SCHOOL OF SCIENCE AND ENGINEERING

COURSEWORK FOR

BSC (HONS) IN COMPUTER SCIENCE

BSC (HONS) INFORMATION SYSTEMS

BSC (HONS) INFORMATION TECHNOLOGY

BACHELOR OF SOFTWARE ENGINEERING (HONS)

BSC (HONS) INFORMATION SYSTEMS (BUSINESS ANALYTICS)

BIS (HONS) IN MOBILE COMPUTING WITH ENTREPRENEURSHIP

BSC (HONS) INFORMATION TECHNOLOGY (COMPUTER NETWORKING AND SECURITY)

YEAR 1 ACADEMIC SESSION APRIL 2021

SEG1201: DATABASE FUNDAMENTALS

GROUP ASSIGNMENT

Lecturer: Dr Ling Mee Hong

DATE OF ASSIGNMENT: 17TH May 2021

DATE OF SUBMISSION: 8TH July 2021

GROUP NO. 8

Group Members	Student ID
Alicia Chong Tsui Ying	20074290
Wong Yi Qing	19028570
Kusselin Yuthasax A/P Prak Chuap	18094409
Mak Mei Qi	20067484
Woo Xin Ping	19033604

- This final assessment (Assignment 2) contributes 50% to your final grade.
- This FIVE-member group assignment is for Course Learning Outcome 2 - Implement a database design group project using appropriate tools such as Oracle SQL.
- Each member of the group is required to present his/her part of the work.

IMPORTANT

- The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.
- Coursework submitted after the deadline but within 1 week will be accepted for a maximum mark of 40%.
- Work handed in following the extension of 1 week after the original deadline will be regarded as a non-submission and marked zero.

Academic Honesty Acknowledgement

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

Alicia, Yiqing , Kusselin , Meigi , Xinping (8th July 2021)

(Student signatures/Date)

Table of Contents

Part 1: Construct a case scenario 4
1.1 Company's description 4
1.2 Problem statement 4
1.3 Business Requirements 5
1.4 Business Rules 6
1.5 Sample User Queries Scenario 7
Part 2: Design a database 8
2.1 Flowchart Diagram 8
2.2 Entity Relationship Diagram (ERD) 9
2.3 Relational Database Model (RDM) 10
Part 3: Implement a database 11
3.1 List of Attributes with their datatypes 11
3.2 User Check Constraints 14
3.3 Index List 15
3.4 Database Script 16
Part 4: Query a database 24
4.1 User Queries SQL Commands and Output 24
4.2 Views 28

Part 1: Construct a case scenario

Apple Inc

1.1 Company's description

Apple Inc is an American multinational technology company that was founded in the 1970s by Steven Jobs. It is a company that designs, develops, and sells consumer electronics, computer software, personal computers, smartphones, tablets, and online services. It was the first successful personal computer company and the popularizer of the graphical user interface. Apple's corporate mission is "to bring the best personal computing products and support to students, educators, designers, scientists, engineers, businesspersons and consumers in over 140 countries around the world." This mission statement considers the changing business landscape that influences the possibilities of what Apple Inc. can do. For instance, the company recognizes trends and changes in the consumer electronics market and the industry environment.

1.2 Problem statement

The demand for Apple products has increased during the COVID-19's global pandemic. According to Apple's CEO Tim Cook, customers are depending on Apple products in renewed ways to stay connected, informed, creative, and productive due to the pandemic, especially when meetings and classes are held online. This results in the increase of data stored in the company's database system. However, with the enormous volume of data, the real-time performance for updating records in the database has decreased. Moreover, the current poorly designed database system has induced data redundancies, data anomalies and data inconsistency. Poor data flows, delays in data-sharing agreements and a general lack of structuring and data integration have throttled timely data sharing and analysis. For instance, Apple's database faces problems when many customers shop for products online. When many customers add or remove a specific product in their shopping carts at the same time, the database faces a huge traffic flow and is unable to update the stock amount accurately. Furthermore, due to the COVID-19 standard operating procedures (SOP), there is a change of number of employees in each physical store. This sudden change caused confusion among managers and employees, as they were unsure of which store they are now in charge of carrying out their duties.

Due to high traffic in the system, it may result in frequent system delays or shutdowns. This happens when Apple is introducing some new gadgets into the market and mailshot campaigns is being carried out, customers are camping on the website and constantly refreshing the page in order to get the newly released Apple products which will have a higher possibility of

causing the website to crash. In view of these issues, the current database must be normalized and improved to ensure that the company's operating performance can be improved.

1.3 Business Requirements

The company requires a database that allows them to track customers (in store & online), monitor inventory levels of the company's products, orders, premises, mailshot campaigns and employee information.

For in store purchases, **customers** are required to provide their personal information such as their first name, surname, birth of date, gender, and contact number before purchasing any apple products.

Apple account table is created where each apple id is unique as it is registered under an email address of a customer, customer id is being recorded as well as their passwords. Each customer can have several apple accounts.

For online purchases, customers who are purchasing online must include their shipping address in the checkout page before making any payments. Each shipping address must include street, city, postal code, and country. Customers can either choose to purchase from Apple's physical store or Apple's online store with their apple account.

Each **product** has a unique id, a name, price per unit and product category. The product categories are namely PCs, iPads, iPhones, MacBooks, watches, writing tools, locating devices and music accessories. Writing tool is mainly Apple Pencil, locating device is Air Tag, and music accessories includes AirPods.

Each **order** information includes order id, order date and time, customer id, product id, employee id, payment type, shipping option and shipping id. Employee id is noted down for each order so that the company can track which employee is responsible for which orders made. Each transaction may be paid by credit card, debit card, cash, or check. Customers can choose to either purchase in-store, pickup in-store or delivery service. Since some orders do not require delivery service, not every order has shipping details and ship id. A **shipping details** table is needed to keep records of the ship id, shipping address, city, state, postcode, and country. The ship id is the parcel tracking number for each parcel.

A **premise** table is required to keep records of premise id, premise type, and premise address. The premise types are namely office, warehouse, and physical store. Most product is

stored in the warehouse or physical store. Office is mainly a workplace for employees to hold meetings and handle online orders; therefore, there is no product stored in the office.

An **employee** table is needed so that details of employees such as employee id, name, gender, birthdate, contact, premise ID (workplace ID), date hired, date resigned, position, person to be reported to, and monthly salary. Some employees might need to report to other employees, which could be their manager or someone who has a higher position in the company.

A **product stock** table is required to record data about the whereabouts of each stock in each type of premises. This table records the quantity of each product in each physical store and warehouse.

A **mailshot campaign** table is prepared to keep track of any campaigns being held. This table includes mailshot id, mailshot name, start date, and end date. Mailshot campaigns are used to contact large groups of apple account users and inform that about the new promotion the company has to offer them. The company also needs to know outcome of every mailshot to apple account users. Outcome can be either no response or order obtained. Some Apple account users might choose to unsubscribe from receiving newsletter from Apple.

1.4 Business Rules

1. Each **customer** owns zero or many apple **accounts.**
2. Each **apple account** is owned by only one **customer**.
3. Each **apple account** is associated with zero or many **mailshot campaigns**.
4. Each **mailshot campaign** is associated with many **apple accounts**.
5. Each **customer** makes zero or many **orders**.
6. Every **order** is made by only one **customer**.
7. Every **order** must be associated with at least one valid **product**.
8. Each **product** may be associated with zero or many **orders**.
9. Each **product** may be found in one or many **premises**.
10. Each **premise** may have zero or many **products**.
11. Each **premise** employs one or many **employees**.
12. Each **employee** works at one **premise**.
13. Each **employee** reports to zero or one **employee**.
14. Each **employee** manages zero or many **employees**
15. Each **employee** processes zero or many **orders**.
16. Each **order** is processed by an **employee**.
17. Each **order** has zero or one **ship_details**.
18. Each **ship_details** is associated with one **order**.

1.5 Sample User Queries Scenario

1. The Apple Company would like to know the Apple product with the highest number of sales in the month of June, as well as the total sales of the product during this month.

