

---

# Advance Database

---

## Assignment 1

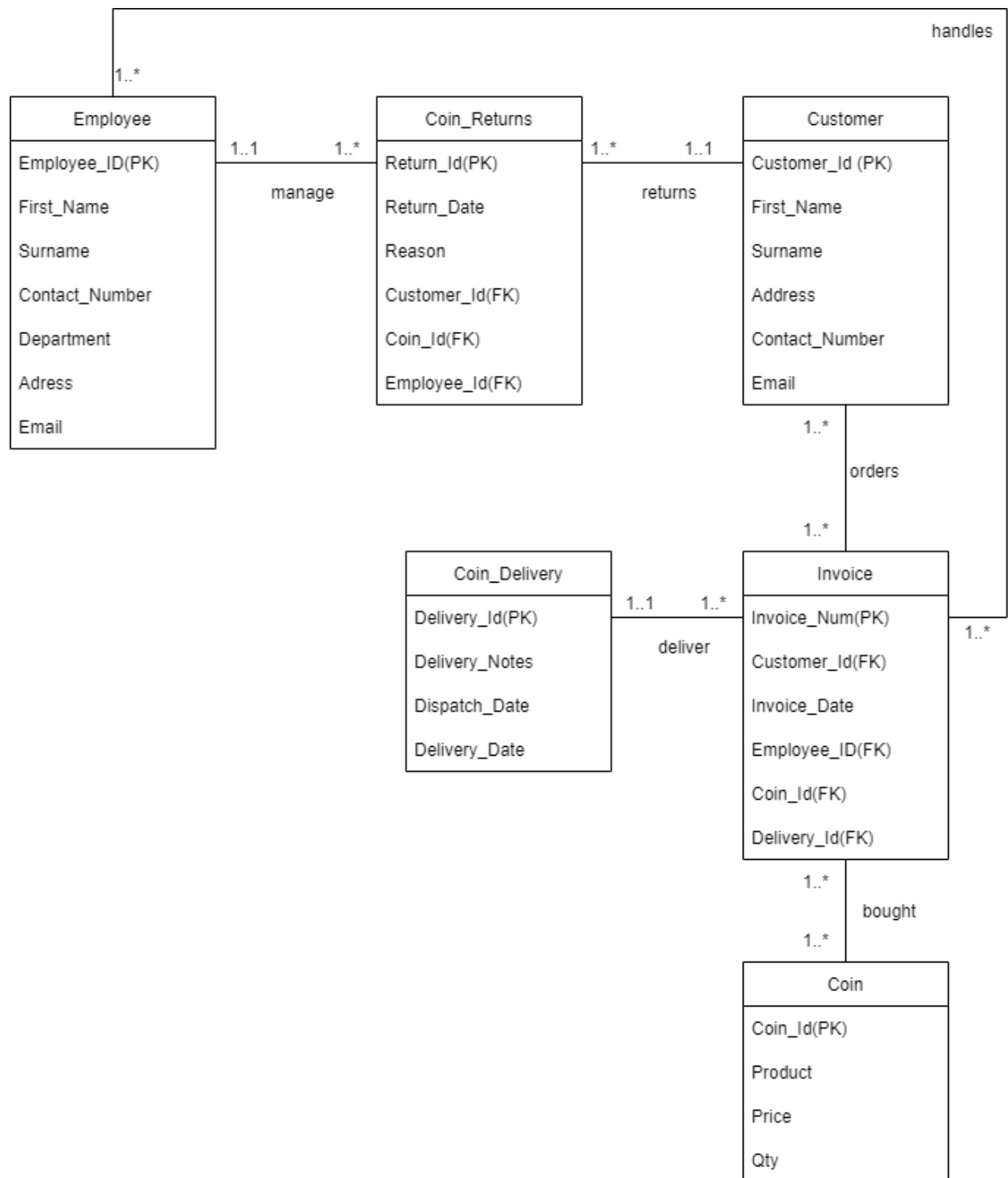
---

St10127198  
Baloyi

Ntsako

---

# Question 1



## Question 2

Q2.1 errors corrected and highlighted with red

set serveroutput on

declare

cust customer.first\_name%Type;

prod product.product%Type;

cursor info is

select cust.first\_name || ', ' || cust.surname CUSTOMER, p.product

from customer cust, billing b, product\_billing pb, product p

where cust.customer\_id = b.customer\_id

where b.bill\_id = pb.bill\_id

and p.product\_id = pb.product\_id

and p.price > 10000

order by p.price;

begin

for rec in info

loop

cust:= rec.CUSTOMER;

prod:= rec.product;

dbms\_output.put\_line('CUSTOMER: ' || cust || chr(13) || 'PRODUCT: ' || prod);

