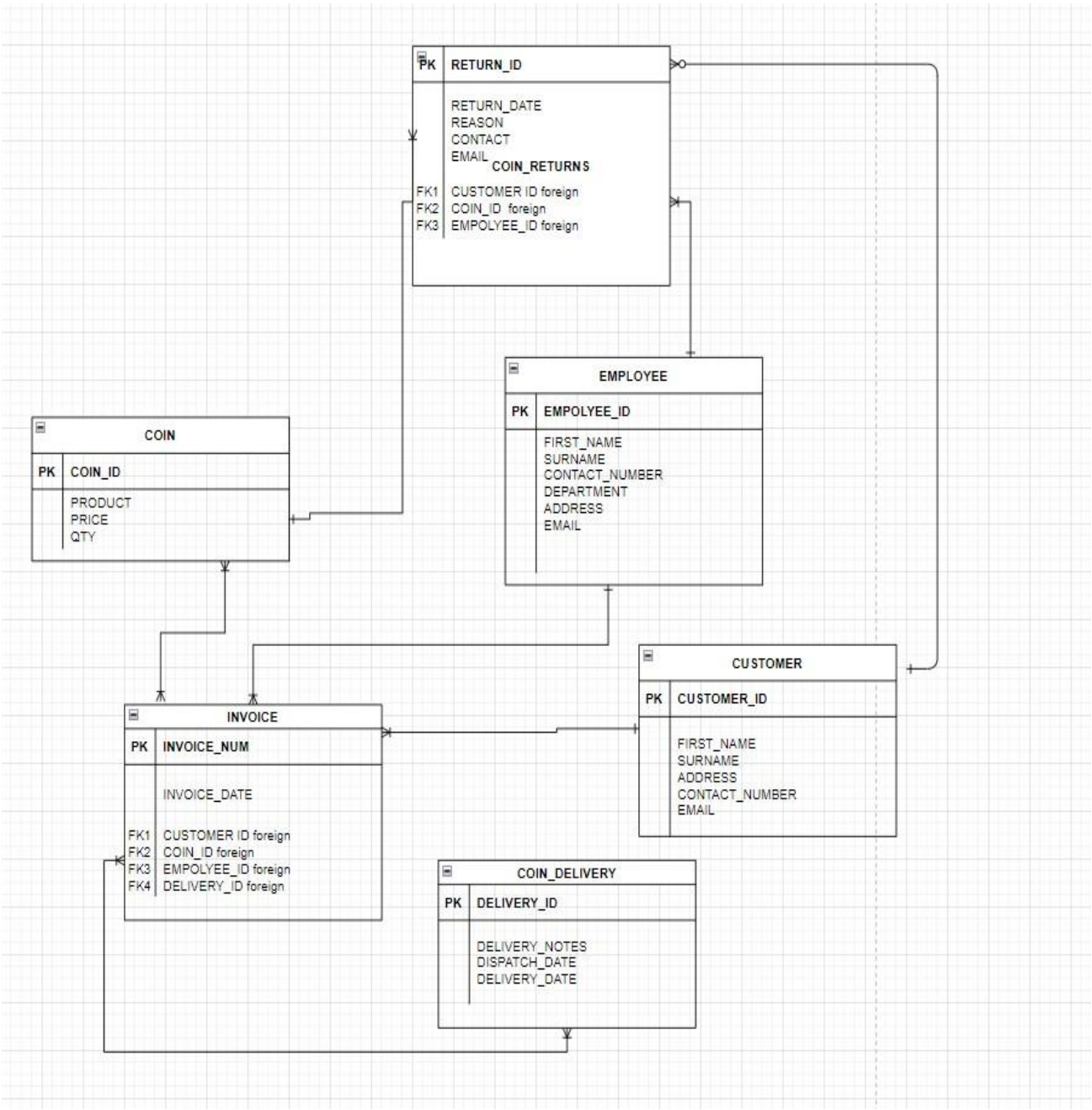
Entity Relationship Diagram (ERD



PL/SQL query:

```
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
  and 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;
```