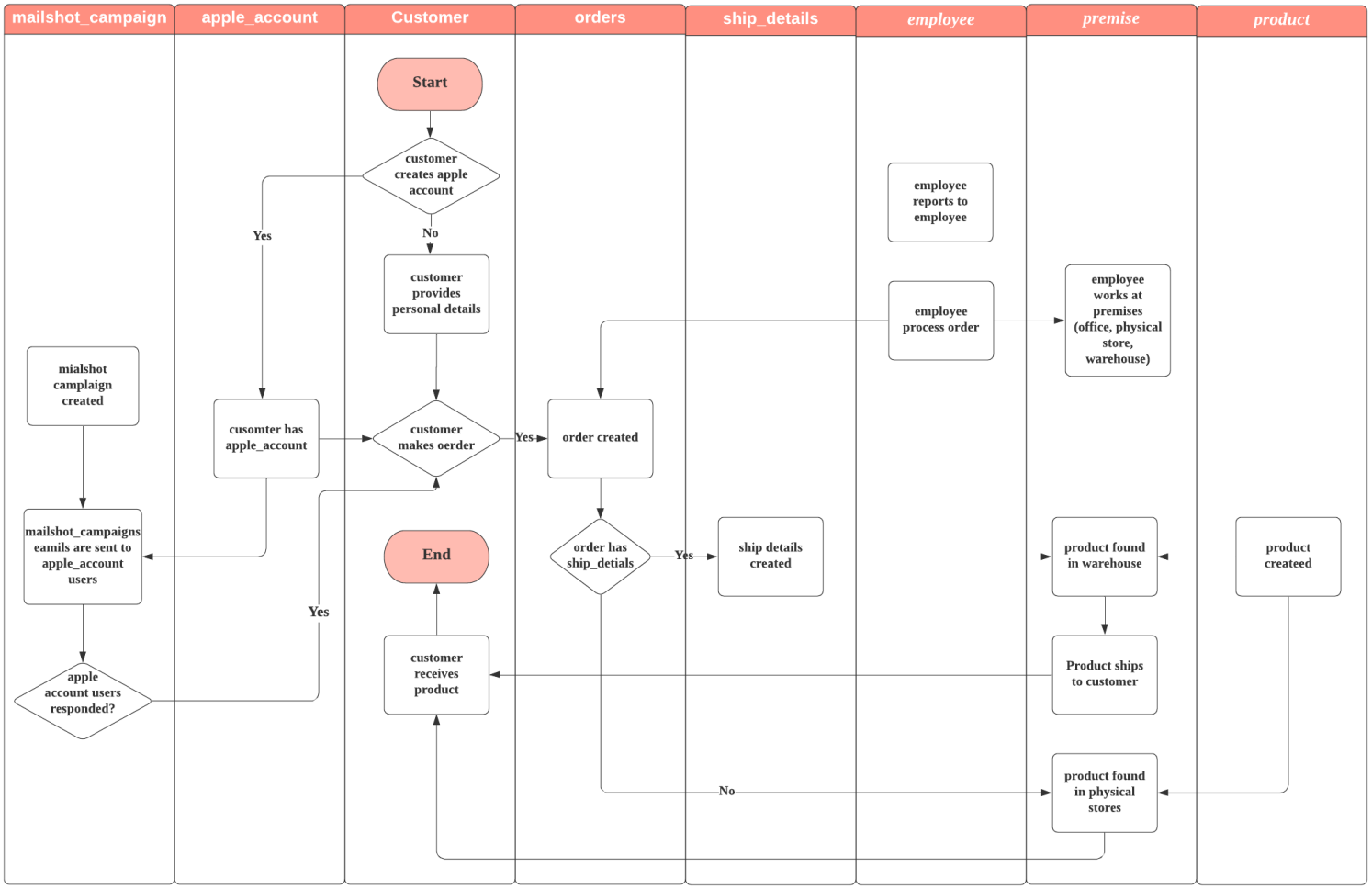
2. With the increasing amount of positive response of Apple products, the Apple Company would like to have a report that displays the product name and product stock of each warehouse and store to keep track of product stock. Only products that are labelled as MacBook Air and premise type is store and warehouse should be listed out in this report. The record must be sorted by premise ID.

3. Apple has released the new iPhone 12 series on the 13th of October 2020. On account of the Covid-19 pandemic, many consumers were affected financially. Therefore, the company has plans to help consumers in need, such as students and young adults, by providing them discounts. However, before that, the company needs to have a better understanding of their consumers' needs. As such, the company would like to know about the demographic information and preference of consumers who bought at least one of the iPhone 12 series during the pandemic. The company wants a report that lists out the ordered iPhone 12 model name, order date, customer's age, customer's gender, customer's shipping option and ship to country. This market research only targets customers who are aged between 18 and 23 and bought at least one of the iPhone 12 series between the 13th of October 2021 and the 12th of July 2021.

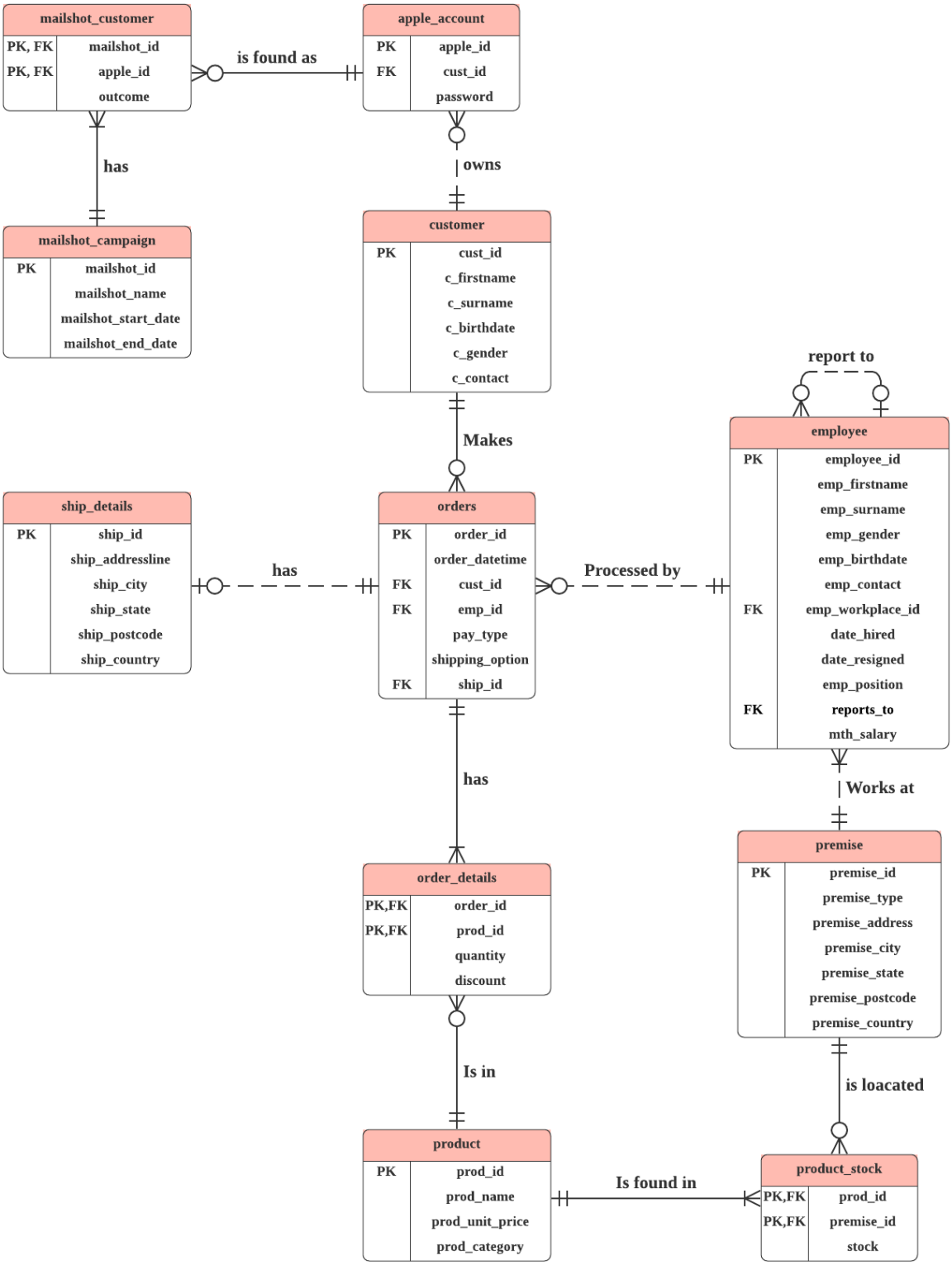
4. Customers who are not satisfied and/or have purchased a faulty product have a maximum of 30 days to return their products. The Apple Company would like to have a clear table to see the customer IDs with the product purchased, and the days they have bought the specific product. Customers with purchase days for more than 30 days will not be eligible to return their products.