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

end loop;

end;

### Question 3

```
CREATE TABLE CUSTOMER(  
    CUSTOMER_ID NUMBER(5) NOT NULL PRIMARY KEY,  
    FIRST_NAME VARCHAR(50),  
    SURNAME VARCHAR(50),  
    ADDRESS VARCHAR(50),  
    CONTACT_NUMBER NUMBER(10),  
    EMAIL VARCHAR(50)  
);
```

```
CREATE TABLE EMPLOYEE(  
    EMPLOYEE_ID VARCHAR(50) NOT NULL PRIMARY KEY,  
    FIRST_NAME VARCHAR(50),  
    SURNAME VARCHAR(50),  
    CONTACT_NUMBER NUMBER(10),  
    DEPARTMENT VARCHAR(50),  
    ADDRESS VARCHAR(50),  
    EMAIL VARCHAR(50)  
);
```

```
CREATE TABLE COIN_DELIVERY(  
    DELIVERY_ID NUMBER(5) NOT NULL PRIMARY KEY,  
    DELIVERY_NOTES VARCHAR(50),  
    DISPATCH_DATE DATE,  
    DELIVERY_DATE DATE  
);
```

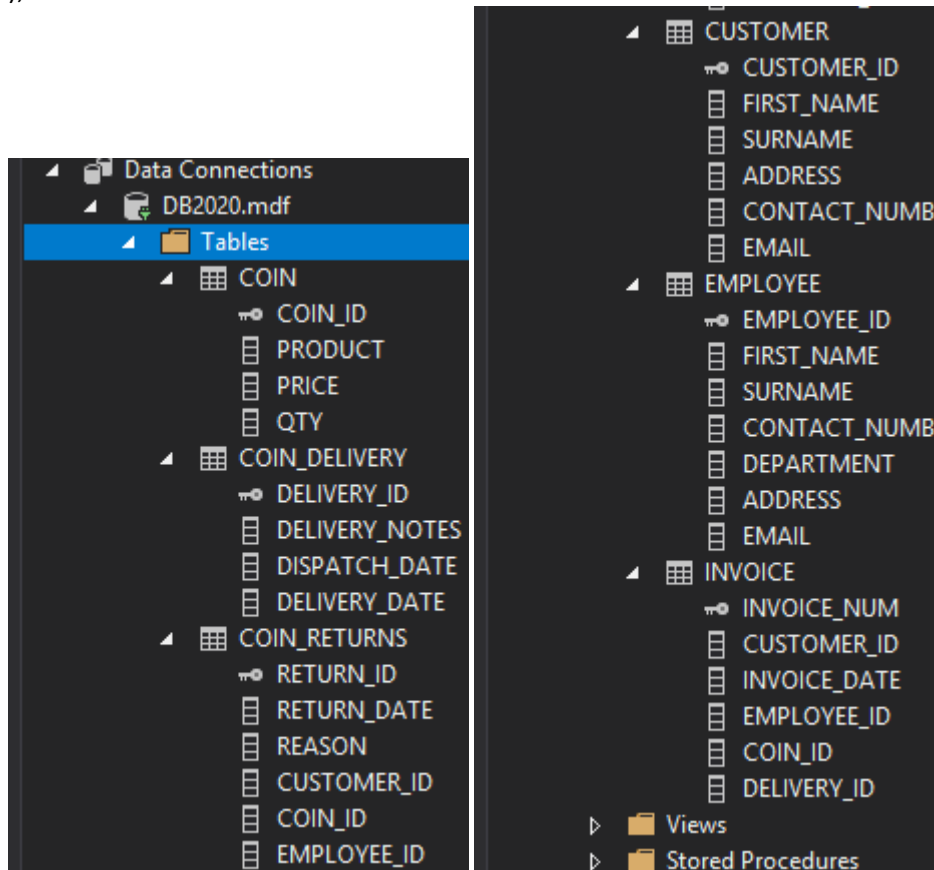
```
CREATE TABLE COIN(  
    COIN_ID NUMBER(5) NOT NULL PRIMARY KEY,  
    PRODUCT VARCHAR(50),  
    PRICE VARCHAR(50),  
    QTY INT  
);
```

```
CREATE TABLE COIN_RETURNS(  
    RETURN_ID VARCHAR(10) NOT NULL PRIMARY KEY,  
    RETURN_DATE DATE,  
    REASON VARCHAR(50),  
    CUSTOMER_ID NUMBER(5),  
    COIN_ID NUMBER(5),  
    EMPLOYEE_ID VARCHAR(50),  
    FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID),  
    FOREIGN KEY (COIN_ID) REFERENCES COIN(COIN_ID),  
    FOREIGN KEY (EMPLOYEE_ID) REFERENCES EMPLOYEE(EMPLOYEE_ID)  
);
```

```

CREATE TABLE INVOICE(
    INVOICE_NUM NUMBER(5) NOT NULL PRIMARY KEY,
    CUSTOMER_ID NUMBER(5) NOT NULL,
    INVOICE_DATE VARCHAR(50),
    EMPLOYEE_ID VARCHAR(50) NOT NULL,
    COIN_ID NUMBER(5) NOT NULL,
    DELIVERY_ID NUMBER(5) NOT NULL,
    FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID),
    FOREIGN KEY (COIN_ID) REFERENCES COIN(COIN_ID),
    FOREIGN KEY (EMPLOYEE_ID) REFERENCES EMPLOYEE(EMPLOYEE_ID),
    FOREIGN KEY (DELIVERY_ID) REFERENCES COIN_DELIVERY(DELIVERY_ID)
);

