0045', 'A0002', '12-Dec-2018', 1.9);
INSERT INTO Ordering
VALUES ('OD0046', 'A0017', '12-Dec-2018', 0.9);
INSERT INTO Ordering
VALUES ('OD0047', 'A0023', '12-Dec-2018', 0.3);
INSERT INTO Ordering
VALUES ('OD0048', 'A0024', '12-Dec-2018', 5.0);
INSERT INTO Ordering
VALUES ('OD0049', 'A0013', '12-Dec-2018', 12.0);
INSERT INTO Ordering
VALUES ('OD0050', 'A0019', '12-Dec-2018', 7.7);

```

```

CREATE TABLE Payment (
    customer_id VARCHAR2(10) REFERENCES Customer (customer_id),
    staff_id VARCHAR2(10) REFERENCES Staff (staff_id),
    amount NUMBER,
    payment_date DATE,
    CONSTRAINT custom_staff_PK PRIMARY KEY (customer_id,staff_id));

```

```

INSERT INTO Payment
VALUES ('A0001','S0001',3.2,'06-Dec-2018');
INSERT INTO Payment
VALUES ('A0002','S0004',6,'06-Dec-2018');
INSERT INTO Payment
VALUES ('A0008','S0001',1.5,'06-Dec-2018');
INSERT INTO Payment
VALUES ('A0014','S0003',4.1,'06-Dec-2018');
INSERT INTO Payment
VALUES ('A0007','S0004',1.5,'06-Dec-2018');
INSERT INTO Payment
VALUES ('A0010','S0002',1.3,'06-Dec-2018');
INSERT INTO Payment
VALUES ('A0002','S0001',2,'07-Dec-2018');
INSERT INTO Payment
VALUES ('A0010','S0003',11.5,'07-Dec-2018');
INSERT INTO Payment
VALUES ('A0011','S0004',6.5,'07-Dec-2018');
INSERT INTO Payment
VALUES ('A0007','S0001',17,'07-Dec-2018');
INSERT INTO Payment
VALUES ('A0005','S0001',1.4,'07-Dec-2018');
INSERT INTO Payment
VALUES ('A0003','S0001',2.2,'08-Dec-2018');
INSERT INTO Payment
VALUES ('A0016','S0002',1.5,'08-Dec-2018');
INSERT INTO Payment
VALUES ('A0016','S0003',2,'08-Dec-2018');
INSERT INTO Payment
VALUES ('A0002','S0002',0.5,'08-Dec-2018');
INSERT INTO Payment
VALUES ('A0015','S0004',3.5,'09-Dec-2018');
INSERT INTO Payment
VALUES ('A0023','S0002',2.3,'09-Dec-2018');
INSERT INTO Payment
VALUES ('A0030','S0003',1.2,'09-Dec-2018');
INSERT INTO Payment
VALUES ('A0028','S0001',1,'09-Dec-2018');
INSERT INTO Payment
VALUES ('A0019','S0001',0.5,'09-Dec-2018');
INSERT INTO Payment
VALUES ('A0013','S0001',2.1,'09-Dec-2018');
INSERT INTO Payment
VALUES ('A0003','S0002',1.7,'09-Dec-2018');
INSERT INTO Payment
VALUES ('A0015','S0001',1.9,'10-Dec-2018');
INSERT INTO Payment
VALUES ('A0022','S0002',5,'10-Dec-2018');
INSERT INTO Payment

```

```

VALUES ('A0011', 'S0002', 4.5, '10-Dec-2018');
INSERT INTO Payment
VALUES ('A0026', 'S0001', 3, '10-Dec-2018');
INSERT INTO Payment
VALUES ('A0008', 'S0002', 4, '10-Dec-2018');
INSERT INTO Payment
VALUES ('A0003', 'S0003', 2.2, '10-Dec-2018');
INSERT INTO Payment
VALUES ('A0001', 'S0003', 0.3, '10-Dec-2018');
INSERT INTO Payment
VALUES ('A0017', 'S0001', 0.3, '10-Dec-2018');
INSERT INTO Payment
VALUES ('A0021', 'S0001', 1, '10-Dec-2018');
INSERT INTO Payment
VALUES ('A0013', 'S0002', 0.5, '10-Dec-2018');
INSERT INTO Payment
VALUES ('A0005', 'S0002', 1, '11-Dec-2018');
INSERT INTO Payment
VALUES ('A0002', 'S0003', 0.3, '11-Dec-2018');
INSERT INTO Payment
VALUES ('A0017', 'S0003', 1, '11-Dec-2018');
INSERT INTO Payment
VALUES ('A0009', 'S0001', 6.5, '11-Dec-2018');
INSERT INTO Payment
VALUES ('A0014', 'S0002', 1.5, '11-Dec-2018');
INSERT INTO Payment
VALUES ('A0023', 'S0001', 1.9, '11-Dec-2018');
INSERT INTO Payment
VALUES ('A0012', 'S0003', 2.3, '11-Dec-2018');
INSERT INTO Payment
VALUES ('A0001', 'S0002', 11.5, '11-Dec-2018');
INSERT INTO Payment
VALUES ('A0012', 'S0001', 0.5, '11-Dec-2018');
INSERT INTO Payment
VALUES ('A0005', 'S0004', 4, '12-Dec-2018');
INSERT INTO Payment
VALUES ('A0028', 'S0004', 2.4, '12-Dec-2018');
INSERT INTO Payment
VALUES ('A0026', 'S0002', 0.3, '12-Dec-2018');
INSERT INTO Payment
VALUES ('A0030', 'S0002', 1.9, '12-Dec-2018');
INSERT INTO Payment
VALUES ('A0017', 'S0002', 0.9, '12-Dec-2018');
INSERT INTO Payment
VALUES ('A0023', 'S0004', 0.3, '12-Dec-2018');
INSERT INTO Payment
VALUES ('A0024', 'S0004', 5, '12-Dec-2018');
INSERT INTO Payment
VALUES ('A0013', 'S0003', 12, '12-Dec-2018');
INSERT INTO Payment
VALUES ('A0019', 'S0004', 7.7, '12-Dec-2018');

```

```
--Query
```

```
--Change customer balance after topup
```



```

UPDATE Customer
SET balance = 20
WHERE customer_id = 'A0002';

--Delete Staff Data
DELETE FROM Staff_tel
WHERE staff_id = 'S0001';

--Displaying Data
--Showing Customer Order on particular Day
SELECT *
FROM payment
WHERE payment_date = '12-Dec-2018'
ORDER BY customer_id;

--Show Delivery Order
SELECT Order_ID, Order_Date, Price, Delivery_Date, Time, Location
FROM ORDERING JOIN DELIVERY
USING (Order_ID);

--Showing Customer Detail of the Order
SELECT *
FROM CUSTOMER
NATURAL JOIN ORDERING;

--Show delivery detail where location is 'Faculty Computing'
SELECT ORDER_ID, DELIVERY_DATE, TIME, LOCATION, CUSTOMER_NAME
FROM DELIVERY
NATURAL JOIN CUSTOMER
WHERE LOCATION = 'Faculty Computing';

--listing the customer who have balance more than 10
select customer_id, balance
from Customer
where balance>10;

--Showing Payment Detail Join with Order
select CUSTOMER_ID, STAFF_ID, AMOUNT, PAYMENT_DATE, ORDER_ID
from payment
natural join ordering
where order_date = payment_date;

--Calculate and Group Data
--Show Daily Sales
SELECT Order_Date, SUM(Price) Total_Sales
FROM Ordering
GROUP BY order_date
ORDER BY order_date;

--Creating report for staff delivery times.
select staff_id, count(staff_id) as delivery_times
from Delivery
group by staff_id;

--finding the average price, maximum price, minimum price and total price in
the Ordering relation
select round(avg(price),2), min(price), max(price), sum(price)

```

```
from Ordering;
```

```
--listing the number of order in each of the date  
select order_date, count(order_date) as number_order  
from Ordering  
group by order_date  
order by order_date asc;
```