Part 2: Design a database

2.1 Flowchart Diagram



2.2 Entity Relationship Diagram (ERD)



2.3 Relational Database Model (RDM)

Customer (**cust_id**, c_firstname, c_surname, c_birthdate, c_gender , c_contact)

Apple_account (**apple_id**, password, *cust_id*)
FOREIGN KEY cust_id REFERENCES TABLE customer (cust_id)

Mailshot_campaign (mailshot_id, mailshot_name, mailshot_start_date, mailshot_end_date)

Mailshsot_customer (mailshot_id, apple_id, outcome)
FOREIGN KEY apple_id REFERENCES TABLE apple_account (apple_id)
FOREIGN KEY mailshot_id REFERENCES TABLE mailshot_campaign (mailshot_id)

Employee (**emp_id**, emp_firstname, emp_surname, emp_gender, emp_birthdate, emp_contact, *emp_workplace_id*, date_hired, date_resigned, emp_position, *reports_to*, mth_salary)
FOREIGN KEY emp_workplace_id REFERENCES TABLE premises (premise_id)
FOREIGN KEY reports_to REFERENCES TABLE employee (emp_id)

Product (**prod_id**, prod_name, prod_category, prod_unit_price)

Order_details (***order_id, prod_id,*** quantity, discount)
FOREIGN KEY order_id REFERENCES TABLE orders (order_id)
FOREIGN KEY prod_id REFERENCES TABLE product (prod_id)

Orders (**order_id,** order_datetime, *cust_id*, *emp_id*, pay_type, shipping_option, *ship_id*)
FOREIGN KEY cust_id REFERENCES TABLE customer (cust_id)
FOREIGN KEY emp_id REFERENCES TABLE employee (employee_id)
FOREIGN KEY ship_id REFERENCES TABLE ship_address (ship_id)

Ship_details (ship_id, ship_addressline, ship_city, ship_state, ship_postcode, ship_country)

Premise (**premise_id**, premise_type, premise_address, premise_city, premise_state, premise_postcode, premise_country)

Product_stock (***prod_id, premise_id***, stock)
FOREIGN KEY prod_id REFERENCES TABLE product (prod_id)
FOREIGN KEY premise_id REFERENCES TABLE premise (premise_id)

Part 3: Implement a database

3.1 List of Attributes with their datatypes

Attribute	Description	Data Type	PK / FK	NOT NULL?	Justification
Name of Table: customer					
cust_id	Unique ID for each customer	NUMBER(5)	PK	NOT NULL	Customer ID can accept numbers with 5 digits only.
c_firstname	Customer's First name	VARCHAR2(20)			Customer's first name is a text that has variable length of 20.
c_surname	Customer's surname	VARCHAR2(20)			Customer's surname is a text that has variable length of 20.
c_birthdate	Customer's birthdate (DD/MON/YYYY)	DATE			Customer's birthdate is a date value that can accept day, month, year in a specific format.
c_gender	Customer's gender (F (female), M (male))	CHAR(1)			Customer's gender is a text which only accepts one character.
c_contact	Customer's contact number	VARCHAR2(15)		NOT NULL	Customer's contact is a text that has variable length of 15.
Name of Table: apple_account					
apple_id	Apple account users registered email address	VARCHAR2(30)	PK	NOT NULL	Apple ID is a text that could accept variable length of 30.
password	Apple account user's password	VARCHAR2(40)		NOT NULL	Password is a text that has variable length of 40.
cust_id	Unique ID for each customer	NUMBER (5)	FK		Customer ID can accept numbers with 5 digits only.
Name of Table: employee					
emp_id	Unique ID for each employee	VARCHAR2(10)	PK	NOT NULL	Employee id is a text that has variable length of 10.
emp_firstname	Employee's first name	VARCHAR2(20)		NOT NULL	Employee's first name is a text that has variable length of 20.
emp_surname	Employee's surname	VARCHAR2(20)		NOT NULL	Employee's surname is a text that has variable length of 20.
emp_gender	Employee's gender	CHAR(1)		NOT NULL	Employee's gender is a text which only accepts one character.
emp_birthdate	Employee's birthdate	DATE		NOT NULL	Employee's birthdate is a date value that can accept day, month, year in a specific format.

emp_contact	Employee's contact	VARCHAR2(15)		NOT NULL	Employee's contact is a text that has variable length of 15.
emp_workplace_id	Employee's workplace ID	VARCHAR2(10)	FK	NOT NULL	Workplace ID is a text that has variable length of 10.
date_hired	Employee's hiring date	DATE		NOT NULL	Employee's hiring date is a date value that can accept day, month, year in a specific format.
date_resigned	Employees resign date.	DATE			Employee's resign date is a date value that can accept day, month, year in a specific format.
emp_position	Employee's position	VARCHAR2(40)		NOT NULL	Employee's position is a text that has variable length of 40.
reports_to	Reports to manager's employee ID	VARCHAR2(10)	FK		Reports to is a text that has variable length of 10.
mth_salary	Employee's monthly salary	NUMBER(10)		NOT NULL	Monthly salary can accept numbers with 10 digits only
Name of Table: product					
prod_id	Unique ID for each product	NUMBER(3)	PK	NOT NULL	Product ID can accept numbers with 3 digits only
prod_name	Product name	VARCHAR2(20)		NOT NULL	Product name is a text that has variable length of 20.
prod_category	Product Category	VARCHAR2(20)		NOT NULL	Product Category is a text that has variable length of 20.
prod_unit_price	Product Unit price	NUMBER(10,2)		NOT NULL	Product Unit price can accept numbers with 10 digits with 2 decimals only.
Name of Table: orders					
Order_id	Unique ID for order	CHAR(4)	PK	NOT NULL	Order ID is a text which only accepts 4 characters.
order_datetime	Order date and time	TIMESTAMP		NOT NULL	Order datetime includes day, month and year with timeline in one statement.
cust_id	Customer's ID	NUMBER (5)	FK	NOT NULL	Customer ID can accept numbers with 5 digits only
emp_id	Employee's ID	VARCHAR2(10)	FK	NOT NULL	Employee's ID is a text that has variable length of 10.
Pay_type	Payment type	VARCHAR2(20)		NOT NULL	Payment type is a text that has variable length of 20.
Shipping_option	Shipping option for the orders	VARCHAR2(40)		NOT NULL	Shipping option is a text that has variable length of 40.

Ship_id	Shipping ID (parcel tracking number)	VARCHAR2(15)	FK		Shipping ID is a text that has variable length of 15.
Name of Table: order_details					
Order_id	Unique ID for order	CHAR(4)	PK/FK	NOT NULL	Order ID is a text which only accepts 4 characters.
prod_id	Unique ID for product	NUMBER(3)	PK/FK	NOT NULL	Product ID can accept numbers with 3 digits only
quantity	Order quantity	NUMBER(1)		NOT NULL	Order quantity can accept numbers with one digit only
discount	Discount for orders	NUMBER(10,2)			Discount can accept numbers with 10 digits and 2 decimals only
Name of Table: ship_details					
ship_id	Shipping_id (parcel tracking number)	VARCHAR2(15)	PK	NOT NULL	Shipping ID is a text that has variable length of 15.
ship_addressline	Shipping addressline	VARCHAR2(100)		NOT NULL	Shipping addressline is a text that has variable length of 100.
ship_city	Shipping city	VARCHAR2(20)		NOT NULL	Shipping city is a text that has variable length of 20.
ship_state	Shipping state	VARCHAR2(20)		NOT NULL	Shipping state is a text that has variable length of 20.
ship_postcode	Shipping postcode	NUMBER(5)		NOT NULL	Shipping postcode can accept numbers with 5 digits only
ship_country	Shipping country	VARCHAR2(20)		NOT NULL	Shipping country is a text that has variable length of 20.
Name of Table: premise					
Premise_id	Unique ID for premises	VARCHAR2(10)	PK	NOT NULL	Premise ID is a text that has variable length of 10.
Premise_type	Premise type	CHAR(10)		NOT NULL	Premise type is a text which only accepts 10 characters.
Premise_address	Premise address	VARCHAR2(100)		NOT NULL	Premise address is a text that has variable length of 100.
premise_city		VARCHAR(20)		NOT NULL	Premise city is a text that has variable length of 20.
premise_state		VARCHAR(20)		NOT NULL	Premise state is a text that has variable length of 20.
premise_postcode		NUMERIC(5)		NOT NULL	Premise postcode can accept numbers with 5 digits only
premise_country		VARCHAR2(20)		NOT NULL	Premise country is a text that has variable length of 20.
Name of Table: product_stock					