```



```

INSERT INTO Customer VALUES( 11011, 'Mat', 'Smith', '18 water RD', 0877277521, 'msmith@isat.com');
INSERT INTO Customer VALUES( 11012, 'Julian', 'Hendricks', '22 water RD', 0863257857, 'jh@mcom.co.za');
INSERT INTO Customer VALUES( 11013, 'Sam', 'Clark', '101 summer lane', 0834567891, 'sclark@mcom.co.za');
INSERT INTO Customer VALUES( 11014, 'Kevin', 'Jones', '55 Mountain way', 0612547895, 'kj@isat.co.za');
INSERT INTO Customer VALUES( 11015, 'Lucy', 'Williams', '5 Main', 0827238521, 'lw@mcal.co.za');

```

	CUSTOMER_ID	FIRST_NAME	SURNAME	ADDRESS	CONTACT_NUMBER	EMAIL
1	11011	Mat	Smith	18 water RD	877277521	msmith@isat.com
2	11012	Julian	Hendricks	22 water RD	863257857	jh@mcom.co.za
3	11013	Sam	Clark	101 summer lane	834567891	sclark@mcom.co.za
4	11014	Kevin	Jones	55 Mountain way	612547895	kj@isat.co.za
5	11015	Lucy	Williams	5 Main	827238521	lw@mcal.co.za

INSERT INTO Employee VALUES( 'emp101', 'Xander', 'Davis',0877277521,'sales', '18 water RD', 'msmith@isat.com');  
 INSERT INTO Employee VALUES( 'emp102', 'Steven', 'Marks',0877277521,'marketing', '18 water RD', 'msmith@isat.com');  
 INSERT INTO Employee VALUES( 'emp103', 'Jessica', 'Andrews',0877277521,'sales', '18 water RD', 'msmith@isat.com');  
 INSERT INTO Employee VALUES( 'emp104', 'Wayne', 'Dryer',0877277521,'sales', '18 water RD', 'msmith@isat.com');  
 INSERT INTO Employee VALUES( 'emp105', 'Xolani', 'Sansom', 0877277521,'manager', '18 water RD', 'msmith@isat.com');

	EMPLOYEE_ID	FIRST_NAME	SURNAME	CONTACT_NUMBER	DEPARTMENT	ADDRESS	EMAIL
1	emp101	Xander	Davis	877277521	sales	18 water RD	msmith@isat.com
2	emp102	Steven	Marks	877277521	marketing	18 water RD	msmith@isat.com
3	emp103	Jessica	Andrews	877277521	sales	18 water RD	msmith@isat.com
4	emp104	Wayne	Dryer	877277521	sales	18 water RD	msmith@isat.com
5	emp105	Xolani	Sansom	877277521	manager	18 water RD	msmith@isat.com