```
--Listing total Order Made by Customer  
select Customer.customer_id, count(ordering.customer_id) as "Order Made"  
from Customer, Ordering  
where Customer.customer_id = Ordering.customer_id  
group by Customer.customer_id  
order by count(ordering.customer_id);
```

## 4.4.2 Data Manipulation Language

### 4.4.2.1 Data Entry

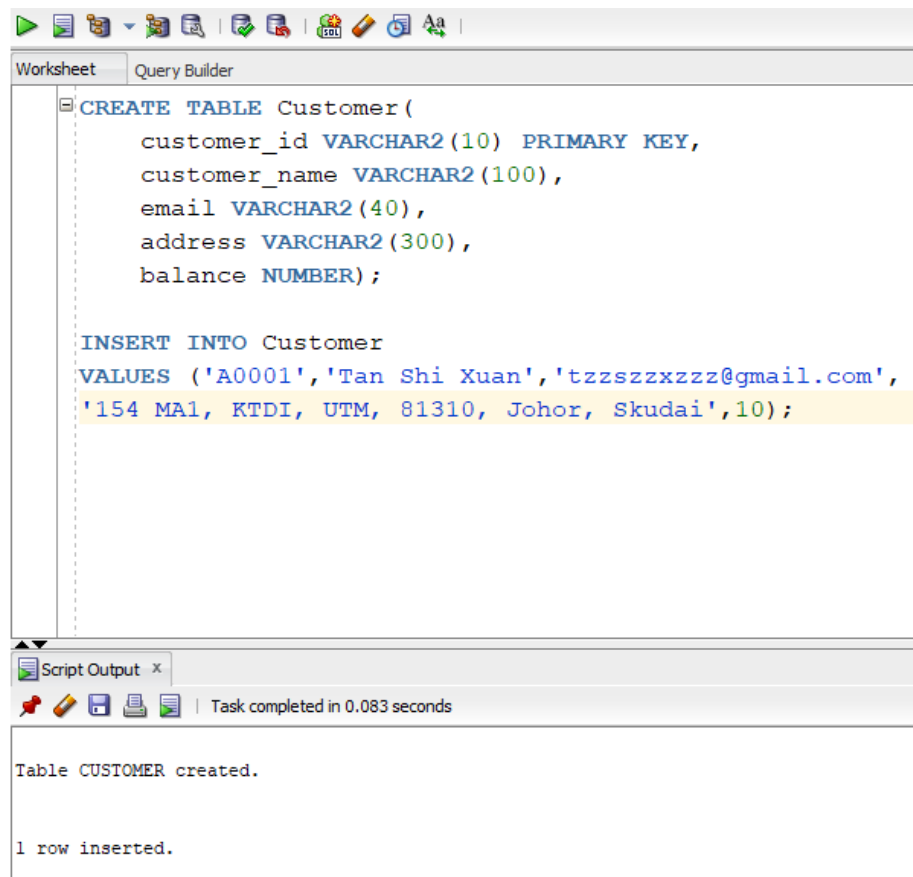
#### Customer registration

INSERT INTO Customer

(customer\_id, customer\_name, email, address, balance)

VALUES

('A0001', 'Tan Shi Xuan', '[tzzszzxzzz@gmail.com](mailto:tzzszzxzzz@gmail.com)', '154 MA1, KTDI, UTM, 81310, Johor, Skudai', '10');



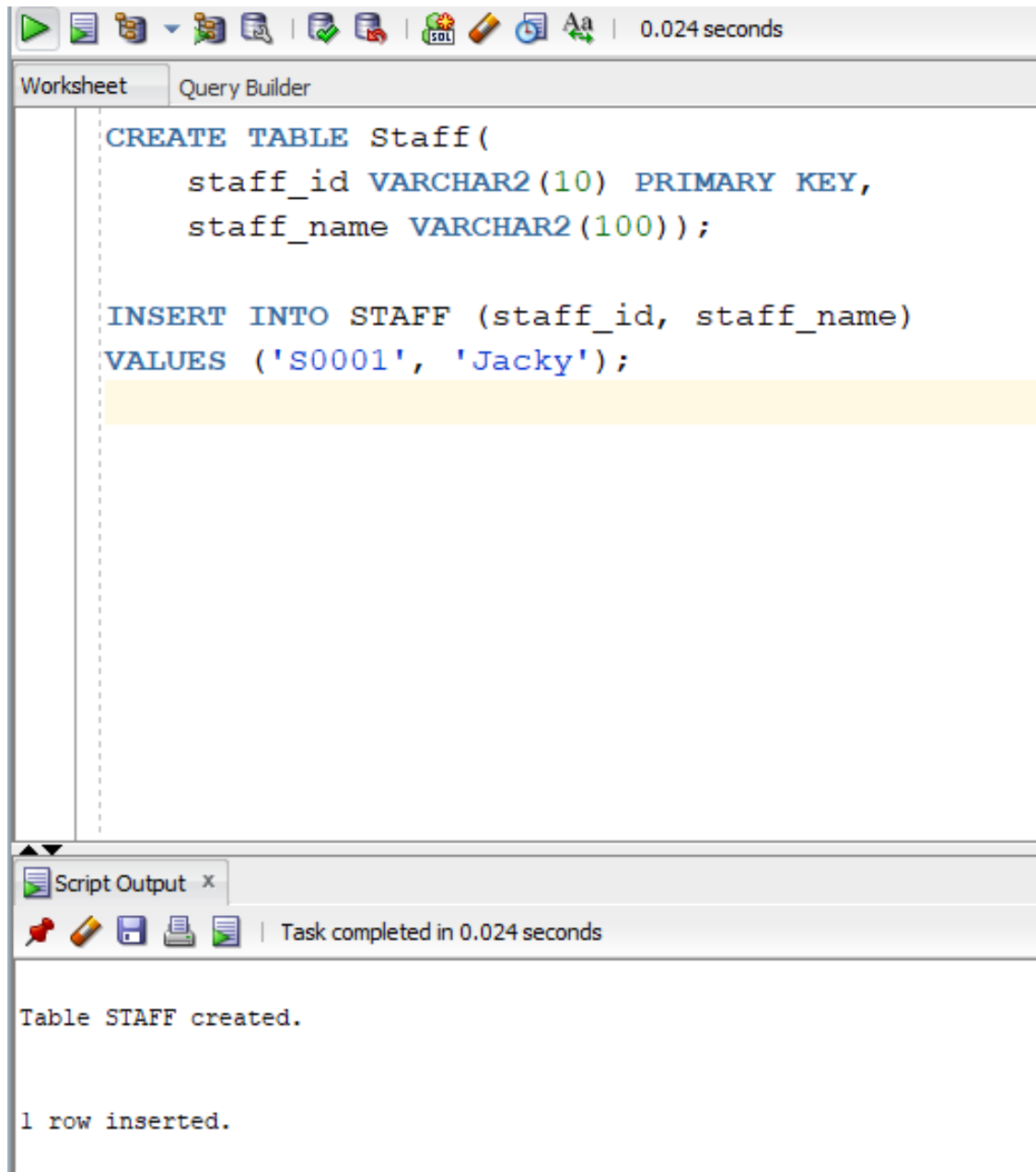
## Staff registration

INSERT INTO Staff

(staff\_id, staff\_name)

VALUES

('S0001', 'Jacky');



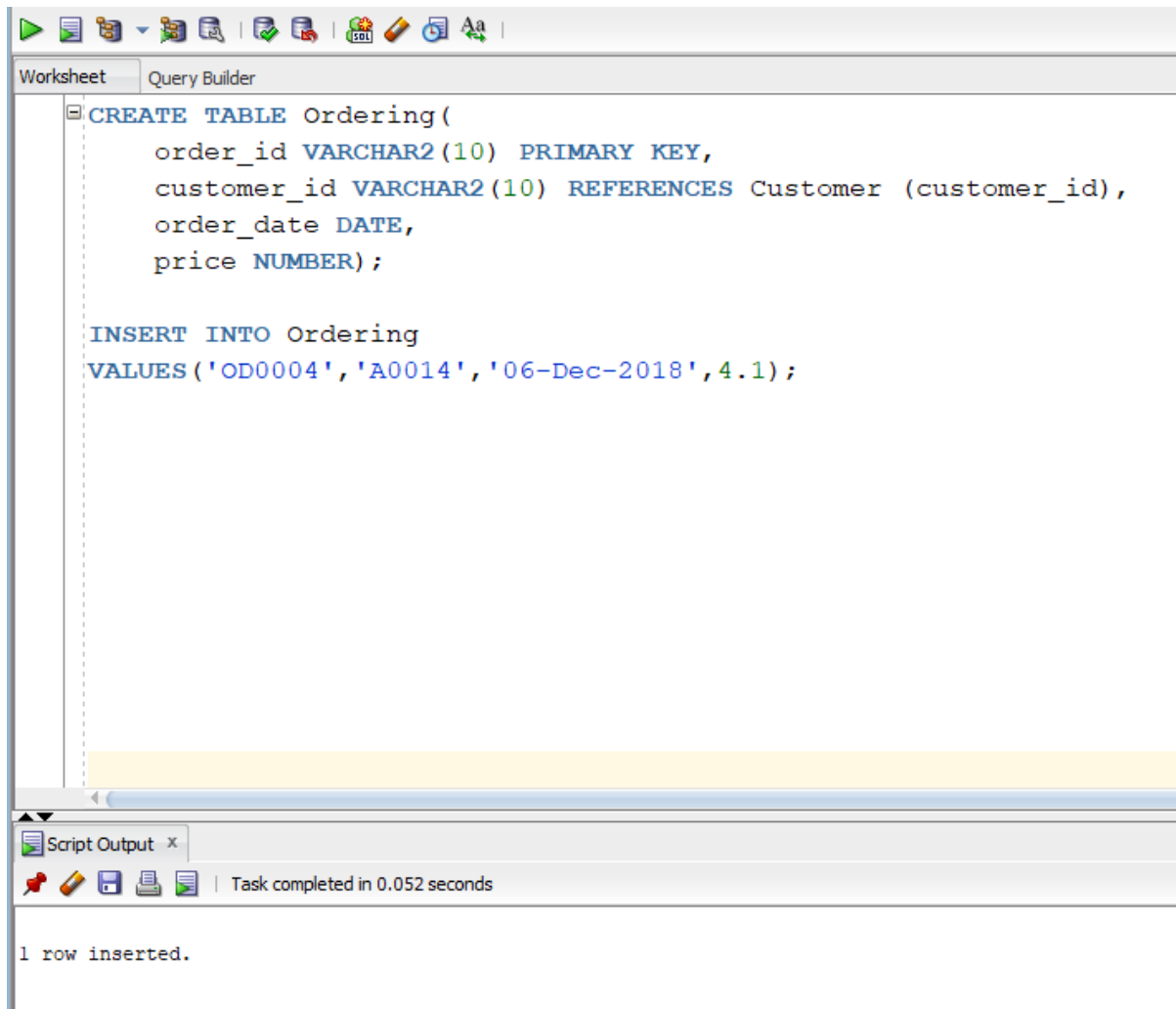
## Order Making

INSERT INTO Order

(order\_id, customer\_id, order\_date, price)

VALUES

('OD0004', 'A0014', '06-DEC-2018', 4.1);



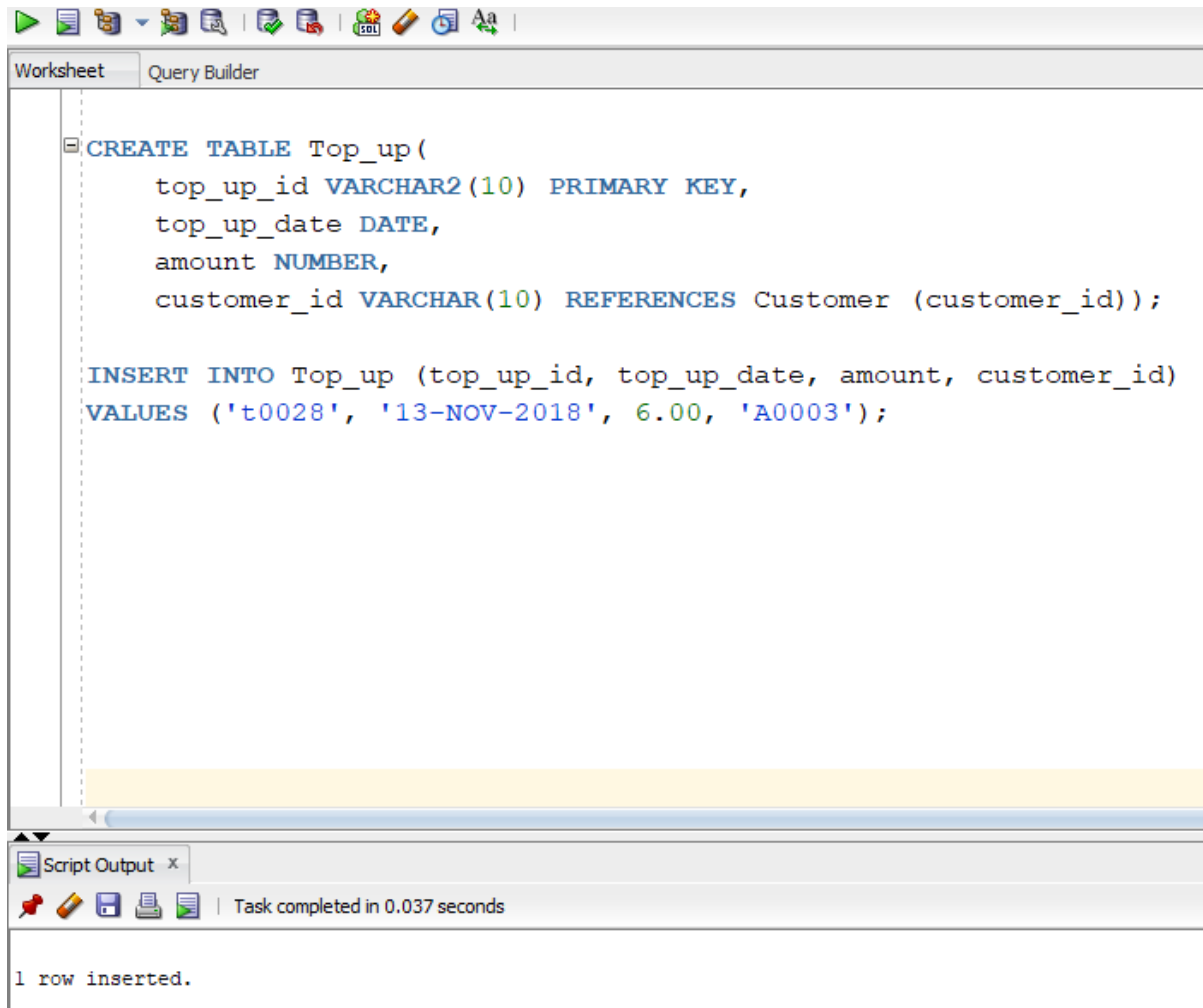
## Top-up Making

INSERT INTO Top-up

(top\_up\_id, top\_up\_date, amount, customer\_id)

VALUES

('t0028', '12-NOV-2018', '6.00', 'A0003');



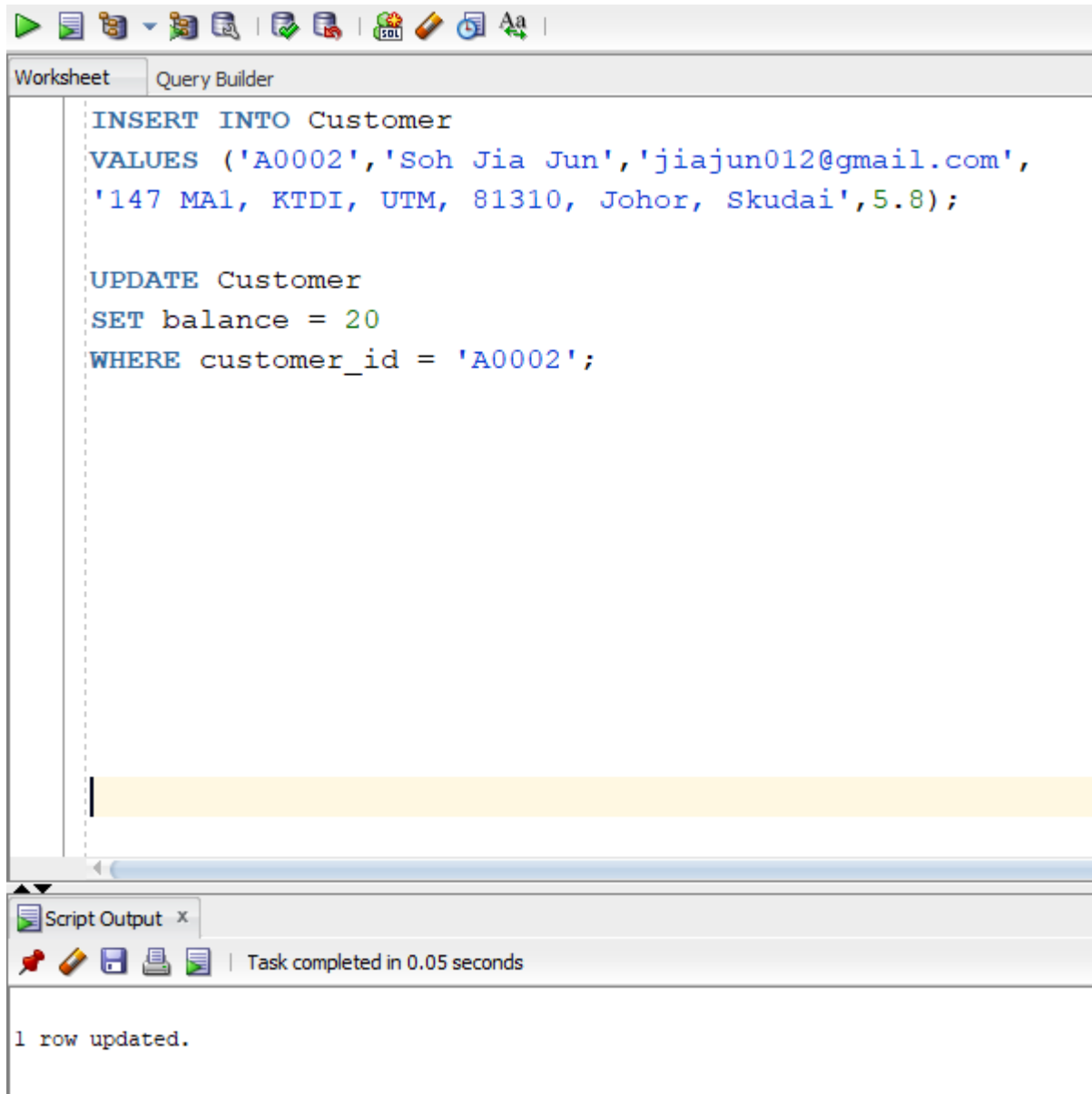
#### 4.4.2.2 Data Update / Deletion

##### Update Balance

UPDATE Customer

SET balance = 20

WHERE customer\_id = 'A0002';



The screenshot shows a SQL query editor window with a toolbar at the top containing icons for running queries, saving, and other functions. The main area is divided into two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying the following SQL code:

```
INSERT INTO Customer
VALUES ('A0002','Soh Jia Jun','jiajun012@gmail.com',
'147 MA1, KTDI, UTM, 81310, Johor, Skudai',5.8);

UPDATE Customer
SET balance = 20
WHERE customer_id = 'A0002';
```

Below the code editor, there is a yellow highlighted area. At the bottom of the window, a 'Script Output' panel shows the execution results:

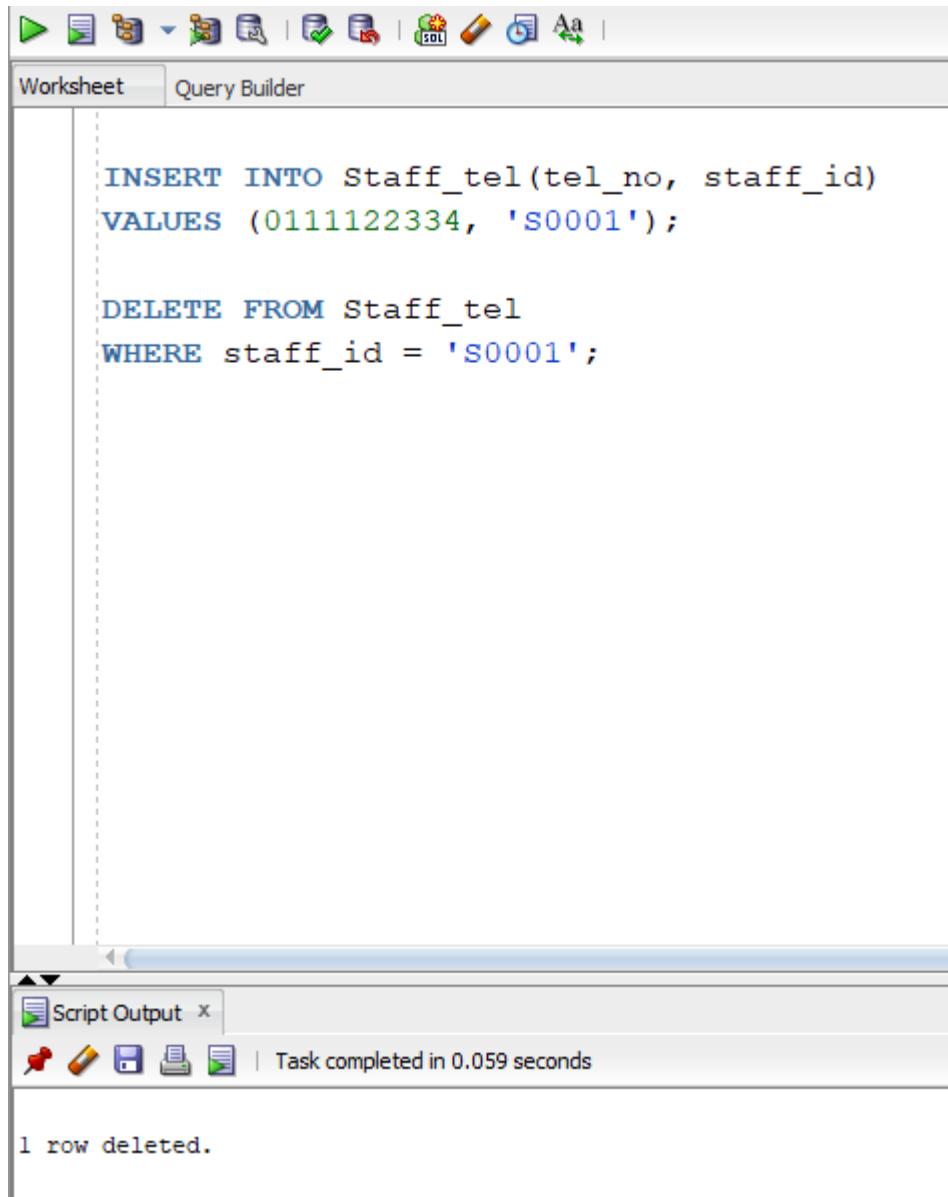
```
Task completed in 0.05 seconds

