A picture containing computer, desk, table, keyboard

Description automatically generated

REPORT (ELECTRONIC STORE)

BY

MUHAMMAD IZZAT NAJMI BIN AZHAR 1716049

MOHAMAD IRWAN BIN ISHAK 1714023

NADZREEN AQIL BIN MOHD YUSOF 1715129

LECTURER: DR. ZAINATUL SHIMA ABDULLAH  
DATE: 7/8/2020

A close up of a logo

Description automatically generated

# Table of contents

[Table of contents 2](#_Toc47237538)

[Members Contribution 3](#_Toc47237539)

[Name 3](#_Toc47237540)

[Position 3](#_Toc47237541)

[Contribution 3](#_Toc47237542)

[Task 1 – Scenario 4](#_Toc47237543)

[Task 2 – ER Diagram 5](#_Toc47237544)

[Task 3 – Data Dictionary 6](#_Toc47237545)

[Task 4 – Create sql. 8](#_Toc47237547)

[Task 5 – Inserting/ Add sql. 14](#_Toc47237548)

[Task 6 – Creating query. 20](#_Toc47237549)

[Task 7 – Creating Procedure 23](#_Toc47237550)

[Task 8 – Creating Function 29](#_Toc47237551)

[Task 9 – Webpages 36](#_Toc47237552)

# Members Contribution

|  |  |  |
| --- | --- | --- |
| Name | Position | Contribution |
| MUHAMMAD IZZAT NAJMI BIN AZHAR | Leader | - create and design web pages (task 9),  - communicate with every member,  - create live webpages  - testing webpages if there are any bugs. |
| MOHAMAD IRWAN BIN ISHAK | Distributer | - Make report  - helping Nazreen with task 1-8  - distribute works accordingly  - testing webpages if there are any bugs. |
| NADZREEN AQIL BIN MOHD YUSOF | Writer | - Make task 1-8  - write and create scenario for every case  - give ideas for every task  - testing webpages if there are any bugs. |
| ***UMIR BIN NAZARUDDIN*** | ***Null*** | ***Null*** |

1. Introduction

# Task 1 – Scenario

The developer team had been asked by the electronic store’s owner to build a system to monitor his store’s sale. The electronic store is selling two types of device, phone, and laptop. We decided to differ the device’s ID into two which are starting with L for laptop and P for phone. All the details of the devices will be stored into the ‘Device Table’ to allow the administrator to monitor anything related to the device easily.

The administrator of the system could see the details about their customers, suppliers and workers stored in their own table, ‘USER2 Table’ , ‘Supplier Table’ and ‘Worker Table’. These three tables have their own primary key that allow others table to use it as the foreign key.

When the customers buy device(s) from the store, the details will be stored in the ‘Order2 Table’. This table keep the data of the transactions like date, IDs of user and worker involved, quantity of the purchase and the date of the transaction. This allows the administrator to keep track of the transaction effectively.

The ‘Price\_list2 Table’ is for the use of finance department. It stores the pricing of each devices in the store including their tax and the discount. The finance department only need to view this table to plan the store’s future financial plan.

# A close up of a logo Description automatically generatedTask 2 – ER Diagram

Figure 1: Er diagram for our company

# Task 3 – Data Dictionary

# 

|  |  |  |  |
| --- | --- | --- | --- |
| **WORKER** |  |  |  |
| Attribute name | data type | size | null |
| **worker\_id** | varchar2 | 5 | no(pk) |
| first\_name | varchar2 | 20 | no |
| last\_name | varchar2 | 20 | no |
| annual\_bonus | number | 5 | no |
| date\_hired | date |  | no |
| salary | number | 5 | no |

|  |  |  |  |
| --- | --- | --- | --- |
| **DEVICE** |  |  |  |
| Attribute name | data type | size | null |
| **device\_id** | varchar2 | 5 | no(pk) |
| *supplier\_id* | varchar2 | 5 | no |
| model\_name | varchar2 | 30 | no |
| operating\_system | varchar2 | 20 | Yes |
| stock(device) | number | 6 | Yes |

|  |  |  |  |
| --- | --- | --- | --- |
| **ORDER2** |  |  |  |
| Attribute name | data type | size | null |
| **order\_id** | varchar2 | 5 | no(pk) |
| *user\_id* | varchar2 | 5 | no |
| *device\_id* | varchar2 | 5 | no |
| *worker\_id* | varchar2 | 5 | no |
| quantity | number | 3 | no |
| date\_purchase | date |  | no |