INSERT INTO COIN\_DELIVERY VALUES(511, 'Double packaging requested', '10 may 2021', '15 may 2021');  
 INSERT INTO COIN\_DELIVERY VALUES(512, 'Delivery to work address', '12 may 2021', '15 may 2021');  
 INSERT INTO COIN\_DELIVERY VALUES(513, 'Signature required', '12 may 2021', '17 may 2021');  
 INSERT INTO COIN\_DELIVERY VALUES(514, 'No notes', '12 may 2021', '15 may 2021');  
 INSERT INTO COIN\_DELIVERY VALUES(515, 'Birthday present wrapping required', '18 may 2021', '19 may 2021');  
 INSERT INTO COIN\_DELIVERY VALUES(516, 'Delivery to work place', '20 may 2021', '25 may 2021');  
 INSERT INTO COIN\_DELIVERY VALUES(517, 'Double packaging requested', '28 may 2021', '27 may 2021');

	DELIVERY_ID	DELIVERY_NOTES	DISPATCH_DATE	DELIVERY_DATE
1	511	Double packaging requested	2021-05-10	2021-05-15
2	512	Delivery to work address	2021-05-12	2021-05-15
3	513	Signature required	2021-05-12	2021-05-17
4	514	No notes	2021-05-12	2021-05-15
5	515	Birthday present wrapping required	2021-05-18	2021-05-19
6	516	Delivery to work place	2021-05-20	2021-05-25
7	517	Double packaging requested	2021-05-28	2021-05-27

INSERT INTO COIN VALUES(7111, '1oz Gold Kruger Rand', 'R 5 999', 10);  
 INSERT INTO COIN VALUES(7112, '1oz Gold Kruger Rand', 'R 12 999', 8);  
 INSERT INTO COIN VALUES(7113, 'Gold Big 5 Uncirculated', 'R 15 999', 8);  
 INSERT INTO COIN VALUES(7114, 'Silver Big 5 Pack', 'R 7 999', 5);  
 INSERT INTO COIN VALUES(7115, '1oz Gold Palaeontology', 'R 11 999', 15);  
 INSERT INTO COIN VALUES(7116, '1oz Silver Palaeontology', 'R 7 999', 12);

	COIN_ID	PRODUCT	PRICE	QTY
1	7111	1oz Gold Kruger Rand	R 5 999	10
2	7112	1oz Gold Kruger Rand	R 12 999	8
3	7113	Gold Big 5 Uncirculated	R 15 999	8
4	7114	Silver Big 5 Pack	R 7 999	5
5	7115	1oz Gold Palaeontology	R 11 999	15
6	7116	1oz Silver Palaeontology	R 7 999	12

```

INSERT INTO INVOICE VALUES(8111, 11011, '15 May 2021', 'emp103', 7111, 511);
INSERT INTO INVOICE VALUES(8112, 11013, '15 May 2021', 'emp101', 7116, 512);
INSERT INTO INVOICE VALUES(8113, 11012, '17 May 2021', 'emp101', 7112, 513);
INSERT INTO INVOICE VALUES(8114, 11015, '17 May 2021', 'emp102', 7111, 514);
INSERT INTO INVOICE VALUES(8115, 11011, '17 May 2021', 'emp102', 7115, 515);
INSERT INTO INVOICE VALUES(8116, 11015, '18 May 2021', 'emp103', 7115, 516);
INSERT INTO INVOICE VALUES(8117, 11012, '19 May 2021', 'emp105', 7112, 517);
INSERT INTO INVOICE VALUES(8118, 11013, '19 May 2021', 'emp105', 7112, 517);

```

	INVOICE_NUM	CUSTOMER_ID	INVOICE_DATE	EMPLOYEE_ID	COIN_ID	DELIVERY_ID
1	8111	11011	15 May 2021	emp103	7111	511
2	8112	11013	15 May 2021	emp101	7116	512
3	8113	11012	17 May 2021	emp101	7112	513
4	8114	11015	17 May 2021	emp102	7111	514
5	8115	11011	17 May 2021	emp102	7115	515
6	8116	11015	18 May 2021	emp103	7115	516
7	8117	11012	19 May 2021	emp105	7112	517
8	8118	11013	19 May 2021	emp105	7112	517

```

INSERT INTO COIN_RETURNS VALUES('ret001','25 May 2021','Customer not satisfied with product', '11011', '7116', 'emp101');
INSERT INTO COIN_RETURNS VALUES('ret002','25May2021','Product missing part', '11013', '7114', 'emp103');