1 row updated.
```

## Delete Staff information

DELETE FROM Staff\_tel

WHERE staff\_id = 'S0001';





### 4.4.2.3 Displaying Data

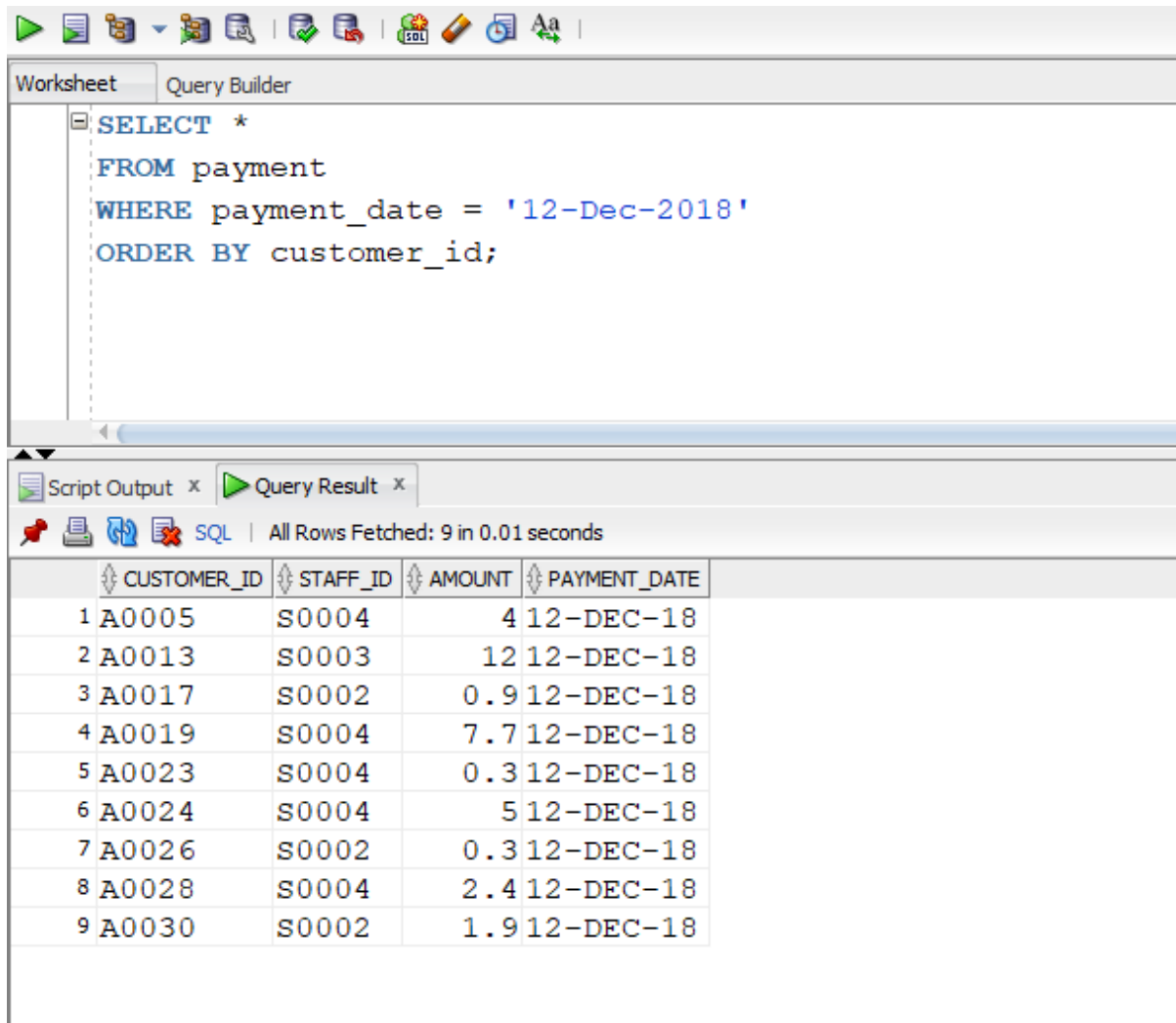
#### Showing customer\_order on particular day

SELECT \*

FROM payment

WHERE payment\_date = '12-Dec-2018'

ORDER BY customer\_id;



The screenshot displays a SQL query editor window with a toolbar at the top. The 'Query Builder' tab is active, showing the following SQL query:

```
SELECT *  
FROM payment  
WHERE payment_date = '12-Dec-2018'  
ORDER BY customer_id;
```

Below the query editor, the 'Query Result' tab is active, showing the results of the query. The status bar indicates 'All Rows Fetched: 9 in 0.01 seconds'. The results are displayed in a table with the following columns: CUSTOMER\_ID, STAFF\_ID, AMOUNT, and PAYMENT\_DATE.

	CUSTOMER_ID	STAFF_ID	AMOUNT	PAYMENT_DATE
1	A0005	S0004	4	12-DEC-18
2	A0013	S0003	12	12-DEC-18
3	A0017	S0002	0.9	12-DEC-18
4	A0019	S0004	7.7	12-DEC-18
5	A0023	S0004	0.3	12-DEC-18
6	A0024	S0004	5	12-DEC-18
7	A0026	S0002	0.3	12-DEC-18
8	A0028	S0004	2.4	12-DEC-18
9	A0030	S0002	1.9	12-DEC-18

### Show delivering Order

SELECT Order\_ID, Order\_Date, Price, Delivery\_Date, Time, Location

FROM ORDERING JOIN DELIVERY

USING (Order\_ID);