prod_id	Unique ID for product	NUMBER(3)	PK/FK	NOT NULL	Product ID can accept numbers with 3 digits only
premise_id	Premise ID	VARCHAR2(10)	PK/FK	NOT NULL	Premise ID is a text that has variable length of 10.
Stock	Stock	NUMBER(10)		NOT NULL	Stock can accept numbers with 10 digits only
Name of Table: mailshot_customer					
Mailshot_id	Unique ID for each mailshot campaign	CHAR(4)	PK/FK	NOT NULL	Mailshot ID is a text which only accepts 4 characters.
Apple_id	Apple's mailshot subscribers	VARCHAR2(30)	PK/FK	NOT NULL	Apple ID is a text that has variable length of 30.
Outcome	Response from subscribers	VARCHAR2(30)			Outcome is a text that has variable length of 30.
Name of Table: mailshot_campaign					
mailshot_id	Unique ID for each mailshot campaign	CHAR(4)	PK	NOT NULL	Mailshot ID is a text which only accepts 4 characters.
mailshot_name	Mailshot name	VARCHAR2(40)		NOT NULL	Mailshot name is a text that has variable length of 40.
mailshot_start_date	Start date of the mailshot campaign	DATE		NOT NULL	Start date is a date value that can accept day, month, year in a specific format.
mailshot_end_date	End date for the mailshot campaign	DATE			End date is a date value that can accept day, month, year in a specific format.

3.2 User Check Constraints

1. Customer gender type can either be 'F' or 'M' only.
2. Premise type must be either 'Office',' Store' or 'Warehouse'
3. Employee gender type can either be 'F' or 'M' only.
4. Payment type is either 'Cash', 'Check', 'Credit Card', 'Debit Card' or 'Online Banking'.
5. Shipping options are 'In-store Purchase', 'Delivery Service' and 'In-store Pickup'.
6. Product id must be BETWEEN 100 and 200.

3.3 Index List

Tables	Need Indexing?	Reason
order_details	No	This table is heavily updated, therefore, indexing it would decrease the data update efficiency.
orders	No	This table is heavily updated, therefore, indexing it would decrease the data update efficiency.
ship_details	No	This table is heavily updated, therefore, indexing it would decrease the data update efficiency.
product_stock	Yes	Attributes prod_id and premise_id are not frequently updated, but they are heavily queried.
	CREATE INDEX idx_prod_id ON product_stock (prod_id); **CREATE INDEX idx_premise_id ON product_stock (premise_id);**	
product	No	Table is not large as there isn't much variety of Apple's products.
employee	Yes	This table is frequently queried but not heavily updated.
	CREATE INDEX idx_emp_surname ON employee (emp_surname, emp_firstname); **CREATE INDEX idx_emp_workplace_id ON employee (emp_workplace_id);** **CREATE INDEX idx_emp_position ON employee (emp_position);**	
premise	Yes	This table is frequently queried but not heavily updated.
	CREATE INDEX idx_premise_type ON premise (premise_type); **CREATE INDEX idx_loc_country ON premise (loc_country);** **CREATE INDEX idx_loc_postcode ON premise (loc_postcode);**	
mailshot_customer	No	This table is heavily updated, therefore, indexing it would decrease the data update efficiency.
mailshot_campaign	Yes	Creating an index on attribute mailshot_name can improve data retrieval speed whereas slower data update speed is acceptable as table data is not heavily updated.
	CREATE INDEX idx_mailshot_name ON mailshot_campaign (mailshot_name);	
apple_account	No	There isn't any attribute that can
customer	Yes	Retrieval speed for retrieving the customer's information by indexing the customer's surname and first name can be increased.
	CREATE INDEX idx_c_name ON customer (c_surname, c_firstname);	