|  |  |  |  |
| --- | --- | --- | --- |
| **SUPPLIER** |  |  |  |
| Attribute name | data type | size | null |
| **supplier\_id** | varchar2 | 5 | no(pk) |
| s\_name | varchar2 | 30 | no |
| s\_nophone | varchar2 | 15 | yes |
| s\_address | varchar2 | 50 | yes |

|  |  |  |  |
| --- | --- | --- | --- |
| **PRICE\_LIST2** |  |  |  |
| Attribute name | data type | size | null |
| **device\_id** | varchar2 | 5 | no(pk) |
| price | number | 6 | no |
| tax | number | 6 | no |
| discount | number | 3 | yes |

# Task 4 – Create sql.

DROP TABLE USER2 cascade constraints;

DROP TABLE WORKER cascade constraints;

DROP TABLE DEVICE cascade constraints;

DROP TABLE ORDER2 cascade constraints;

DROP TABLE SUPPLIER cascade constraints;

DROP TABLE PRICE\_LIST2 cascade constraints;

CREATE TABLE USER2

(USER\_ID varchar2(5),

FIRST\_NAME varchar2(20),

LAST\_NAME varchar2(20),

ADDRESS varchar2(50),

AGE number (3),

CONSTRAINT USER2\_USER\_ID\_PK PRIMARY KEY (USER\_ID));

A screenshot of a cell phone

Description automatically generated

Figure - User2 desc

CREATE TABLE WORKER

(WORKER\_ID varchar2(5),

FIRST\_NAME varchar2(20),

LAST\_NAME varchar2(20),

ANNUAL\_BONUS number (5),

DATE\_HIRED DATE,

SALARY number (5),

CONSTRAINT WORKER\_WORKER\_ID\_PK PRIMARY KEY (WORKER\_ID));

A screenshot of a cell phone

Description automatically generated

Figure - Worker desc

CREATE TABLE SUPPLIER

(SUPPLIER\_ID VARCHAR2(5),

S\_NAME VARCHAR2(30),

S\_NOPHONE VARCHAR2(15),

S\_ADDRESS VARCHAR2(50),

CONSTRAINT SUPPLIER\_SUPPLIER\_ID\_PK PRIMARY KEY (SUPPLIER\_ID));

A screenshot of a cell phone

Description automatically generated

Figure - Supplier desc

CREATE TABLE DEVICE

(DEVICE\_ID VARCHAR2(5),

SUPPLIER\_ID VARCHAR2(5),

MODEL\_NAME VARCHAR2(30),

OPERATING\_SYSTEM VARCHAR2(20),

STOCK NUMBER (6),

CONSTRAINT DEVICE\_DEVICE\_ID\_PK PRIMARY KEY (DEVICE\_ID),

CONSTRAINT DEVICE\_SUPPLIER\_ID\_FK FOREIGN KEY (SUPPLIER\_ID) REFERENCES SUPPLIER);

A screenshot of a cell phone

Description automatically generated

Figure - Device desc

CREATE TABLE ORDER2

(ORDER\_ID VARCHAR2(5),

USER\_ID VARCHAR2(5),

DEVICE\_ID VARCHAR2(5),

WORKER\_ID VARCHAR2(5),

QUANTITY NUMBER (3),

DATE\_PURCHASE DATE,

CONSTRAINT ORDER2\_ORDER\_ID\_PK PRIMARY KEY (ORDER\_ID),

CONSTRAINT ORDER2\_USER\_ID\_FK FOREIGN KEY (USER\_ID) REFERENCES USER2,

CONSTRAINT ORDER2\_WORKER\_ID\_FK FOREIGN KEY (WORKER\_ID) REFERENCES WORKER,

CONSTRAINT ORDER2\_DEVICE\_ID\_FK FOREIGN KEY (DEVICE\_ID) REFERENCES DEVICE);

A screenshot of a cell phone

Description automatically generated

Figure – Order desc

CREATE TABLE PRICE\_LIST2

(DEVICE\_ID VARCHAR2(5),

PRICE NUMBER (6),

TAX NUMBER (3,2),

DISCOUNT NUMBER (3,2),

CONSTRAINT PRICE\_LIST2\_DEVICE\_ID\_PK PRIMARY KEY (DEVICE\_ID));

A screenshot of a cell phone

Description automatically generated

Figure - price list desc

# Task 5 – Inserting/ Add sql.

INSERT INTO USER2 VALUES('U001', 'Ali', 'Ahmad', '3434 Mulberry St.',38);