```
SELECT Order_ID, Order_Date, Price, Delivery_Date, Time, Location
FROM ORDERING JOIN DELIVERY
USING (Order_ID);
```

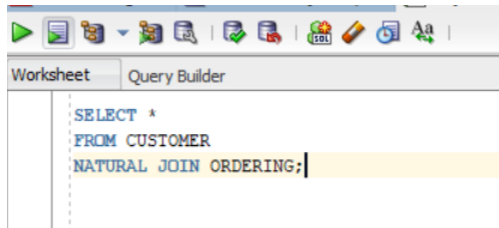
	ORDER_ID	ORDER_DATE	PRICE	DELIVERY_DATE	TIME	LOCATION
1	OD0001	06-DEC-18	3.2	06-DEC-18	1200	Faculty Computing
2	OD0003	06-DEC-18	1.5	06-DEC-18	1700	Faculty Computing
3	OD0005	06-DEC-18	1.5	06-DEC-18	1630	Arked Angkasa
4	OD0007	07-DEC-18	2	07-DEC-18	1200	Arked Meranti
5	OD0008	07-DEC-18	11.5	07-DEC-18	1530	Arked Angkasa
6	OD0009	07-DEC-18	6.5	07-DEC-18	1400	Faculty Computing
7	OD0010	07-DEC-18	17	07-DEC-18	1500	Arked Meranti
8	OD0011	07-DEC-18	1.4	07-DEC-18	1130	Arked Angkasa
9	OD0012	08-DEC-18	2.2	08-DEC-18	1800	Faculty Computing
10	OD0013	08-DEC-18	1.5	08-DEC-18	1740	Faculty Computing
11	OD0023	10-DEC-18	1.9	10-DEC-18	2000	MA1, KTDI
12	OD0024	10-DEC-18	5	10-DEC-18	2000	MA1, KTDI
13	OD0027	10-DEC-18	4	10-DEC-18	1720	Mak Ngah, KTDI, UTM
14	OD0028	10-DEC-18	2.2	10-DEC-18	1630	M27, KTDI, UTM
15	OD0036	11-DEC-18	6.5	11-DEC-18	1900	G18, KRP, UTM
16	OD0042	12-DEC-18	4	12-DEC-18	1330	H04, KTF, UTM
17	OD0043	12-DEC-18	4.2	12-DEC-18	1400	K11, KRP, UTM
18	OD0049	12-DEC-18	12	12-DEC-18	1230	FABU
19	OD0050	12-DEC-18	7.7	12-DEC-18	1200	FABU

## Showing Customer Detail of the Order

SELECT \*

FROM CUSTOMER

NATURAL JOIN ORDERING;



Script Output x Query Result x

SQL | All Rows Fetched: 50 in 0.173 seconds

CUSTOMER_ID	CUSTOMER_NAME	EMAIL	ADDRESS	BALANCE	ORDER_ID	ORDER_DATE	PRICE
1 A0001	Tan Shi Xuan	tzszszszsz@gmail.com	154 MA1, KTDI, UTM, 81310, Johor, Skudai	10	OD0001	06-DEC-18	3.2
2 A0002	Soh Jia Jun	jiajun012@gmail.com	147 MA1, KTDI, UTM, 81310, Johor, Skudai	5.8	OD0002	06-DEC-18	6
3 A0008	Ibrahim Mohamed	ibrahra@hotmail.com	108 MA4, KTDI, UTM, 81310, Johor, Skudai	5.2	OD0003	06-DEC-18	1.5
4 A0014	Mohammad Gala	GalaGaga@gmail.com	100 M17, KTDI, UTM, 81310, Johor, Skudai	7.8	OD0004	06-DEC-18	4.1
5 A0007	Mellisa Lim	mellisallll@gmail.com	211 MA4, KTDI, UTM, 81310, Johor, Skudai	7	OD0005	06-DEC-18	1.5
6 A0010	Siti Nuraliyah Zaina	aliyah9797@gmail.com	233 M27, KTDI, UTM, 81310, Johor, Skudai	7	OD0006	06-DEC-18	1.3
7 A0002	Soh Jia Jun	jiajun012@gmail.com	147 MA1, KTDI, UTM, 81310, Johor, Skudai	5.8	OD0007	07-DEC-18	2
8 A0010	Siti Nuraliyah Zaina	aliyah9797@gmail.com	233 M27, KTDI, UTM, 81310, Johor, Skudai	7	OD0008	07-DEC-18	11.5
9 A0011	Mohamed Qatadah	QQQQ@hotmail.com	111, M15, KTDI, UTM, 81310, Johor, Skudai	7	OD0009	07-DEC-18	6.5
10 A0007	Mellisa Lim	mellisallll@gmail.com	211 MA4, KTDI, UTM, 81310, Johor, Skudai	7	OD0010	07-DEC-18	17
11 A0005	Goh Chiang Cheng	kentkk@gmail.com	250 MA1, KTDI, UTM, 81310, Johor, Skudai	1.2	OD0011	07-DEC-18	1.4
12 A0003	Tan See Jou	seejou666@gmail.com	412 MA7, KTDI, UTM, 81310, Johor, Skudai	3.3	OD0012	08-DEC-18	2.2
13 A0016	Evelyn Lim	EvveeeL@gmail.com	G01, M11, KTDI, UTM, 81310, Johor, Skudai	0	OD0013	08-DEC-18	1.5
14 A0016	Evelyn Lim	EvveeeL@gmail.com	G01, M11, KTDI, UTM, 81310, Johor, Skudai	0	OD0014	08-DEC-18	2
15 A0002	Soh Jia Jun	jiajun012@gmail.com	147 MA1, KTDI, UTM, 81310, Johor, Skudai	5.8	OD0015	08-DEC-18	0.5
16 A0015	Lin Jun Jie	JJLin@gmail.com	108, M16, KTDI, UTM, 81310, Johor, Skudai	8	OD0016	08-DEC-18	3.5
17 A0023	Puan Naliyah	naliyah1002@gmail.com	G01 MA4 KTDI,UTM, 81310, Johor, Skudai	1.6	OD0017	09-DEC-18	2.3
18 A0030	Charlie Puth	charlie91@hotmail.com	450 MA1, KTDI, UTM, 81310, Johor, Skudai	11.1	OD0018	09-DEC-18	1.2
19 A0028	Liew Chia Jia	lcj09@gmail.com	229, MA7, KTDI, UTM, 81310, Johor, Skudai	5.1	OD0019	09-DEC-18	1
20 A0019	Lee Joshua	joshualee@gmail.com	269 M15. KTDI, UTM, 81310, Johor, Skudai	50	OD0020	09-DEC-18	0.5
21 A0013	Leo Tan	Leo7496@gmail.com	204 M15. KTDI, UTM, 81310, Johor, Skudai	4.6	OD0021	09-DEC-18	2.1
22 A0003	Tan See Jou	seejou666@gmail.com	412 MA7, KTDI, UTM, 81310, Johor, Skudai	3.3	OD0022	09-DEC-18	1.7

23 A0015	Lin Jun Jie	JJLin@gmail.com	108, M16, KTDI, UTM, 81310, Johor, Skudai	8	OD0023	10-DEC-18	1.9
24 A0022	Siti Azalina	12Siti@gmail.com	G11 MA7,KTDI, UTM, 81310, Johor, Skudai	1.2	OD0024	10-DEC-18	5
25 A0011	Mohamed Qatadah	QQQQ@hotmail.com	111, M15, KTDI, UTM, 81310, Johor, Skudai	7	OD0025	10-DEC-18	4.5
26 A0026	Ahmad Fitri	afitri8@gmail.com	122 MA1, KTDI, UTM, 81310, Johor, Skudai	2	OD0026	10-DEC-18	3
27 A0008	Ibrahim Mohamed	ibrahra@hotmail.com	108 MA4, KTDI, UTM, 81310, Johor, Skudai	5.2	OD0027	10-DEC-18	4
28 A0003	Tan See Jou	seejou666@gmail.com	412 MA7, KTDI, UTM, 81310, Johor, Skudai	3.3	OD0028	10-DEC-18	2.2
29 A0001	Tan Shi Xuan	tzszszszsz@gmail.com	154 MA1, KTDI, UTM, 81310, Johor, Skudai	10	OD0029	10-DEC-18	0.3
30 A0017	Lim Hooi Hooi	Lhh@hotmail.com	112, M11, KTDI, UTM, 81310, Johor, Skudai	0	OD0030	10-DEC-18	0.3
31 A0021	Ong Thien Ming	MinTea22@gmail.com	165 M16, KTDI, UTM, 81310, Johor, Skudai	10	OD0031	10-DEC-18	1
32 A0013	Leo Tan	Leo7496@gmail.com	204 M15. KTDI, UTM, 81310, Johor, Skudai	4.6	OD0032	10-DEC-18	0.5
33 A0005	Goh Chiang Cheng	kentkk@gmail.com	250 MA1, KTDI, UTM, 81310, Johor, Skudai	1.2	OD0033	11-DEC-18	1
34 A0002	Soh Jia Jun	jiajun012@gmail.com	147 MA1, KTDI, UTM, 81310, Johor, Skudai	5.8	OD0034	11-DEC-18	0.3
35 A0017	Lim Hooi Hooi	Lhh@hotmail.com	112, M11, KTDI, UTM, 81310, Johor, Skudai	0	OD0035	11-DEC-18	1
36 A0009	Ali Razak Zul Fikri	alirazak@yahoo.com	222 M16, KTDI, UTM, 81310, Johor, Skudai	8.9	OD0036	11-DEC-18	6.5
37 A0014	Mohammad Gala	GalaGaga@gmail.com	100 M17, KTDI, UTM, 81310, Johor, Skudai	7.8	OD0037	11-DEC-18	1.5
38 A0023	Puan Naliyah	naliyah1002@gmail.com	G01 MA4 KTDI,UTM, 81310, Johor, Skudai	1.6	OD0038	11-DEC-18	1.9
39 A0012	Ahmad Fadil	Fadil@yahoo.com	402 M17, KTDI, UTM, 81310, Johor, Skudai	4.3	OD0039	11-DEC-18	2.3
40 A0001	Tan Shi Xuan	tzszszszsz@gmail.com	154 MA1, KTDI, UTM, 81310, Johor, Skudai	10	OD0040	11-DEC-18	11.5
41 A0012	Ahmad Fadil	Fadil@yahoo.com	402 M17, KTDI, UTM, 81310, Johor, Skudai	4.3	OD0041	11-DEC-18	0.5
42 A0005	Goh Chiang Cheng	kentkk@gmail.com	250 MA1, KTDI, UTM, 81310, Johor, Skudai	1.2	OD0042	12-DEC-18	4
43 A0028	Liew Chia Jia	lcj09@gmail.com	229, MA7, KTDI, UTM, 81310, Johor, Skudai	5.1	OD0043	12-DEC-18	4.2

44 A0026	Ahmad Fitri	afitri8@gmail.com	122 MA1, KTDI, UTM, 81310, Johor, Skudai	2 OD0044	12-DEC-18	0.3
45 A0002	Soh Jia Jun	jiajun012@gmail.com	147 MA1, KTDI, UTM, 81310, Johor, Skudai	5.8 OD0045	12-DEC-18	1.9
46 A0017	Lim Hooi Hooi	Lhh@hotmail.com	112, M11, KTDI, UTM, 81310, Johor, Skudai	0 OD0046	12-DEC-18	0.9
47 A0023	Puan Naliyah	naliyah1002@gmail.com	G01 MA4 KTDI,UTM, 81310, Johor, Skudai	1.6 OD0047	12-DEC-18	0.3
48 A0024	Nur Zaniah	zaniah@gmail.com	209 MA6, KTDI, UTM, 81310, Johor, Skudai	7.8 OD0048	12-DEC-18	5
49 A0013	Leo Tan	Leo7496@gmail.com	204 M15. KTDI, UTM, 81310, Johor, Skudai	4.6 OD0049	12-DEC-18	12
50 A0019	Lee Joshua	joshualee@gmail.com	269 M15. KTDI, UTM, 81310, Johor, Skudai	50 OD0050	12-DEC-18	7.7

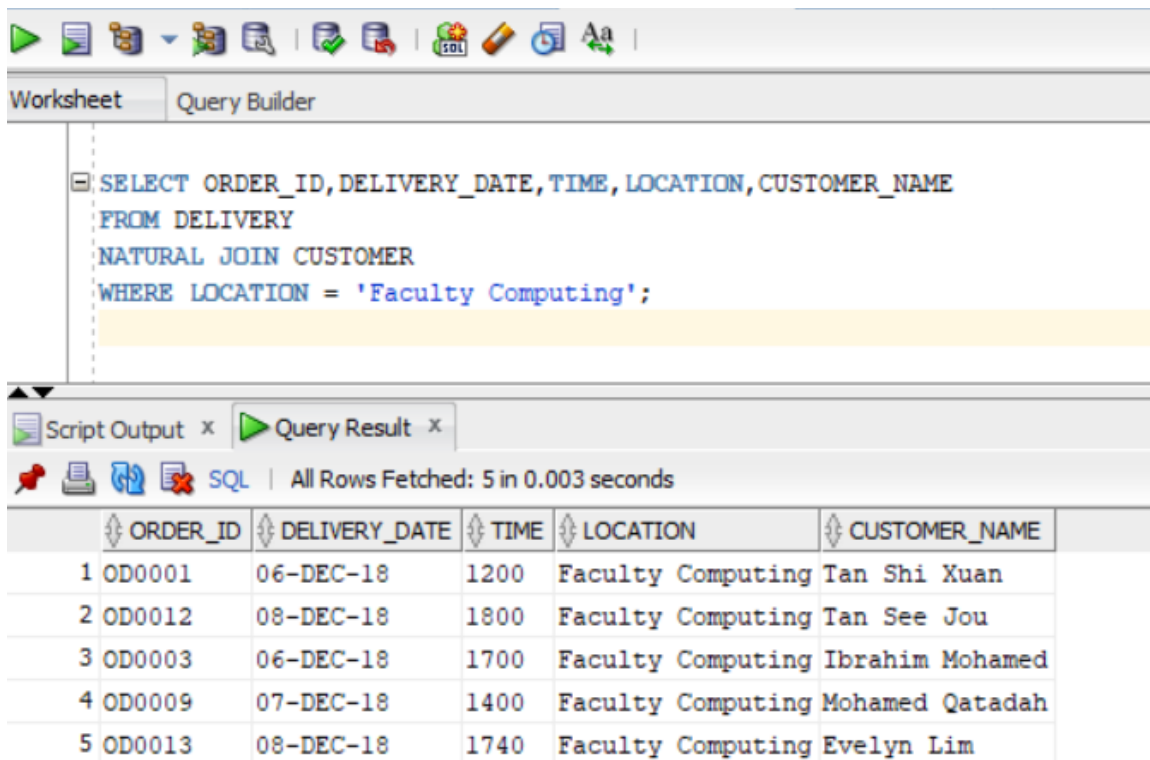
### Show delivery detail where location is 'Faculty Computing'

SELECT ORDER\_ID, DELIVERY\_DATE, TIME, LOCATION, CUSTOMER\_NAME

FROM DELIVERY

NATURAL JOIN CUSTOMER

WHERE LOCATION = 'Faculty Computing';



The screenshot shows a database query tool interface. The top toolbar includes icons for running queries, saving, and other functions. Below the toolbar, there are two tabs: 'Worksheet' and 'Query Builder'. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT ORDER_ID, DELIVERY_DATE, TIME, LOCATION, CUSTOMER_NAME
FROM DELIVERY
NATURAL JOIN CUSTOMER
WHERE LOCATION = 'Faculty Computing';
```