3.4 Database Script

* Remarks: Only parts of the insert values are included in this document. Please refer to the sql file for the full script.

```
ALTER SESSION SET NLS_DATE_FORMAT='DD-MON-YYYY HH24:MI';
DROP TABLE order_details;
DROP TABLE orders;
DROP TABLE ship_details;
DROP TABLE product_stock;
DROP TABLE product;
DROP TABLE employee;
DROP TABLE premise;
DROP TABLE mailshot_customer;
DROP TABLE mailshot_campaign;
DROP TABLE apple_account;
DROP TABLE customer;
-----------------------------------
-- customer table
-----------------------------------
CREATE TABLE customer (
    cust_id NUMBER(5) PRIMARY KEY,
    c_firstname VARCHAR2(20),
    c_surname VARCHAR2(20),
    c_birthdate DATE,
    c_gender CHAR(1) CONSTRAINT check_cgender CHECK (c_gender IN ('F','M')),
    c_contact VARCHAR2(15) NOT NULL
);
INSERT INTO customer VALUES (50001,'Isabelle','Olson','10-SEP-2001','F','410-562-8643');
INSERT INTO customer VALUES (50002,'Daphne','Morrison','31-JAN-2000','F','223-636-2238');
INSERT INTO customer VALUES (50003,'Ray','Green','05-MAR-1995','M','440-731-6693');
INSERT INTO customer VALUES (50004,'Nicholas','Sparks','21-MAY-1998','M','520-828-3824');
INSERT INTO customer VALUES (50005,'Joshua','McKenzie','04-AUG-1999','M','207-256-2765');
INSERT INTO customer VALUES (50006,'Fiona','Brady','08-JUL-2000','F','6011-290-0414');
INSERT INTO customer VALUES (50007,'Bethany','Manning','02-MAR-1995','F','724-379-7733');
INSERT INTO customer VALUES (50008,'Simon','Hall','07-DEC-1993','M','206-267-6945');
INSERT INTO customer VALUES (50009,'Desmond','Kim','21-OCT-2001','M','304-458-1441');
INSERT INTO customer VALUES (50010,'Taylor','Swift','30-JUN-2000','M','6010-405-3587');

CREATE INDEX idx_c_name ON customer (c_surname, c_firstname);
------------------------------
-- apple account table
------------------------------
CREATE TABLE apple_account (
    apple_id VARCHAR2(30) PRIMARY KEY,
    cust_id NUMBER(5),
```

```
    password VARCHAR2(40) NOT NULL,
    FOREIGN KEY (cust_id) references customer(cust_id)
);
INSERT INTO apple_account VALUES ('isabelleolson@gmail.com',50001,'isa23J');
INSERT INTO apple_account VALUES ('daphnemorrison@gmail.com',50002,'d4phn3#');
INSERT INTO apple_account VALUES ('raygreen@outlook.com',50003,'r4ygreEn');
INSERT INTO apple_account VALUES ('nicsparks@gmail.com',50004,'panicspark123');
INSERT INTO apple_account VALUES ('joshmck@outlook.com',50005,'josh277*B');
INSERT INTO apple_account VALUES ('fionabrady@gmail.com',50006,'fionab993q');
INSERT INTO apple_account VALUES ('bethmanning@outlook.com',50007,'breathingH2o');
INSERT INTO apple_account VALUES ('simonhall@gmail.com',50008,'halloffame123');
INSERT INTO apple_account VALUES ('desmondkim@outlook.com',50009,'kpossible99');
INSERT INTO apple_account VALUES ('taylorswift@gmail.com',50010,'fearless13');
-----------------------------------------------
-- mailshot campaign table
-----------------------------------------------
CREATE TABLE mailshot_campaign (
    mailshot_id CHAR(4) PRIMARY KEY,
    mailshot_name VARCHAR2(40) NOT NULL,
    mailshot_start_date DATE NOT NULL,
    mailshot_end_date DATE
);
INSERT INTO mailshot_campaign VALUES ('MS01','Black friday sales','23-NOV-2018','23-NOV-2018');
INSERT INTO mailshot_campaign VALUES ('MS02','Black friday sales','29-NOV-2019','29-NOV-2019');
INSERT INTO mailshot_campaign VALUES ('MS03','Black friday sales','27-NOV-2020','27-NOV-2020');
INSERT INTO mailshot_campaign VALUES ('MS04','11.11 sale','01-NOV-2018','11-NOV-2018');
INSERT INTO mailshot_campaign VALUES ('MS05','11.11 sale','01-NOV-2019','11-NOV-2019');
INSERT INTO mailshot_campaign VALUES ('MS06','11.11 sale','01-NOV-2020','11-NOV-2020');
INSERT INTO mailshot_campaign VALUES ('MS07','Spring Clearance','22-JUN-2018','29-JUN-2018');
INSERT INTO mailshot_campaign VALUES ('MS08','Spring Clearance','22-JUN-2019','29-JUN-2019');
INSERT INTO mailshot_campaign VALUES ('MS09','Spring Clearance','22-JUN-2020','29-JUN-2020');
INSERT INTO mailshot_campaign VALUES ('MS10','Happy education offer','01-JAN-2019','07-JAN-2019');

CREATE INDEX idx_mailshot_name ON mailshot_campaign (mailshot_name);
-----------------------------------------
-- mailshot customer table
-----------------------------------------
CREATE TABLE mailshot_customer (
    mailshot_id CHAR(4),
    apple_id VARCHAR2(30),
    outcome VARCHAR2(30),
    PRIMARY KEY (mailshot_id, apple_id),
    FOREIGN KEY (mailshot_id) references mailshot_campaign (mailshot_id),
    FOREIGN KEY (apple_id) references apple_account (apple_id)
);
```

```
INSERT INTO mailshot_customer VALUES ('MS01','isabelleolson@gmail.com','No Response');
INSERT INTO mailshot_customer VALUES ('MS01','daphnemorrison@gmail.com','No Response');
INSERT INTO mailshot_customer VALUES ('MS01','raygreen@outlook.com','Order Obtained');
INSERT INTO mailshot_customer VALUES ('MS01','nicsparks@gmail.com','Order Obtained');
INSERT INTO mailshot_customer VALUES ('MS01','joshmck@outlook.com','Order Obtained');
INSERT INTO mailshot_customer VALUES ('MS01','fionabrady@gmail.com','No Response');
INSERT INTO mailshot_customer VALUES ('MS02','bethmanning@outlook.com','Order Obtained');
INSERT INTO mailshot_customer VALUES ('MS02','simonhall@gmail.com','Order Obtained');
INSERT INTO mailshot_customer VALUES ('MS02','desmondkim@outlook.com','Order Obtained');
INSERT INTO mailshot_customer VALUES ('MS02','taylorswift@gmail.com','Order Obtained');
-----------------------------------
-- premise table
-----------------------------------
CREATE TABLE premise (
    premise_id VARCHAR2(10) PRIMARY KEY,
    premise_type CHAR(10) NOT NULL CONSTRAINT check_premise CHECK (premise_type IN ('Office','Store','Ware
house')),
    premise_address VARCHAR2(100) NOT NULL,
    premise_city VARCHAR(20) NOT NULL,
    premise_state VARCHAR(20) NOT NULL,
    premise_postcode NUMERIC(5) NOT NULL,
    premise_country VARCHAR2(20) NOT NULL
);
INSERT INTO premise VALUES ('WH001CA','Warehouse','654 S Myers St','Los Angeles','California',94027,'USA')
;
INSERT INTO premise VALUES ('WH002CA','Warehouse','655 S Myers St','Los Angeles','California',94027,'USA')
;
INSERT INTO premise VALUES ('OF001LA','Office','W Jefferson Dr','Los Angeles','California',94027,'USA');
INSERT INTO premise VALUES ('OF002LA','Office','10000 Washington Blvd','Los Angeles','California',94027,'U
SA');
INSERT INTO premise VALUES ('ST001LA','Store','189 The Grove Dr','Los Angeles','California',94027,'USA');
INSERT INTO premise VALUES ('ST002SD','Store','4305 La Jolla Village Dr','San Diego','California',94027,'U
SA');
INSERT INTO premise VALUES ('WH001TX','Warehouse','15505 Long Vista Dr # 210','Austin','Texas',20001,'USA'
);
INSERT INTO premise VALUES ('WH002TX','Warehouse','8002 Burleson Rd','Austin','Texas',20001,'USA');
INSERT INTO premise VALUES ('ST001AT','Store','3121 Palm Way','Austin','Texas',20001,'USA');
INSERT INTO premise VALUES ('OF001HT','Office','5085 Westheimer Rd','Houston','Texas',20001,'USA');

CREATE INDEX idx_premise_type ON premise (premise_type);
CREATE INDEX idx_loc_country ON premise (loc_country);
CREATE INDEX idx_loc_postcode ON premise (loc_postcode);

-----------------------------------
```

```
-- employee table
-----------------------------------
CREATE TABLE employee (
    emp_id VARCHAR2(10) PRIMARY KEY,
    emp_firstname VARCHAR2(20) NOT NULL,
    emp_surname VARCHAR2(20) NOT NULL,
    emp_gender CHAR(1) NOT NULL CONSTRAINT check_egender CHECK (emp_gender IN ('F','M')),
    emp_birthdate DATE NOT NULL,
    emp_contact VARCHAR2(15) NOT NULL,
    emp_workplace_id VARCHAR2(10) NOT NULL,
    date_hired DATE NOT NULL,
    date_resigned DATE,
    emp_position VARCHAR2(40) NOT NULL,
    reports_to VARCHAR2(10),
    mth_salary NUMBER(10) NOT NULL,
    FOREIGN KEY (emp_workplace_id) references premise(premise_id)
);
INSERT INTO employee VALUES ('RE100','Joe','Jonas','M','06-MAY-2000','215-443-8345','ST001LA','08-AUG-
2012','06-DEC-2015','Retail','COO100',2500);
INSERT INTO employee VALUES ('SE100','Benny','Sim','M','27-SEP-1985','725-389-2138','OF001LA','13-OCT-
2012',NULL,'Software Engineer','SSM100',8000);
INSERT INTO employee VALUES ('COO100','Takashi','Marumoto','M','16-JAN-1992','223-532-6793','OF001LA','06-
NOV-2013',NULL,'Chief Operating Officer','SCOO100',9100);
INSERT INTO employee VALUES ('MAN100','Selena','Gomez','F','11-APR-1987','520-327-3424','ST002SD','26-JUL-
2016',NULL,'Manager','SM200',5500);
INSERT INTO employee VALUES ('MAN200','Tamia','Nguyen','F','28-FEB-1991','207-316-7256','WH001TX','04-JUL-
2015','17-JUL-2016','Manager','SNM100',5500);