INSERT INTO USER2 VALUES('U002', 'Abu', 'Sufian','334 Tailgate Ln',44);

INSERT INTO USER2 VALUES('U003', 'Siti', 'Rafidah','8814 W. Apple ',37);

INSERT INTO USER2 VALUES('U004', 'Tihani', 'chong','445 Oak St.',28);

INSERT INTO USER2 VALUES('U005', 'Lee', 'Wei','8898 Bighill Drive',27);

INSERT INTO USER2 VALUES('U006', 'Lin', 'Dan','298 Murphy School Rd',26);

INSERT INTO USER2 VALUES('U007', 'David', 'Becham','334 Pansie Hill Rd.',25);

INSERT INTO USER2 VALUES('U008', 'Ronaldo', 'Satria','456 E. Grand #14',50);

INSERT INTO USER2 VALUES('U009', 'Wei', 'Sinnin','6543 W. Parkview Ln.',40);

INSERT INTO USER2 VALUES('U010', 'Mannan', 'Kamui','6657 N. Allen',30);

A screenshot of a cell phone

Description automatically generated

Figure - Display all users

INSERT INTO WORKER VALUES('W001', 'Husnul', 'Tarmizi', 2000, TO\_DATE('11-JAN-11'),3800);

INSERT INTO WORKER VALUES('W002', 'Nadzreen', 'Aqil', 100, TO\_DATE('10-FEB-10'),4400);

INSERT INTO WORKER VALUES('W003', 'Irwan', 'Ishak',500, TO\_DATE('21-MARCH-12'),3700);

INSERT INTO WORKER VALUES('W004', 'Amir', 'Ng',10000, TO\_DATE('23-APRIL-09'),28000);

INSERT INTO WORKER VALUES('W005', 'Saliha', 'Salina',1200, TO\_DATE('11-MAY-11'),2700);

INSERT INTO WORKER VALUES('W006', 'Raju', 'Sink',900, TO\_DATE('29-JUNE-12'),2600);

INSERT INTO WORKER VALUES('W007', 'Amelin', 'Sofia',850, TO\_DATE('30-JULY-12'),2500);

INSERT INTO WORKER VALUES('W008', 'Daniel', 'Kriss',2500, TO\_DATE('31-AUGUST-09'),5000);

INSERT INTO WORKER VALUES('W009', 'Elizabeth', 'Tan',300, TO\_DATE('22-SEPTEMBER-12'),4000);

INSERT INTO WORKER VALUES('W010', 'Sabar', 'Soleh',980, TO\_DATE('24-OCTOBER-12'),3000);

A screenshot of a cell phone

Description automatically generated

Figure - Display all workers

INSERT INTO SUPPLIER VALUES('S001', 'SAMSUNG SDN BHD', '110-235-4403', 'South Korea');

INSERT INTO SUPPLIER VALUES('S002', 'XIAOMI', '882-923-6123', 'China');

INSERT INTO SUPPLIER VALUES('S003', 'LENOVO', '912-592-5478', 'China');

INSERT INTO SUPPLIER VALUES('S004', 'ASUS', '621-483-5392', 'Taiwan');

INSERT INTO SUPPLIER VALUES('S005', 'DELL', '530-324-9238', 'United States');

INSERT INTO SUPPLIER VALUES('S006', 'ONE PLUS', '540-012-1293', 'China');

INSERT INTO SUPPLIER VALUES('S007', 'HP', '701-059-6492', 'United State');

INSERT INTO SUPPLIER VALUES('S008', 'BLACKBERRY', '012-609-3204', 'Canada');

INSERT INTO SUPPLIER VALUES('S009', 'ACER', '163-604-3041', 'Taiwan');

INSERT INTO SUPPLIER VALUES('S010', 'APPLE SDN BHD', '865-453-8234', 'United States');

A screenshot of a cell phone

Description automatically generated

Figure - Display all suppliers

INSERT INTO DEVICE VALUES('P001', 'S010', 'IPHONE X', 'IOS14',3800);

INSERT INTO DEVICE VALUES('P002', 'S001', 'SAMSUNG S20', 'ANDROID 12',440);

INSERT INTO DEVICE VALUES('L001', 'S003', 'LENOVO IDEAPAD', 'WINDOWS 10',370);

INSERT INTO DEVICE VALUES('P003', 'S002', 'BLACKSHARK XIAOMI', 'MIUI 12',288);