```

	RETURN_ID	RETURN_DATE	REASON	CUSTOMER_ID	COIN_ID	EMPLOYEE_ID
1	ret001	2021-05-25	Customer not satisfied with product	11011	7116	emp101
2	ret002	2021-05-25	Product missing part	11013	7114	emp103

#### Question 4

```
SELECT CUSTOMER.FIRST_NAME||','|| CUSTOMER.SURNAME AS CUSTOMER, EMPLOYEE.EMPLOYEE_ID,  
COIN_DELIVERY.DELIVERY_NOTES AS DESCRIPTION, COIN.PRODUCT AS COIN, INVOICE.INVOICE_NUM,  
INVOICE.INVOICE_DATE  
FROM CUSTOMER  
JOIN INVOICE ON INVOICE.CUSTOMER_ID = CUSTOMER.CUSTOMER_ID  
JOIN COIN ON COIN.COIN_ID = INVOICE.COIN_ID  
JOIN COIN_DELIVERY ON COIN_DELIVERY.DELIVERY_ID = INVOICE.DELIVERY_ID  
JOIN EMPLOYEE ON EMPLOYEE.EMPLOYEE_ID = INVOICE.EMPLOYEE_ID  
WHERE INVOICE.INVOICE_DATE >= TO_DATE ('18 may 2021','DD-MON-RR');
```

CUSTOMER	EMPLOYEE_ID	DESCRIPTION	COIN	INVOICE_NUM	INVOICE_DATE
1 Lucy,Williams	emp103	Delivery to work place	1oz Gold Palaeontology	8116	18 May 2021
2 Julian,Hendricks	emp105	Double packaging requested	1oz Gold Kruger Rand	8117	19 May 2021
3 Sam,Clark	emp105	Double packaging requested	1oz Gold Kruger Rand	8118	19 May 2021



### Question 5

CREATE OR REPLACE VIEW DATAVIEW

AS

SELECT EMPLOYEE.EMPLOYEE\_ID, EMPLOYEE.FIRST\_NAME, EMPLOYEE.SURNAME, 'R' ||

SUM(COIN.PRICE) AS COINPRICE, 'R' || SUM(COIN.PRICE \* 0.1) AS COMMISION

FROM EMPLOYEE, COIN, INVOICE

WHERE EMPLOYEE.EMPLOYEE\_ID = INVOICE.EMPLOYEE\_ID

AND COIN.COIN\_ID = INVOICE.COIN\_ID

GROUP BY EMPLOYEE.EMPLOYEE\_ID, EMPLOYEE.FIRST\_NAME, EMPLOYEE.SURNAME

ORDER BY EMPLOYEE.SURNAME DESC;



	EMPLOYEE_ID	FIRST_NAME	SURNAME	COIN_PRICE	COMMISION
1	empl05	Xolain	Samson	R25998	R2599.8
2	empl02	Steven	Marks	R17998	R1799.8
3	empl01	Xander	Davis	R20998	R2099.8
4	empl03	Jessica	Andrews	R17998	R1799.8

## Question 6

declare

```
customer_name varchar(255);
coin_product varchar(255);
coin_price number;
delivery varchar(255);
```

cursor myC is

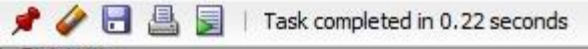
```
SELECT CUSTOMER.FIRST_NAME, COIN.PRODUCT, COIN.PRICE, COIN_DELIVERY.DELIVERY_NOTES
FROM INVOICE
INNER JOIN COIN_DELIVERY
ON COIN_DELIVERY.DELIVERY_ID = INVOICE.DELIVERY_ID
INNER JOIN COIN
ON COIN.COIN_ID = INVOICE.COIN_ID
INNER JOIN CUSTOMER
ON CUSTOMER.CUSTOMER_ID = INVOICE.CUSTOMER_ID
WHERE COIN.PRICE <= 8000;
```

begin

```
open myC;
loop
    fetch myC into customer_name, coin_product, coin_price, delivery;
    exit when myC %notfound or myC %rowcount > 5;
    dbms_output.put_line('CUSTOMER:' || customer_name);
    dbms_output.put_line('COIN:' || coin_product);
    dbms_output.put_line('PRICE: R' || coin_price);
    dbms_output.put_line('NOTES:' || delivery);
    dbms_output.put_line('-----');
end loop;
```

close myC;

end;