Below the query editor, there is a 'Script Output' and 'Query Result' section. The 'Query Result' section shows the results of the query, indicating that all rows were fetched in 0.003 seconds. The results are displayed in a table with the following columns: ORDER\_ID, DELIVERY\_DATE, TIME, LOCATION, and CUSTOMER\_NAME.

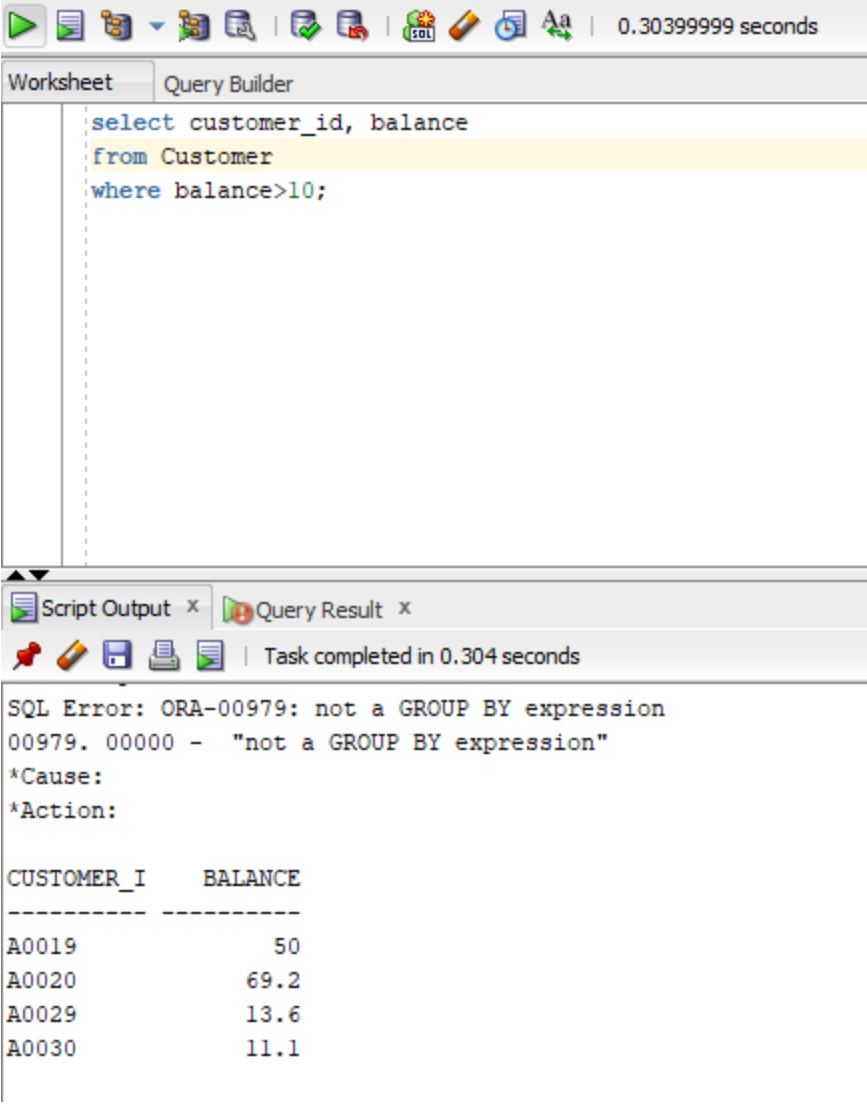
	ORDER_ID	DELIVERY_DATE	TIME	LOCATION	CUSTOMER_NAME
1	OD0001	06-DEC-18	1200	Faculty Computing	Tan Shi Xuan
2	OD0012	08-DEC-18	1800	Faculty Computing	Tan See Jou
3	OD0003	06-DEC-18	1700	Faculty Computing	Ibrahim Mohamed
4	OD0009	07-DEC-18	1400	Faculty Computing	Mohamed Qatadah
5	OD0013	08-DEC-18	1740	Faculty Computing	Evelyn Lim

### Listing the customer who have balance more than 10

SELECT customer\_id, balance

FROM Customer

WHERE balance>10;



The screenshot shows a SQL query builder window with a toolbar at the top containing icons for execution, saving, and other functions. The main area is divided into a 'Worksheet' tab and a 'Query Builder' tab. The 'Query Builder' tab contains the following SQL query:

```
select customer_id, balance
from Customer
where balance>10;
```

Below the query editor, there is a 'Script Output' tab and a 'Query Result' tab. The 'Query Result' tab is active and displays the following error message:

```
SQL Error: ORA-00979: not a GROUP BY expression
00979. 00000 - "not a GROUP BY expression"
*Cause:
*Action:
```

Below the error message, the query results are displayed in a table format:

CUSTOMER_I	BALANCE
A0019	50
A0020	69.2
A0029	13.6
A0030	11.1

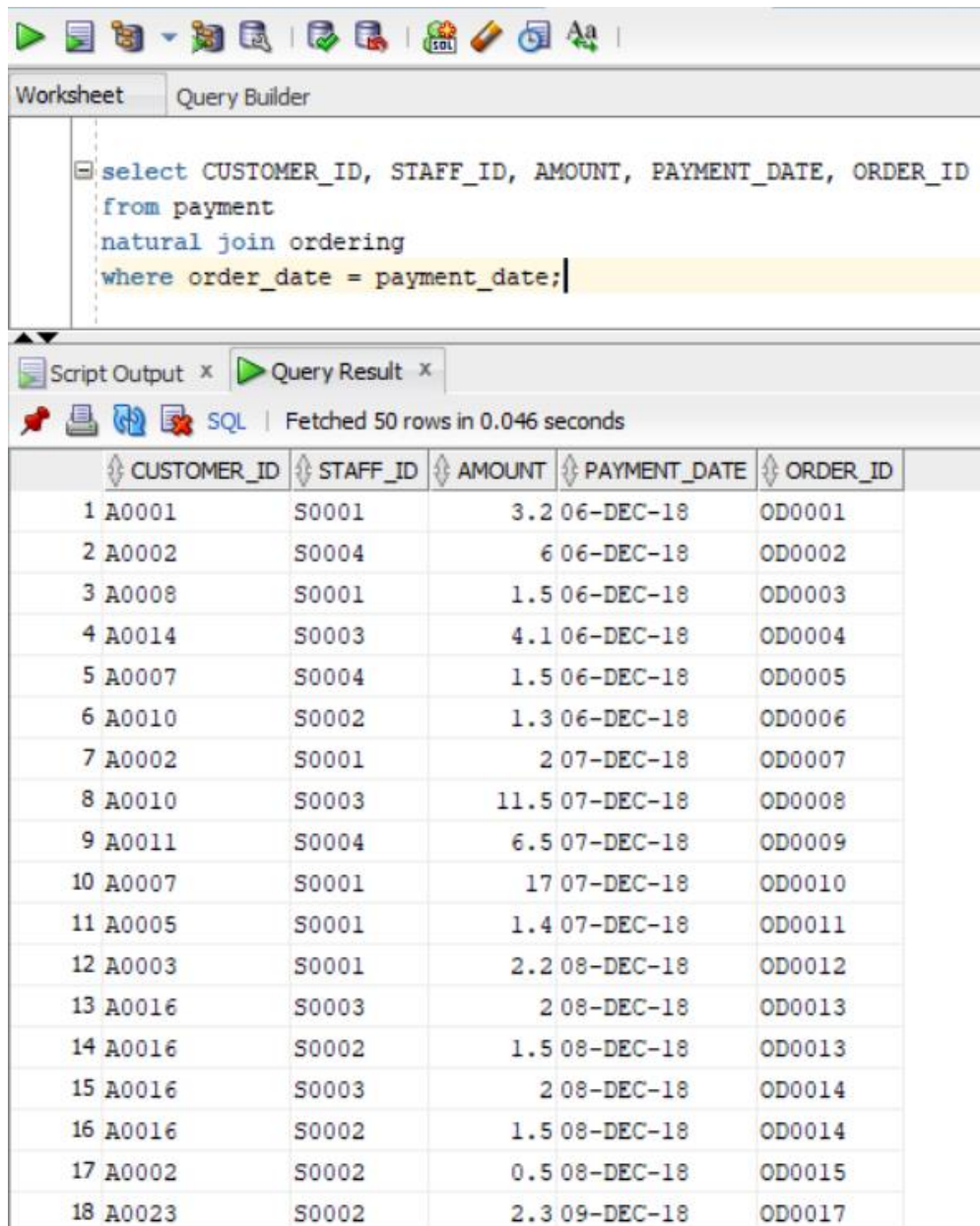
### Showing Payment Detail Join with Order

SELECT CUSTOMER\_ID, STAFF\_ID, AMOUNT, PAYMENT\_DATE, ORDER\_ID

FROM payment

NATURAL JOIN ordering

WHERE order\_date = payment\_date;



The screenshot shows a database query tool interface. At the top is a toolbar with various icons. Below it is a tabbed interface with 'Worksheet' and 'Query Builder'. The 'Query Builder' tab is active, displaying a SQL query in a text area. Below the query area is a 'Script Output' tab, which is also active, showing the query results. The results are displayed in a table with 5 columns: CUSTOMER\_ID, STAFF\_ID, AMOUNT, PAYMENT\_DATE, and ORDER\_ID. The table contains 18 rows of data, numbered 1 through 18. The 'AMOUNT' column is formatted with a comma as a thousands separator. The 'PAYMENT\_DATE' column shows dates in YYYY-MM-DD format. The 'ORDER\_ID' column shows unique identifiers for each order.