INSERT INTO DEVICE VALUES('P004', 'S006', 'ONE PLUS 7T', 'OXYGEN OS 3',2703);

INSERT INTO DEVICE VALUES('L002', 'S005', 'DELL XPS 15', 'WINDOWS 8',2655);

INSERT INTO DEVICE VALUES('L003', 'S009', 'ACER DUO 12', 'WINDOWS XP',251);

INSERT INTO DEVICE VALUES('P005', 'S010', 'APPLE XS', 'IOS14',5023);

INSERT INTO DEVICE VALUES('L004', 'S007', 'ASUS ROG 15 ELITE', 'WINDOWS 10 PRO',5500);

INSERT INTO DEVICE VALUES('P006', 'S010', 'IPHONE 12', 'IOS14',30000);

A screenshot of a cell phone

Description automatically generated

Figure - Display all devices

INSERT INTO ORDER2 VALUES('O001', 'U001', 'L001','W010', 3,TO\_DATE('11-FEB-13'));

INSERT INTO ORDER2 VALUES('O002', 'U005', 'P003','W009', 1,TO\_DATE('10-MARCH-13'));

INSERT INTO ORDER2 VALUES('O003', 'U001', 'P004','W008', 2,TO\_DATE('04-JUNE-13'));

INSERT INTO ORDER2 VALUES('O004', 'U007', 'P005','W007', 4,TO\_DATE('08-JULY-13'));

INSERT INTO ORDER2 VALUES('O005', 'U001', 'P001','W006', 1,TO\_DATE('19-AUGUST-13'));

INSERT INTO ORDER2 VALUES('O006', 'U008', 'P001','W005', 2,TO\_DATE('23-SEPTEMBER-13'));

INSERT INTO ORDER2 VALUES('O007', 'U009', 'L003','W004', 1,TO\_DATE('28-OCT-13'));

INSERT INTO ORDER2 VALUES('O008', 'U010', 'P006','W004', 8,TO\_DATE('13-NOV-13'));

INSERT INTO ORDER2 VALUES('O009', 'U002', 'P001','W003', 3,TO\_DATE('01-DEC-13'));

INSERT INTO ORDER2 VALUES('O010', 'U003', 'P002','W002', 4,TO\_DATE('10-AUGUST-13'));

A screenshot of a cell phone

Description automatically generated

Figure - Display all orders

INSERT INTO PRICE\_LIST2 VALUES('P001', 2500, 0.16, 0.02);

INSERT INTO PRICE\_LIST2 VALUES('P002', 3000, 0.10, 0.10);

INSERT INTO PRICE\_LIST2 VALUES('L001', 1500, 0.15, 0.20);

INSERT INTO PRICE\_LIST2 VALUES('P003', 2900, 0.13, 0.09);

INSERT INTO PRICE\_LIST2 VALUES('P004', 3500, 0.17, 0.05);

INSERT INTO PRICE\_LIST2 VALUES('L002', 6900, 0.16, 0.01);

INSERT INTO PRICE\_LIST2 VALUES('L003', 3100, 0.11, 0.25);

INSERT INTO PRICE\_LIST2 VALUES('P005', 3500, 0.16, 0.01);

INSERT INTO PRICE\_LIST2 VALUES('L004', 5500, 0.10, 0.20);

INSERT INTO PRICE\_LIST2 VALUES('P006', 5500, 0.16, 0.01);

A screenshot of a cell phone

Description automatically generated

Figure - Display all prices

# Task 6 – Creating query.

/// The administrator want to see all details of the device sold in the store

1) select \* from device;

A screenshot of a cell phone

Description automatically generated

Figure – Display all devices

//The administrator want to add certain details of new device into the system

2) insert into device(device\_id,supplier\_id,model\_name) values('L005','S007','HP ELITE');

A screenshot of a social media post

Description automatically generated

Figure - Add new row

//The administrator want to delete details of the device with device ID 'P001'

3) alter table device drop constraint device\_supplier\_id\_fk;

alter table order2 drop constraint order2\_device\_id\_fk;

delete from device where device\_id = 'P001';

A screenshot of a cell phone

Description automatically generated

Figure - Altered table

//The administrator want to see the devices name and their supplier's ID, name and adress

4) select model\_name, supplier\_id, s\_name,s\_address

from device natural join supplier

order by device\_id;

A screenshot of a cell phone

Description automatically generated

Figure - Use natural join

//The administrator want to see devices' model name, their price and discount

5) select model\_name, price, discount

from device natural join price\_list2;

A screenshot of a cell phone

Description automatically generated

Figure - Display model name and price

//The administrator want to see the customer's name, details of the transaction and the staff's name when the user ID is U008, device ID is P006, and the worker ID is W005.