Task completed in 0.22 seconds

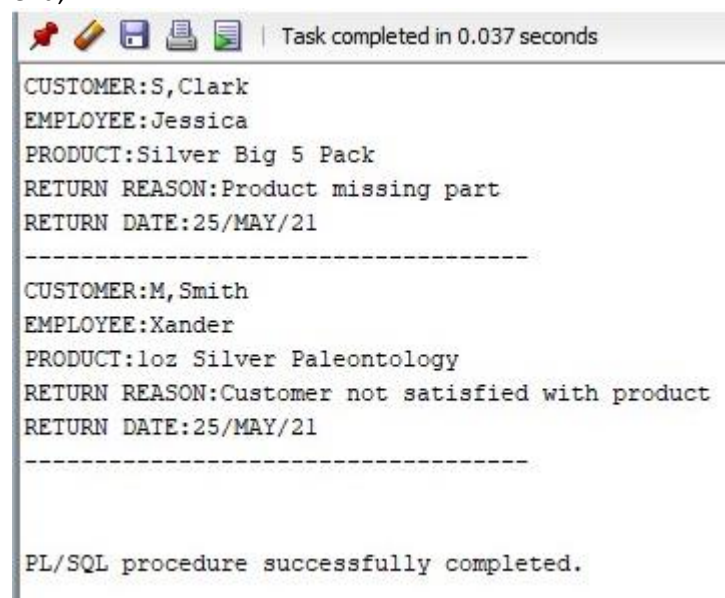
```

Action:
CUSTOMER:Mat,Smith
COIN:1oz Gold Kruger Rand
PRICE: R5999
NOTES:Double packaging requested
-----
CUSTOMER:Sam,Clark
COIN:1oz Silver Paleontology
PRICE: R7999
NOTES:Delivery to work address
-----
CUSTOMER:Lucy,Williams
COIN:1oz Gold Kruger Rand
PRICE: R5999
NOTES:No notes
-----
```

## Question 7

```
declare
  customer_name varchar(255);
  employee_name varchar(255);
  employee_surname varchar(255);
  coin_name varchar(255);
  reason varchar(255);
  date date;
  cursor myC is
    SELECT SUBSTR(CUSTOMER.FIRST_NAME, 1, 1) || ',' || CUSTOMER.SURNAME AS CUSTOMER ,
    EMPLOYEE.FIRST_NAME, COIN.PRODUCT, COIN_RETURNS.REASON, COIN_RETURNS.RETURN_DATE
    FROM COIN_RETURNS
    INNER JOIN COIN
    ON COIN_RETURNS.COIN_ID = COIN.COIN_ID
    INNER JOIN CUSTOMER
    ON CUSTOMER.CUSTOMER_ID = COIN_RETURNS.CUSTOMER_ID
    INNER JOIN EMPLOYEE
    ON EMPLOYEE.EMPLOYEE_ID = COIN_RETURNS.EMPLOYEE_ID;
begin
  open myC;
  loop
    fetch myC into customer_name, employee_name, employee_surname, coin_name, reason, date;
    exit when myC %notfound or myC %rowcount > 5;
    dbms_output.put_line('CUSTOMER:' || customer_name || ',' || employee_surname);
    dbms_output.put_line('EMPLOYEE:' || employee_name || ',' || employee_surname);
    dbms_output.put_line('PRODUCT: R' || coin_name);
    dbms_output.put_line('RETURN REASON:' || reason);
    dbms_output.put_line('RETURN DATE:' || date);
    dbms_output.put_line('-----');
  end loop;

  close myC;
end;
```



```
Task completed in 0.037 seconds

CUSTOMER:S,Clark
EMPLOYEE:Jessica
PRODUCT:Silver Big 5 Pack
RETURN REASON:Product missing part
RETURN DATE:25/MAY/21
-----
CUSTOMER:M,Smith
EMPLOYEE:Xander
PRODUCT:Ioz Silver Paleontology
RETURN REASON:Customer not satisfied with product
RETURN DATE:25/MAY/21
-----

PL/SQL procedure successfully completed.
```