Using the Entity Relationship Diagram, I've created the tables and insert the values supplied in each table.

```
CREATE TABLE CUSTOMER(
        CUSTOMER_ID SMALLINT NOT NULL PRIMARY KEY,
        FIRST_NAME VARCHAR (10),
        SURNAME VARCHAR (10),
        ADDRESS VARCHAR (255),
        CONTACT_NUMBER INT,
        EMAIL VARCHAR (20)
);

INSERT INTO CUSTOMER VALUES ('11011' , 'MAT' , 'SMITH' , '10 WATER RD' , '0877277521' , 'MSMITH@ISAT.COM');
INSERT INTO CUSTOMER VALUES ('11012' , 'JULIEN' , '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 RD' , '0827238521' , 'LW@MCAL.CO.ZA');

SELECT * FROM CUSTOMER;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 5 in 0.352 seconds

	CUSTOMER_ID	FIRST_NAME	SURNAME	ADDRESS	CONTACT_NUMBER	EMAIL
1	11011	Mat	Smith	18 Water Rd	0877277521	msmith@isat.com
2	11012	Julien	Hendricks	22 Water Rd	0863257857	jh@mcom.co.za
3	11013	Sam	Clark	101 Summer Lane	0834567891	sclark@mcom.co.za
4	11014	Kevin	Jones	55 Mountain Way	0612547895	kj@isat.co.za
5	11015	Lucy	Williams	5 Main Rd	0827238521	lw@mcal.co.za

```
CREATE TABLE EMPLOYEE(
```

```
        EMP_ID VARCHAR (8) NOT NULL PRIMARY KEY,
        FIRST_NAME VARCHAR (15),
        SURNAME VARCHAR (15),
        CONTACT_NUMBER INT,
        DEPARTMENT VARCHAR (10),
        ADDRESS VARCHAR (255),
        EMAIL VARCHAR (20)
);
```

```
INSERT INTO EMPLOYEE VALUES ('EMP101' , 'XANDER' , 'DAVIS' , '0877277521' , 'SALES' , '10 MAIN ROAD' , 'XAND#ISAT.COM');
INSERT INTO EMPLOYEE VALUES ('EMP102' , 'STEVEN' , 'MARKS' , '0837377522' , 'MARKETING' , '18 WATER ROAD' , 'SM@ISAT.COM');
INSERT INTO EMPLOYEE VALUES ('EMP103' , 'JESSICA' , 'ANDREWS' , '0817117523' , 'SALES' , '21 CIRCLE LANE' , 'JA@ISAT.COM');
INSERT INTO EMPLOYEE VALUES ('EMP104' , 'WAYNE' , 'DRYER' , '0797215244' , 'SALES' , '1 SEA ROAD' , 'DRYER@ISAT.COM');
INSERT INTO EMPLOYEE VALUES ('EMP105' , 'XOLANI' , 'SAMSON' , '0827122255' , 'MANAGER' , '12 MAIN ROAD' , 'XOSAM@ISAT.COM');
```

```
SELECT * FROM EMPLOYEE;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.006 seconds

	EMPLOYEE_ID	FIRST_NAME	SURNAME	CONTACT_NUMBER	DEPARTMENT	ADDRESS	EMAIL
1	emp101	Xander	Davis	0877277521	sales	10 Main Road	xand@isat.com
2	emp102	Steven	Marks	0837377522	marketing	18 Water Road	sm@isat.com
3	emp103	Jessica	Andrews	0817117523	sales	21 Circle Lane	ja@isat.com
4	emp104	Wayne	Dryer	0797215244	sales	1 Sea Road	dryer@isat.com
5	emp105	Xolani	Samson	0827122255	manager	12 Main Road	xosam@isat.com

```
CREATE TABLE COIN_DELIVERY(
```

```
        DELIVERY_ID SMALLINT NOT NULL PRIMARY KEY,
        DELIVERY_NOTES VARCHAR (40),
        DISPATCH_DATE VARCHAR (30) NOT NULL,
        DELIVERY_DATE VARCHAR (30) NOT NULL
);

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 ADDRESS' , '20 MAY 2021' , '25 MAY 2021');
INSERT INTO COIN_DELIVERY VALUES ('517' , 'DOUBLE PACKAGING REQUESTED' , '25 MAY 2021' , '27 MAY 2021');

SELECT * FROM COIN_DELIVERY;
```