INSERT INTO employee VALUES ('RE200','Chris','Perry','M','30-MAR-1991','318-930-6494','ST001LA','25-SEP-
2019',NULL,'Retail','COO100',2500);
INSERT INTO employee VALUES ('MAR100','Lilian','Carroll','F','30-OCT-1994','725-609-7733','ST001AT','24-
SEP-2016',NULL,'Marketing','AD100',6000);
INSERT INTO employee VALUES ('HE100','Joel','Tan','M','08-JUL-1980','215-638-7345','WH001NY','16-NOV-
2009',NULL,'Hardware Engineer','SSM100',7500);
INSERT INTO employee VALUES ('MAN300','Suzuki','Ayato','M','20-APR-1976','304-473-1421','WH002TX','22-SEP-
2010',NULL,'Manager','SNM100',5500);
INSERT INTO employee VALUES ('RE300','Nellie','Palmer','F','19-NOV-1995','610-957-3247','ST002SD','17-MAY-
2016','01-JUN-2020','Retail','COO100',2500);

ALTER TABLE employee
ADD CONSTRAINT reports_to FOREIGN KEY(reports_to) REFERENCES employee (emp_id);
CREATE INDEX idx_emp_surname ON employee (emp_surname, emp_firstname);
CREATE INDEX idx_emp_workplace_id ON employee (emp_workplace_id);
CREATE INDEX idx_emp_position ON employee (emp_position);

-------------------------------------
```

```
-- product table
--------------------------------------
CREATE TABLE product (
    prod_id NUMBER(3) PRIMARY KEY
        CONSTRAINT check_prod_id CHECK (prod_id BETWEEN 100 and 200),
    prod_name VARCHAR2(20) NOT NULL,
    prod_unit_price NUMERIC(10, 2) NOT NULL,
    prod_category VARCHAR2(20) NOT NULL
);
INSERT INTO product VALUES (100,'27-inch iMac',359.99,'PC');
INSERT INTO product VALUES (101,'Mac mini',319.99,'PC');
INSERT INTO product VALUES (102,'M1 MacBook Pro',299.99,'MacBook');
INSERT INTO product VALUES (103,'M1 MacBook Air',249.99,'MacBook');
INSERT INTO product VALUES (104,'13-inch MacBook Pro',239.99,'MacBook');
INSERT INTO product VALUES (105,'MacBook Air',229.99,'MacBook');
INSERT INTO product VALUES (106,'iPhone SE',179.99,'iPhone');
INSERT INTO product VALUES (107,'iPhone 11',189.99,'iPhone');
INSERT INTO product VALUES (108,'iPhone 11 Pro',200.99,'iPhone');
INSERT INTO product VALUES (109,'iPhone 11 Pro Max',219.99,'iPhone');
INSERT INTO product VALUES (110,'iPhone 12 ',259.99,'iPhone');
INSERT INTO product VALUES (111,'iPhone 12 Pro',279.99,'iPhone');
INSERT INTO product VALUES (112,'iPhone 12 Pro Max',300.99,'iPhone');
INSERT INTO product VALUES (113,'iPad Pro',199.99,'iPad');
INSERT INTO product VALUES (114,'iPad Air',159.99,'iPad');
INSERT INTO product VALUES (115,'Apple Watch Series 6',89.99,'Watches');
INSERT INTO product VALUES (116,'Apple Watch SE',79.99,'Watches');
INSERT INTO product VALUES (117,'Airpods Pro',69.99,'Music Accesories');
INSERT INTO product VALUES (118,'AirPods Max',59.99,'Music Accesories');
INSERT INTO product VALUES (119,'Apple Pencil',49.99,'Writing Tool');
INSERT INTO product VALUES (120,'Air Tag',39.99,'Locating Device');
-----------------------------------
-- product stock table
-----------------------------------
CREATE TABLE product_stock (
    prod_id NUMBER(3),
    premise_id VARCHAR2(10),
    stock NUMERIC(10) NOT NULL,
    PRIMARY KEY (prod_id, premise_id),
    FOREIGN KEY (prod_id) references product (prod_id),
    FOREIGN KEY (premise_id) references premise (premise_id)
);
INSERT INTO product_stock VALUES (100,'ST001LA',47);
INSERT INTO product_stock VALUES (100,'WH001TX',11684);
INSERT INTO product_stock VALUES (100,'ST001AT',59);
INSERT INTO product_stock VALUES (100,'WH001NY',12032);
```

```
INSERT INTO product_stock VALUES (100,'ST001NY',99);
INSERT INTO product_stock VALUES (100,'WH001KL',1469);
INSERT INTO product_stock VALUES (100,'ST002KL',88);
INSERT INTO product_stock VALUES (100,'WH001TYO',1745);
INSERT INTO product_stock VALUES (100,'ST001AKH',102);
INSERT INTO product_stock VALUES (100,'WH002WL',1600);
INSERT INTO product_stock VALUES (100,'WH002ST',1405);
INSERT INTO product_stock VALUES (100,'ST002GC',64);
INSERT INTO product_stock VALUES (100,'ST001KB',58);
INSERT INTO product_stock VALUES (100,'ST001BJ',39);
INSERT INTO product_stock VALUES (100,'ST002BJ',74);
INSERT INTO product_stock VALUES (100,'ST003SH',58);
INSERT INTO product_stock VALUES (100,'WH001VAN',1945);
INSERT INTO product_stock VALUES (101,'WH001CA',12001);
INSERT INTO product_stock VALUES (101,'ST001LA',63);
INSERT INTO product_stock VALUES (101,'WH002NY',1761);
INSERT INTO product_stock VALUES (101,'ST001KL',55);

CREATE INDEX idx_prod_id ON product_stock (prod_id);
CREATE INDEX idx_premise_id ON product_stock (premise_id);

----------------------------------
-- ship details table
----------------------------------
CREATE TABLE ship_details (
    ship_id VARCHAR2(15) PRIMARY KEY,
    ship_addressline VARCHAR2(100) NOT NULL,
    ship_city VARCHAR2(20) NOT NULL,
    ship_state VARCHAR2(20) NOT NULL,
    ship_postcode NUMBER(5) NOT NULL,
    ship_country VARCHAR2(20) NOT NULL
);
INSERT INTO ship_details VALUES ('AP856665017','11807 Westheimer Rd #550','Houston','Texas',20001,'USA');
INSERT INTO ship_details VALUES ('AP554618648','3801 N Capital of Texas Hwy Ste E240','Austin','Texas',200
01,'USA');
INSERT INTO ship_details VALUES ('AP452979650','1234 Wilshire Blvd #104','Los Angeles','California',94027,
'USA');
INSERT INTO ship_details VALUES ('AP922731465','1235 Wilshire Blvd #104','Los Angeles','California',94027,
'USA');
INSERT INTO ship_details VALUES ('AP351526482','1236 Wilshire Blvd #104','Los Angeles','California',94027,
'USA');
INSERT INTO ship_details VALUES ('AP157898844','1237 Wilshire Blvd #104','Los Angeles','California',94027,
'USA');
INSERT INTO ship_details VALUES ('AP887761129','8601 NW 72nd St','Miami','Florida',10007,'USA');
INSERT INTO ship_details VALUES ('AP908848415','8602 NW 72nd St','Miami','Florida',10007,'USA');
```

```
INSERT INTO ship_details VALUES ('AP100686842','Fortune City One, 1 Ngan Shing St','Tai Long Wan','Sha Tin
',90033,'Hong Kong');
INSERT INTO ship_details VALUES ('AP223246166','Near West Police Station, Choumuhani','Agartala','Tripura'
,55327,'India');


--------------------------------
-- order table
--------------------------------
CREATE TABLE orders (
    order_id CHAR(4) PRIMARY KEY,
    order_datetime DATE NOT NULL,
    cust_id NUMBER(5) NOT NULL,
    emp_id VARCHAR2(10) NOT NULL,
    pay_type VARCHAR2(20) NOT NULL CONSTRAINT check_pay_type CHECK (pay_type IN ('Cash','Check','Credit Ca
rd','Debit Card','Online Banking')),
    shipping_option VARCHAR2(40) NOT NULL CONSTRAINT check_shipping_option CHECK (shipping_option IN ('In-
store Purchase','Delivery Service','In-store Pickup')),
    ship_id VARCHAR2(15),
    FOREIGN KEY (cust_id) references customer (cust_id),
    FOREIGN KEY (emp_id) references employee (emp_id),
    FOREIGN KEY (ship_id) references ship_details (ship_id)
);
INSERT INTO orders VALUES ('A001','10-JUN-2021 13:23',50001,'RE200','Cash','In-store Purchase',NULL);
INSERT INTO orders VALUES ('A002','10-JUN-2021 13:30',50002,'RE200','Check','In-store Purchase',NULL);
INSERT INTO orders VALUES ('A003','10-JUN-2021 14:35',50003,'RE200','Credit Card','In-store Pickup',NULL);
INSERT INTO orders VALUES ('A004','10-JUN-2021 14:44',50004,'RE400','Cash','In-store Purchase',NULL);
INSERT INTO orders VALUES ('A005','10-JUN-2021 14:45',50005,'RE400','Cash','In-store Purchase',NULL);
INSERT INTO orders VALUES ('A006','10-JUN-2021 15:50',50006,'RE400','Credit Card','In-store Pickup',NULL);
INSERT INTO orders VALUES ('A007','10-JUN-2021 15:55',50007,'RE400','Credit Card','In-store Pickup',NULL);
INSERT INTO orders VALUES ('A008','10-JUN-2021 16:56',50008,'RE400','Debit Card','In-store Pickup',NULL);
INSERT INTO orders VALUES ('A009','10-JUN-
2021 16:59',50009,'RE400','Online Banking','Delivery Service','AP856665017');
INSERT INTO orders VALUES ('A010','10-JUN-
2021 17:00',50010,'RE400','Credit Card','Delivery Service','AP554618648');
-----------------------------------
-- order details table
-----------------------------------
CREATE TABLE order_details (
    order_id CHAR(4),
    prod_id NUMBER(3),
    quantity NUMBer(1) NOT NULL,
    discount NUMBer(10, 2),
    PRIMARY KEY (order_id, prod_id),
    FOREIGN KEY (order_id) references orders(order_id),
    FOREIGN KEY (prod_id) references product(prod_id)
```

```
);
INSERT INTO order_details VALUES ('A001',111,1,0.1);
INSERT INTO order_details VALUES ('A002',119,2,0.15);
INSERT INTO order_details VALUES ('A003',103,2,0.2);
INSERT INTO order_details VALUES ('A004',120,2,0.1);
INSERT INTO order_details VALUES ('A005',107,3,0.2);
INSERT INTO order_details VALUES ('A006',100,1,0.1);
INSERT INTO order_details VALUES ('A007',112,1,0.3);
INSERT INTO order_details VALUES ('A008',114,4,0.1);
INSERT INTO order_details VALUES ('A009',103,1,0.2);
INSERT INTO order_details VALUES ('A010',117,1,0.2);
```