## Question 8

declare

```
invoice_num varchar(255);  
coin_name varchar(255);  
price number;  
theNote varchar(255);
```

cursor myC is

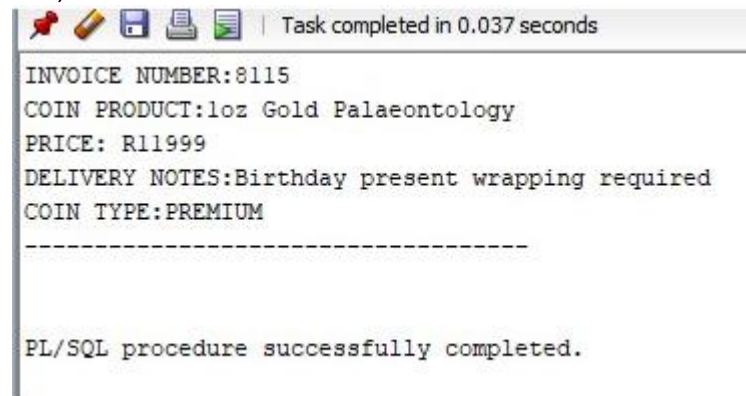
```
SELECT invoice.invoice_num, coin.product, coin.price, COIN_DELIVERY.delivery_notes  
FROM COIN_DELIVERY  
INNER JOIN INVOICE  
ON coin_delivery.delivery_id = invoice.delivery_id  
INNER JOIN COIN  
ON coin.coin_id = invoice.coin_id;
```

begin

```
open myC;  
loop  
    fetch myC into invoice_num, coin_name, price, theNote;  
    exit when myC %notfound or myC %rowcount > 5;  
    dbms_output.put_line('INVOICE NUMBER:' || invoice_num);  
    dbms_output.put_line('COIN NAME:' || coin_name);  
    dbms_output.put_line('PRICE: R' || price);  
    dbms_output.put_line('NOTES:' || theNote);  
    dbms_output.put_line('COIN TYPE:PREMIUM');  
    dbms_output.put_line('-----');  
end loop;
```

close myC;

end;



The screenshot shows a PL/SQL execution window with a toolbar at the top containing icons for a pin, a pencil, a save icon, a print icon, and a list icon. The status bar indicates "Task completed in 0.037 seconds". The output area displays the following text:

```
INVOICE NUMBER:8115  
COIN PRODUCT:1oz Gold Palaeontology  
PRICE: R11999  
DELIVERY NOTES:Birthday present wrapping required  
COIN TYPE:PREMIUM  
-----  
  
PL/SQL procedure successfully completed.
```

## Question 9

declare

```
customer_name varchar(255);
customer_surname varchar(255);
coin_id varchar(255);
coin_price number;
coin_discount number;
coin_del_date date;
invoice_date date;
```

cursor cursor1 is

```
SELECT customer.first_name, customer.surname, coin.coin_id, COIN.price, (COIN.price * 0.75),
coin_delivery.delivery_date, invoice.invoice_date
FROM INVOICE
INNER JOIN CUSTOMER
ON invoice.customer_id = customer.customer_id
INNER JOIN COIN_DELIVERY
ON coin_delivery.delivery_id = INVOICE.delivery_id
INNER JOIN COIN
ON INVOICE.coin_id = COIN.coin_id;
```

begin

open myC;

loop

```
fetch myC into customer_name, customer_surname, coin_id, coin_price, coin_discount,
coin_del_date, invoice_date;
```

exit when myC %notfound or myC %rowcount > 5;

dbms\_output.put\_line('FIRST NAME:' || customer\_name);

dbms\_output.put\_line('SURNAME:' || customer\_surname);

dbms\_output.put\_line('COIN ID:' || coin\_id);

dbms\_output.put\_line('PRICE:' || coin\_price);

dbms\_output.put\_line('DISCOUNT:' || coin\_discount);

dbms\_output.put\_line('DELIVERY DATE:' || coin\_del\_date);

dbms\_output.put\_line('DELIVERY DATE:' || invoice\_date);

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

end loop;

close myC;

end;

 Task completed in 0.038 seconds

FIRST NAME:Lucy  
SURNAME:Williams  
COIN ID:7115  
PRICE:11999  
DISCOUNT:8999.25  
DELIVERY DATE:25/MAY/21  
DELIVERY DATE:18/MAY/21  
-----

FIRST NAME:Julien  
SURNAME:Hendricks  
COIN ID:7112  
PRICE:12999  
DISCOUNT:9749.25  
DELIVERY DATE:27/MAY/21  
DELIVERY DATE:19/MAY/21  
-----

FIRST NAME:Sam  
SURNAME:Clark  
COIN ID:7112  
PRICE:12999  
DISCOUNT:9749.25  
DELIVERY DATE:27/MAY/21  
DELIVERY DATE:19/MAY/21  
-----

PL/SQL procedure successfully completed.