6) select a.first\_name, b.order\_id, b.device\_id, b.date\_purchase, c.first\_name, c.address

from worker a, order2 b, user2 c

where b.device\_id = 'P006' and c.user\_id = 'U008' and a.worker\_id = 'W005'A screenshot of a cell phone

Description automatically generated

Figure - Use triple join

# Task 7 – Creating Procedure

--The administrator want to use updatePrice procedure to update the devices' price. The new price of the devices will be determine based on their current value.

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

set serveroutput on

set verify off

CREATE OR REPLACE PROCEDURE updatePrice(i\_device\_id IN price\_list2.device\_id%TYPE)

IS

CURSOR up\_device(p\_device\_id price\_list2.device\_id %TYPE)

IS

SELECT device\_id, price,tax,DISCOUNT

FROM price\_list2

WHERE device\_id = p\_device\_id;

v\_device\_id price\_list2.device\_id%TYPE;

v\_price price\_list2.price%TYPE;

v\_tax price\_list2.tax%TYPE;

v\_discount price\_list2.discount%TYPE;

v\_newprice price\_list2.price%TYPE;

v\_rebate price\_list2.price%TYPE;

v\_newpricerebate price\_list2.price%TYPE;

BEGIN

OPEN up\_device(i\_device\_id);

LOOP

FETCH up\_device

INTO v\_device\_id, v\_price,v\_tax,v\_discount;

exit when up\_device%NOTFOUND;

IF v\_price <= 3000 THEN

v\_newprice := (v\_price)+(v\_price\*v\_tax)\*(v\_discount);

dbms\_output.put\_line('Old price is:' || v\_price);

dbms\_output.put\_line('The new price is: '||v\_newprice);

elsif v\_price >= 5500 THEN

v\_rebate := 500;

v\_newpricerebate := (v\_price+(v\_price\*v\_tax)) - v\_rebate;

dbms\_output.put\_line('Old price is:' || v\_price);

dbms\_output.put\_line('The new price is: '|| v\_newpricerebate );

ELSE

dbms\_output.put\_line('second');

END IF;

END LOOP;

CLOSE up\_device;

END;

--###########

BEGIN

updatePrice('P006');

END;

BEGIN

updatePrice('L001');

END;

A screenshot of a cell phone

Description automatically generated

Figure - Procedure update price

--##############################2

--The administrator want to use addDevice procedure to add details of the new devices into the system. The procedure will print the error if the device ID is in invalid format.

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

set serveroutput on

set verify off

create or replace procedure addDevice(

p\_deviceid device.device\_id%type,

p\_modelname device.model\_name%type,

p\_operatingsystem device.operating\_system%type,

p\_stock device.stock%type)

as

ex\_device\_id exception;

pragma exception\_init(ex\_device\_id, -12899);

begin

insert into device(device\_id,model\_name,operating\_system, stock)

values(p\_deviceid,p\_modelname,p\_operatingsystem,p\_stock);

if SQL%NOTFOUND THEN

raise ex\_device\_id;

end if;

exception

when ex\_device\_id then

dbms\_output.put\_line('The device ID is too big!!! PLease retry again (DEVICE ID ONLY UP TO 5 CHARACTERS!!!).');

end addDevice;

accept p\_deviceid PROMPT 'Enter device id:'

accept p\_modelname PROMPT 'Enter model name:'

accept p\_operatingsystem PROMPT 'Enter operating system:'

accept p\_stock PROMPT 'Enter stock:'

begin

addDevice('&p\_deviceid','&p\_modelname','&p\_operatingsystem','&p\_stock');

if SQL%FOUND then

dbms\_output.put\_line('New row is added!!');

end if;

end;

select \* from device;

A screenshot of a cell phone

Description automatically generated

Figure - Device before

A screenshot of a cell phone

Description automatically generated

Figure - Device After

# Task 8 – Creating Function

--Senario : The administrator want to use function device\_discount to calculate the devices' sale price. The sale price will be calculate by the base price minus its discount.