Part 4: Query a database

4.1 User Queries SQL Commands and Output

No	Query Description
1	**2 subqueries & 1 function** The Apple Company would like to know the Apple product with the highest number of sales in the month of June, as well as the total sales of the product during this month.

```
1  SELECT prod_name, SUM(quantity), SUM((prod_unit_price * quantity * (1 - discount))) AS "TOTAL SALES"
2    FROM product, orders, order_details
3   WHERE product.prod_id = order_details.prod_id
4     AND orders.order_id = order_details.order_id
5     AND TO_CHAR(order_datetime, 'MON') = 'JUN'
6  GROUP BY prod_name
7  HAVING SUM(quantity) = (SELECT MAX(sum_qty)
8      FROM (SELECT prod_id, SUM(quantity) AS sum_qty
9          FROM order_details
10         GROUP BY prod_id));
```

Results | Explain | Describe | Saved SQL | History

PROD_NAME	SUM(QUANTITY)	TOTAL SALES
Air Tag	21	755.811

1 rows returned in 0.07 seconds Download

2 **3 table join, 2 user conditions, GROUP BY**

With the increasing amount of positive response of Apple products, the Apple Company would like to have a report that displays the product name and product stock of each warehouse and store to keep track of product stock. Only products that are labelled as MacBook Air and premise type is store and warehouse should be listed out in this report. The record must be sorted by premise ID.

```
1  SELECT prod_name AS "PRODUCT NAME", premise_type, premise.premise_id, stock
2    FROM product, premise, product_stock
3   WHERE product.prod_name = 'MacBook Air'
4     AND premise.premise_type IN ('Store','Warehouse')
5     AND product.prod_id = product_stock.prod_id
6     AND product_stock.premise_id = premise.premise_id
7  GROUP BY prod_name ,premise_type, premise.premise_id, stock
8  ORDER BY premise.premise_id;
```

Results Explain Describe Saved SQL History

PRODUCT NAME	PREMISE_TYPE	PREMISE_ID	STOCK
MacBook Air	Store	ST001AKH	45
MacBook Air	Store	ST001COQ	52
MacBook Air	Store	ST001HRJ	78
MacBook Air	Store	ST001LA	45
MacBook Air	Store	ST001NY	75
MacBook Air	Store	ST001TLW	187
MacBook Air	Store	ST002BJ	47
MacBook Air	Store	ST002NY	99
MacBook Air	Store	ST002SD	68
MacBook Air	Warehouse	WH001BJ	1745
MacBook Air	Warehouse	WH001CA	17462
MacBook Air	Warehouse	WH001NY	18756
MacBook Air	Warehouse	WH001QL	1775
MacBook Air	Warehouse	WH001ST	1674
MacBook Air	Warehouse	WH001TYO	1756
MacBook Air	Warehouse	WH001WL	598

16 rows returned in 0.04 seconds Download

3 **Outer join & 3 user conditions (one of them must use LIKE keyword)**

Apple has released the new iPhone 12 series on the 13th of October 2020. On account of the Covid-19 pandemic, many consumers were affected financially. Therefore, the company has plans to help consumers in need, such as students and young adults, by providing them discounts. However, before that, the company needs to have a better understanding of their consumers' needs. As such, the company would like to know about the demographic information and preference of consumers who bought at least one of the iPhone 12 series between the 13th of October 2021 and the 12th of July 2021. The company wants a report that lists out the ordered iPhone 12 model name, order date, customer's age, customer's gender, customer's shipping option and ship to country. This market research only targets customers who are aged between 18 and 23.

```
SELECT prod_name AS "Product Name",
       TO_CHAR(order_datetime, 'DD-MON-YYYY') AS "Ordered date",
       TO_CHAR(sysdate, 'YYYY') - TO_CHAR(c_birthdate, 'YYYY') AS "Age",
       c_gender AS "Gender",
       pay_type AS "Payment Method",
       shipping_option AS "SHIPPING OPTION",
       ship_country AS "SHIP TO COUNTRY"
FROM product, customer, order_details, orders
LEFT JOIN ship_details ON orders.ship_id = ship_details.ship_id
WHERE product.prod_id = order_details.prod_id
  AND order_details.order_id = orders.order_id
  AND orders.cust_id = customer.cust_id
  AND prod_name LIKE 'iPhone 12%'
  AND TO_CHAR(sysdate, 'YYYY') - TO_CHAR(c_birthdate, 'YYYY') BETWEEN 18 AND 23
  AND order_datetime BETWEEN '13-OCT-2020' AND '12-JUL-2021'
```

Results | Explain | Describe | Saved SQL | History

Product Name	Ordered date	Age	Gender	Payment Method	SHIPPING OPTION	SHIP TO COUNTRY
iPhone 12 Pro	10-JUN-2021	20	F	Cash	In-store Purchase	-
iPhone 12 Pro Max	12-JUN-2021	23	M	Cash	In-store Purchase	-
iPhone 12 Pro Max	12-JUN-2021	20	M	Online Banking	Delivery Service	USA

3 rows returned in 0.03 seconds Download

4 Table aliases & date function

Customers who are not satisfied and/or have purchased a faulty product have a maximum of 30 days to return their products. The Apple Company would like to have a clear table to see the customer IDs with the product purchased, and the days they have bought the specific product. Customers with purchase days for more than 30 days will not be eligible to return their products.

```
1  SELECT o.cust_id AS Customer_ID,
2         order_details.prod_id AS Product_ID,
3         o.order_datetime,
4         ROUND(TO_CHAR(SYSDATE - o.order_datetime)) AS Days_Purchased
5  FROM orders o
6  INNER JOIN order_details ON o.order_id = order_details.order_id
7  WHERE ROUND(TO_CHAR(SYSDATE - o.order_datetime)) <= 30;
8
```

Results | Explain | Describe | Saved SQL | History

CUSTOMER_ID	PRODUCT_ID	ORDER_DATETIME	DAYS_PURCHASED
50001	111	2021-06-10 13:23	25
50002	119	2021-06-10 13:30	25
50003	103	2021-06-10 14:35	25
50004	120	2021-06-10 14:44	25
50005	107	2021-06-10 14:45	25
50006	100	2021-06-10 15:50	25
50007	112	2021-06-10 15:55	25
50008	114	2021-06-10 16:56	25
50009	103	2021-06-10 16:59	25
50010	117	2021-06-10 17:00	25

More than 10 rows available. Increase rows selector to view more rows.