	CUSTOMER_ID	STAFF_ID	AMOUNT	PAYMENT_DATE	ORDER_ID
1	A0001	S0001	3.2	06-DEC-18	OD0001
2	A0002	S0004	6	06-DEC-18	OD0002
3	A0008	S0001	1.5	06-DEC-18	OD0003
4	A0014	S0003	4.1	06-DEC-18	OD0004
5	A0007	S0004	1.5	06-DEC-18	OD0005
6	A0010	S0002	1.3	06-DEC-18	OD0006
7	A0002	S0001	2	07-DEC-18	OD0007
8	A0010	S0003	11.5	07-DEC-18	OD0008
9	A0011	S0004	6.5	07-DEC-18	OD0009
10	A0007	S0001	17	07-DEC-18	OD0010
11	A0005	S0001	1.4	07-DEC-18	OD0011
12	A0003	S0001	2.2	08-DEC-18	OD0012
13	A0016	S0003	2	08-DEC-18	OD0013
14	A0016	S0002	1.5	08-DEC-18	OD0013
15	A0016	S0003	2	08-DEC-18	OD0014
16	A0016	S0002	1.5	08-DEC-18	OD0014
17	A0002	S0002	0.5	08-DEC-18	OD0015
18	A0023	S0002	2.3	09-DEC-18	OD0017



#### 4.4.2.4 Calculate Data

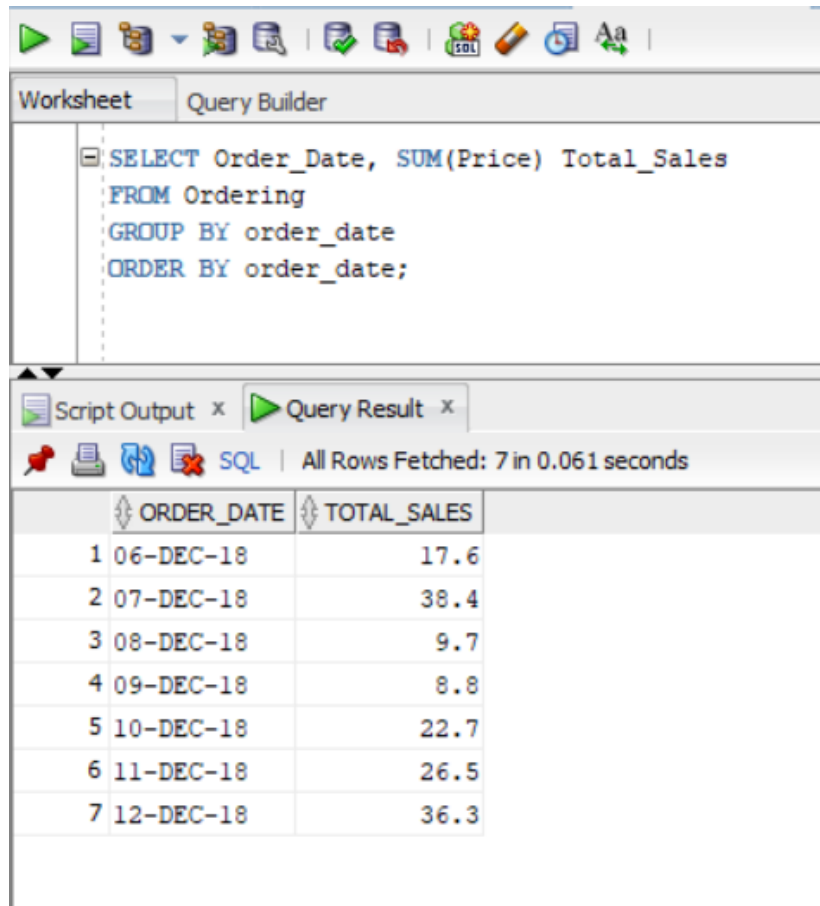
##### Show Daily Sales

SELECT Order\_Date, SUM(Price) Total\_Sales

FROM Ordering

GROUP BY order\_date

ORDER BY order\_date;



The screenshot shows a SQL query builder window with a toolbar at the top. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT Order_Date, SUM(Price) Total_Sales
FROM Ordering
GROUP BY order_date
ORDER BY order_date;
```

Below the query editor, the 'Query Result' tab is active, showing the results of the query. The status bar indicates 'All Rows Fetched: 7 in 0.061 seconds'. The results are displayed in a table with two columns: 'ORDER\_DATE' and 'TOTAL\_SALES'.

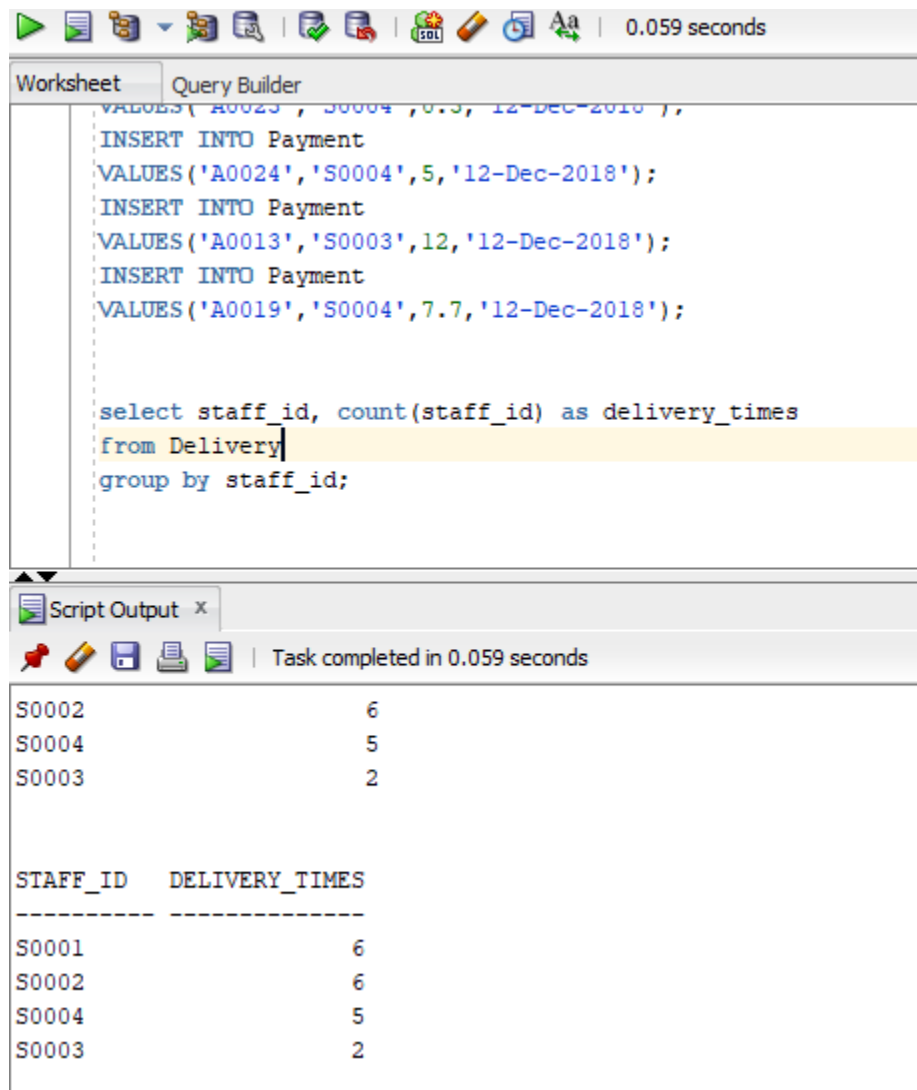
	ORDER_DATE	TOTAL_SALES
1	06-DEC-18	17.6
2	07-DEC-18	38.4
3	08-DEC-18	9.7
4	09-DEC-18	8.8
5	10-DEC-18	22.7
6	11-DEC-18	26.5
7	12-DEC-18	36.3

### Creating report for staff delivery times.

SELECT staff\_id, count(staff\_id) AS delivery\_times

FROM Delivery

GROUP BY staff\_id;



The screenshot shows a database query tool interface. The top toolbar includes icons for running queries, saving, and other functions, with a timer showing 0.059 seconds. The main window is titled 'Query Builder' and contains the following SQL code:

```
VALUES('A0023', 'S0004', 0.3, '12-Dec-2018'),  
INSERT INTO Payment  
VALUES('A0024', 'S0004', 5, '12-Dec-2018');  
INSERT INTO Payment  
VALUES('A0013', 'S0003', 12, '12-Dec-2018');  
INSERT INTO Payment  
VALUES('A0019', 'S0004', 7.7, '12-Dec-2018');  
  
select staff_id, count(staff_id) as delivery_times  
from Delivery  
group by staff_id;
```

Below the query editor, the 'Script Output' window shows the results of the query. It displays a table with two columns: STAFF\_ID and DELIVERY\_TIMES. The data is as follows:

STAFF_ID	DELIVERY_TIMES
S0002	6
S0004	5
S0003	2



**Finding the average price, maximum price, minimum price and total price in the Ordering relation**

SELECT ROUND (AVG (price),2), MIN (price), MAX (price), SUM (price)

FROM Ordering;

The screenshot displays a database query tool interface. At the top, a toolbar contains various icons for file operations, execution, and formatting, along with a timer showing '0.046 seconds'. Below the toolbar, the 'Query Builder' tab is active, showing the following SQL query:

```
select round(avg(price),2), min(price), max(price), sum(price)
from Ordering;
```

Below the query editor, the 'Script Output' and 'Query Result' tabs are visible. The 'Query Result' tab shows the output of the query, which is a single row of data. The output is displayed in a table format with the following columns and values:

ROUND (AVG (PRICE) , 2)	MIN (PRICE)	MAX (PRICE)	SUM (PRICE)
3.2	.3	17	160

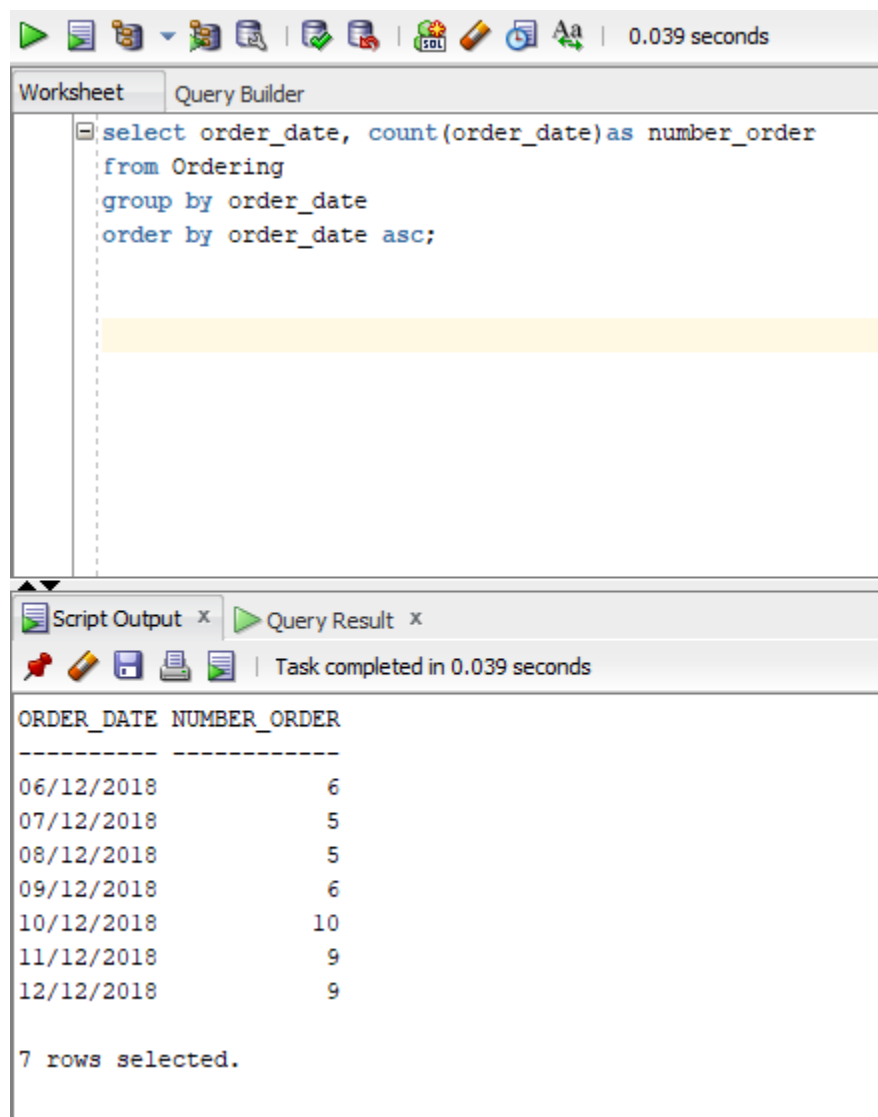
### Listing the number of order in each of the date

SELECT order\_date, count(order\_date) AS number\_order

FROM Ordering

GROUP BY order\_date

ORDER BY order\_date asc;



The screenshot shows a database query tool interface. The top toolbar includes icons for running queries, saving, and other functions, with a timer showing 0.039 seconds. The main window is titled 'Query Builder' and contains the following SQL query:

```
select order_date, count(order_date) as number_order
from Ordering
group by order_date
order by order_date asc;
```

Below the query editor, the 'Query Result' tab is active, displaying the results of the query. The results are shown in a table with two columns: 'ORDER\_DATE' and 'NUMBER\_ORDER'. The data is as follows:

ORDER_DATE	NUMBER_ORDER
06/12/2018	6
07/12/2018	5
08/12/2018	5
09/12/2018	6
10/12/2018	10
11/12/2018	9
12/12/2018	9

At the bottom of the results, it states '7 rows selected.'

## Displaying the amount of order make by the customer

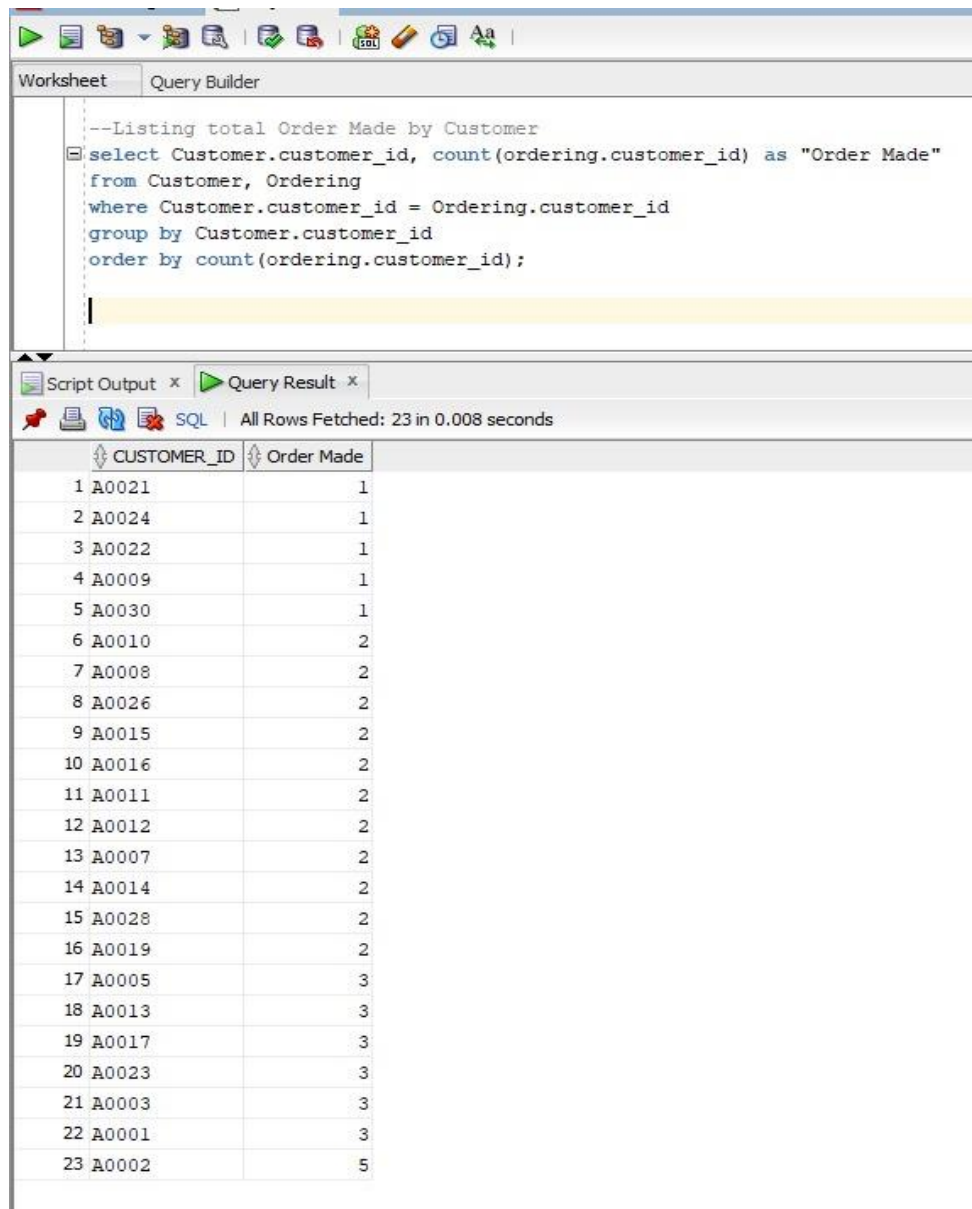
SELECT Customer.customer\_id, count(ordering.customer\_id) AS "Order Made"

FROM Customer, Ordering

WHERE Customer.customer\_id = Ordering.customer\_id

GROUP BY Customer.customer\_id

ORDER BY count(ordering.customer\_id);



The screenshot shows a database query tool interface. The top section is the 'Query Builder' tab, which contains the following SQL query:

```
--Listing total Order Made by Customer
select Customer.customer_id, count(ordering.customer_id) as "Order Made"
from Customer, Ordering
where Customer.customer_id = Ordering.customer_id
group by Customer.customer_id
order by count(ordering.customer_id);
```

Below the query editor is the 'Query Result' tab, which displays the results of the query. The results are shown in a table with two columns: 'CUSTOMER\_ID' and 'Order Made'. The table contains 23 rows of data, sorted by the 'Order Made' column in ascending order.

CUSTOMER_ID	Order Made
1 A0021	1
2 A0024	1
3 A0022	1
4 A0009	1
5 A0030	1
6 A0010	2
7 A0008	2
8 A0026	2
9 A0015	2
10 A0016	2
11 A0011	2
12 A0012	2
13 A0007	2
14 A0014	2
15 A0028	2
16 A0019	2
17 A0005	3
18 A0013	3
19 A0017	3
20 A0023	3
21 A0003	3
22 A0001	3
23 A0002	5