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

Set serveroutput on

Set verify off

create or replace function device\_discount(device\_id IN varchar2)

return varchar2

is

discount number;

begin

if device\_id = 'P006' THEN discount := 0.16;

elsif device\_id = 'L002' then discount := 0.20;

else discount := 0.1;

end if;

return(discount);

end;

accept device\_id prompt 'Enter The device id: '

declare

v\_total number(6,2);

v\_price price\_list2.price%type;

v\_deviceid price\_list2.device\_id%type :='&device\_id';

begin

select price into v\_price

from price\_list2

where device\_id = '&device\_id';

v\_total := v\_price-(v\_price\*device\_discount('&device\_id'));

dbms\_output.put\_line('The device ' || v\_deviceid ||' before being discount is '||v\_price);

dbms\_output.put\_line('The device ' || v\_deviceid ||' price after being discount is '||v\_total);

end;

A screenshot of a cell phone

Description automatically generated

Figure - Device discount

------------------------------------------------------------second function

set verify off

set serveroutput on

create or replace function gsttax(v\_datepurchase in date)

return varchar2

is

settax varchar(50);

begin

if v\_datepurchase > '15-JULY-2013' then settax := 'Your purchase will add with tax which is 6%';

else

settax := 'Your purchase does not have any tax';

end if;

return (settax);

end;

accept order\_id prompt 'Enter the order id: '

declare

v\_modelname device.model\_name%type;

v\_price price\_list2.price%type;

v\_datepurchase order2.date\_purchase%type;

v\_settax varchar2(60);

begin

select model\_name, price, date\_purchase

into v\_modelname,v\_price,v\_datepurchase

from price\_list2 natural join device natural join order2

where order\_id = '&order\_id';

v\_settax := gsttax(v\_datepurchase);

dbms\_output.put\_line('The status for your devices is: ' || v\_settax);

end;

A screenshot of a cell phone

Description automatically generated

Figure - Tax

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

The administrators want to use function worker id to generate the official email for the workers. The email is created based on their worker ID and their name.

--\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

set serveroutput on

set verify off

accept userid prompt 'Enter the user id:'

declare

type my\_refcur is ref cursor return user2%rowtype;

cur\_var my\_refcur;

record\_var user2%rowtype;

generateEmail varchar2(50);

v\_userid user2.user\_id%type := '&userid';

ex\_userid exception;

pragma exception\_init(ex\_userid, -06502);

begin

open cur\_var for

select \* from user2

where user\_id = v\_userid;

loop

fetch cur\_var into record\_var;

if record\_var.user\_id is null then

dbms\_output.put\_line('Please re-type your user id');

end if;

exit when cur\_var%notfound;

generateEmail := createEmail(record\_var.user\_id);

dbms\_output.put\_line(generateEmail);

-- raise ex\_userid;

end loop;

close cur\_var;

-- exception

-- when ex\_userid then

-- dbms\_output.put\_line('The user id is too big');

end;

create or replace function createEmail(f\_userid in user2.user\_id%type)

return varchar2

is

text varchar2(100);

begin

for x in (select user\_id, first\_name,last\_name from user2 where user\_id = f\_userid)

LOOP

text := 'The email address is '|| x.user\_id || '.'|| x.first\_name || x.last\_name|| '@gmail.com';

end loop;

return text;

end;

A screenshot of a cell phone

Description automatically generated

Figure - creating Email address

2) Appendix

# Task 9 – Webpages

A screenshot of a cell phone

Description automatically generated

Figure – Homepages

A screenshot of a cell phone

Description automatically generated

Figure – Users page

A screenshot of a social media post

Description automatically generated

Figure – User details

*A screenshot of a cell phone

Description automatically generated*

Figure - Worker page

A screenshot of a social media post

Description automatically generated

Figure - Workers details

A screenshot of a social media post

Description automatically generated

Figure - About Us

A screenshot of a cell phone

Description automatically generated

Figure - Laptop page

A screenshot of a cell phone

Description automatically generated

Figure - Laptop details

A screenshot of a cell phone

Description automatically generated

Figure - Laptop update page

A screenshot of a cell phone

Description automatically generated

Figure - Laptop after update

A screenshot of a cell phone

Description automatically generated

Figure - Laptop add page

A screenshot of a cell phone

Description automatically generated

Figure - Laptop after add page

A screenshot of a cell phone

Description automatically generated

Figure - Mobile page

A screenshot of a cell phone

Description automatically generated

Figure - Mobile details

A screenshot of a cell phone

Description automatically generated

Figure - Mobile update page

A screenshot of a cell phone

Description automatically generated

Figure - Mobile after update

A screenshot of a cell phone

Description automatically generated

Figure - Mobile add page

A screenshot of a cell phone

Description automatically generated

Figure – New mobile added!