4.2 Views

No	Views Description
1	EMPLOYEE_VIEW Shows employees who are still working with the company, their position, their manager, workplace HR/finance department can determine if the employees should get a pay raise (+salary)

SQL Code:

```
CREATE OR REPLACE FORCE VIEW "EMPLOYEE_VIEW" ("EMP_ID", "EMP_FIRSTNAME", "EMP_SURNAME",
 "EMP_WORKPLACE_ID", "DATE_HIRED", "EMP_POSITION", "MTH_SALARY") AS
    SELECT  EMPLOYEE.EMP_ID,
            EMPLOYEE.EMP_FIRSTNAME,
            EMPLOYEE.EMP_SURNAME,
            EMPLOYEE.EMP_WORKPLACE_ID,
            EMPLOYEE.DATE_HIRED,
            EMPLOYEE.EMP_POSITION,
            EMPLOYEE.MTH_SALARY
    FROM EMPLOYEE
    WHERE EMPLOYEE.DATE_RESIGNED IS NULL;
```

Results:

```
1  SELECT *
2  FROM employee_view
```

Results | Explain | Describe | Saved SQL | History

EMP_ID	EMP_FIRSTNAME	EMP_SURNAME	EMP_WORKPLACE_ID	DATE_HIRED	EMP_POSITION	MTH_SALARY
SE100	Benny	Sim	OF001LA	13-OCT-2012 00:00	Software Engineer	8000
COO100	Takashi	Marumoto	OF001LA	06-NOV-2013 00:00	Chief Operating Officer	9100
MAN100	Selena	Gomez	ST002SD	26-JUL-2016 00:00	Manager	5500
RE200	Chris	Perry	ST001LA	25-SEP-2019 00:00	Retail	2500
MAR100	Lilian	Carroll	ST001AT	24-SEP-2016 00:00	Marketing	6000
HE100	Joel	Tan	WH001NY	16-NOV-2009 00:00	Hardware Engineer	7500
MAN300	Suzuki	Ayato	WH002TX	22-SEP-2010 00:00	Manager	5500
HR100	Dexter	Chan	OF001HT	18-JAN-2017 00:00	Human Resources	3300
COO200	Chantelle	Lee	OF001LA	03-JAN-2011 00:00	Chief Operating Officer	9100
RE400	Christine	Ford	ST001NY	22-MAY-2019 00:00	Retail	2500

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.04 seconds Download

2	stock_view Shows the total stock of each product from all premises. This view allows the company to keep track of the total product stock without having to sum them up.

SQL Code:

```
CREATE OR REPLACE VIEW stock_view AS
    SELECT ps.prod_id AS "PRODUCT_ID", p.prod_name AS "PRODUCT NAME", SUM(ps.stock) AS
"STOCK"
    FROM product_stock ps, product p
    WHERE ps.prod_id = p.prod_id
    GROUP BY ps.prod_id, p.prod_name
    ORDER BY ps.prod_id;
```

Results:

```
1  SELECT *
2  FROM stock_view
```

Results | Explain | Describe | Saved SQL | History

PRODUCT_ID	PRODUCT NAME	STOCK
100	27-inch iMac	32568
101	Mac mini	20235
102	M1 MacBook Pro	51105
103	M1 MacBook Air	58473
104	13-inch MacBook Pro	61728
105	MacBook Air	44462
106	iPhone SE	25522
107	iPhone 11	87964
108	iPhone 11 Pro	43161
109	iPhone 11 Pro Max	35485

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.09 seconds Download

3 PRODUCT_SALES_VIEW

This view displays the details of each order as the customer's first name, surname, order date and time, product name, product unit price, quantity, discount and subtotal is displayed. This view shall allow the sales department to get a detailed information of each order.

SQL Code:

```
CREATE OR REPLACE FORCE VIEW "PRODUCT_SALES_VIEW" ("C_FIRSTNAME", "C_SURNAME", "ORDER_D
ATETIME", "PROD_NAME", "PROD_UNIT_PRICE", "QUANTITY", "DISCOUNT", "SUBTOTAL") AS
  select CUSTOMER.C_FIRSTNAME,
         CUSTOMER.C_SURNAME,
         ORDERS.ORDER_DATETIME,
         PRODUCT.PROD_NAME,
         PRODUCT.PROD_UNIT_PRICE,
         ORDER_DETAILS.QUANTITY,
         ORDER_DETAILS.DISCOUNT,
         PROD_UNIT_PRICE * QUANTITY * (1 - DISCOUNT) as SUBTOTAL
    from PRODUCT,ORDER_DETAILS,ORDERS,CUSTOMER
    where PRODUCT.PROD_ID = ORDER_DETAILS.PROD_ID
    and ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
    and CUSTOMER.CUST_ID = ORDERS.CUST_ID
    order by ORDER_DATETIME desc;
```

Results:

```
1  SELECT *
2  FROM PRODUCT_SALES_VIEW
```

Results | Explain | Describe | Saved SQL | History

C_FIRSTNAME	C_SURNAME	ORDER_DATETIME	PROD_NAME	PROD_UNIT_PRICE	QUANTITY	DISCOUNT	SUBTOTAL
Maddie	Smith	17-JUN-2021 12:00	Airpods Pro	69.99	2	.2	111.984
Melanie	Wong	16-JUN-2021 18:20	13-inch MacBook Pro	239.99	1	0	239.99
Gerald	Carter	16-JUN-2021 18:10	M1 MacBook Air	249.99	3	.2	599.976
Maya	Quinn	16-JUN-2021 18:00	iPhone 11 Pro Max	219.99	1	.15	186.9915
Jasmine	Marquez	16-JUN-2021 16:59	27-inch iMac	359.99	1	.1	323.991
Yasmin	Charles	16-JUN-2021 16:56	iPhone 12 Pro	279.99	1	.1	251.991
Serena	Willis	16-JUN-2021 15:55	iPhone SE	179.99	1	.2	143.992
Joe	Austin	15-JUN-2021 15:50	iPhone 11	189.99	2	.2	303.984
Jack	Lynch	15-JUN-2021 14:45	iPhone 11 Pro	200.99	3	0	602.97
Bill	Jenkins	15-JUN-2021 11:30	MacBook Air	229.99	2	0	459.98

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.08 seconds Download

4 Mailshot campaigns view

Shows mailshot campaigns and the customers that were part of the campaigns along with the outcomes. This view allows the marketing department to know which customer responded to the mailshot campaign.

SQL Code:

```
CREATE OR REPLACE FORCE VIEW "MAILSHOT_VIEW" ("MAILSHOT_NAME", "MAILSHOT_START_DATE", "
MAILSHOT_END_DATE", "C_FIRSTNAME", "C_SURNAME", "OUTCOME") AS
  select MAILSHOT_CAMPAIGN.MAILSHOT_NAME,
         MAILSHOT_CAMPAIGN.MAILSHOT_START_DATE,
         MAILSHOT_CAMPAIGN.MAILSHOT_END_DATE,
         CUSTOMER.C_FIRSTNAME,
         CUSTOMER.C_SURNAME,
         MAILSHOT_CUSTOMER.OUTCOME
    from CUSTOMER,APPLE_ACCOUNT,MAILSHOT_CUSTOMER,MAILSHOT_CAMPAIGN
    where MAILSHOT_CAMPAIGN.MAILSHOT_ID = MAILSHOT_CUSTOMER.MAILSHOT_ID
    and MAILSHOT_CUSTOMER.APPLE_ID = APPLE_ACCOUNT.APPLE_ID
    and APPLE_ACCOUNT.CUST_ID = CUSTOMER.CUST_ID
    order by MAILSHOT_END_DATE desc;
```

Results:

```
1  SELECT *
2  FROM MAILSHOT_VIEW
```

Results | Explain | Describe | Saved SQL | History

MAILSHOT_NAME	MAILSHOT_START_DATE	MAILSHOT_END_DATE	C_FIRSTNAME	C_SURNAME	OUTCOME
Father's Day Promo	15-JUN-2021 00:00	20-JUN-2021 00:00	Maddie	Smith	No Response
Mother's Day Promo	02-MAY-2021 00:00	09-MAY-2021 00:00	Yasmin	Charles	No Response
Happy education offer	10-JAN-2021 00:00	17-JAN-2021 00:00	Rita	Kim	Order Obtained
Happy education offer	10-JAN-2021 00:00	17-JAN-2021 00:00	Sophie	Young	Order Obtained
Happy education offer	25-DEC-2020 00:00	31-DEC-2020 00:00	Olivia	Owen	Order Obtained
Happy education offer	25-DEC-2020 00:00	31-DEC-2020 00:00	Justin	Phillips	Order Obtained
Happy education offer	25-DEC-2020 00:00	31-DEC-2020 00:00	Johnny	See	No Response
Black friday sales	27-NOV-2020 00:00	27-NOV-2020 00:00	Sara	Flynn	Order Obtained
Black friday sales	27-NOV-2020 00:00	27-NOV-2020 00:00	Andrew	Lee	No Response
Black friday sales	27-NOV-2020 00:00	27-NOV-2020 00:00	Brandon	Bennett	No Response

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.07 seconds Download

----------END OF ASSIGNMENT ----------