Script Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 7 in 0.004 seconds

	DELIVERY_ID	DELIVERY_NOTES	DISPATCH_DATE	DELIVERY_DATE
1	511	DOUBLE PACKAGING REQUESTED	10 MAY 2021	15 MAY 2021
2	512	DELIVERY TO WORK ADDRESS	12 MAY 2021	15 MAY 2021
3	513	SIGNATURE REQUIRED	12 MAY 2021	17 MAY 2021
4	514	NO NOTES	12 MAY 2021	15 MAY 2021
5	515	BIRTHDAY PRESENT WRAPPING REQUIRED	18 MAY 2021	19 MAY 2021
6	516	DELIVERY TO WORK ADDRESS	20 MAY 2021	25 MAY 2021
7	517	DOUBLE PACKAGING REQUESTED	25 MAY 2021	27 MAY 2021

```
CREATE TABLE COIN(
        COIN_ID VARCHAR (10) NOT NULL PRIMARY KEY,
        PRODUCT VARCHAR (50),
        PRICE INT,
        QTY SMALLINT
);
INSERT INTO COIN VALUES ('7111' , '1oz GOLD KRUGER RAND' , 5999 , 10);
```

```
INSERT INTO COIN VALUES ('7112' , '1oz SILVER KRUGER RAND' , 12999 , 8);
INSERT INTO COIN VALUES ('7113' , 'GOLD BIG 5 UNCIRCULATED' , 15999 , 8);
INSERT INTO COIN VALUES ('7114' , 'SILVERBIG 5 PACK' , 7999 , 5);
INSERT INTO COIN VALUES ('7115' , '1oz GOLD PALAEONTOLOGY' , 11999 , 15);
INSERT INTO COIN VALUES ('7116' , '1oz SILVER PALAEONTOLOGY' , 7999 , 12);

SELECT * FROM COIN;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 6 in 0.004 seconds

	PRODUCT_ID	PRODUCT	PRICE	QTY
1	7111	loz Gold Kruger Rand	5999	10
2	7112	loz Silver Kruger Rand	12999	8
3	7113	Gold Big 5 Uncirculated	15999	8
4	7114	Silver Big 5 Pack	7999	5
5	7115	loz Gold Palaeontology	11999	15
6	7116	loz Silver Palaeontology	7999	12

```
CREATE TABLE COIN_RETURNS(
        RETURN_ID VARCHAR(10) NOT NULL PRIMARY KEY,
        RETURN_DATE VARCHAR(30),
        REASON VARCHAR (150),
        CUSTOMER_ID SMALLINT ,
        COIN_ID VARCHAR (10),
        EMP_ID VARCHAR (8),
  FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID),
  FOREIGN KEY (COIN_ID) REFERENCES COIN(COIN_ID),
  FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID)
);


INSERT INTO COIN_RETURNS VALUES ('RET001' , '25 MAY 2021' , 'CUSTOMER NOT SATISFIED WITH
PRODUCT' , 11011 , '7116' , 'EMP101');
INSERT INTO COIN_RETURNS VALUES ('RET002' , '25 MAY 2021' , 'PRODUCT MISSING PART' ,
11013 , '7114' , 'EMP103');


SELECT * FROM COIN_RETURNS;
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.004 seconds

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

```
CREATE TABLE INVOICE(
        INVOICE_NUM VARCHAR(50)  NOT NULL PRIMARY KEY,
        CUSTOMER_ID SMALLINT,
        INVOICE_DATE VARCHAR(30),
        EMP_ID VARCHAR (8),
        COIN_ID VARCHAR (10),
        DELIVERY_ID SMALLINT,
  FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID),
  FOREIGN KEY (COIN_ID) REFERENCES COIN(COIN_ID),
  FOREIGN KEY (DELIVERY_ID) REFERENCES COIN_DELIVERY(DELIVERY_ID)
);

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');

SELECT * FROM INVOICE;
```

```
select * from invoice
```

Script Output × Query Result ×

SQL | All Rows Fetched: 8 in 0.008 seconds

	INVOICE_NUM	CUSTOMER_ID	INVOICE_DATE	EMPLOYEE_ID	PRODUCT_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

<u>Creating a SQL query to display the combined customer name, employee id, delivery notes, coin ordered and invoice number. This query only display the results that have any invoice date from 18th May 2021</u>

```
SELECT C.FIRST_NAME||', '|| C.SURNAME AS "CUSTOMER",
E.EMP_ID AS "EMPLOYEE_ID",
D.DELIVERY_NOTES AS "DESANDCRIPTION",
C.PRODUCT AS "COIN",
IV.INVOICE_NUM AS "INVOICE_NUM",
IV.INVOICE_DATE AS "INVOICE_DATE"
FROM CUSTOMER C, COIN_DELIVERY D, COIN CC, INVOICE IV, EMPLOYEE E
WHERE IV.INVOICE_DATE >= '17?MAY?21'
AND E.EMP_ID = IV.EMP_ID
AND D.DELIVERY_ID = IV.DELIVERY_ID
AND CC.COIN_ID = IV.COIN_ID
AND C.CUSTOMER_ID = IV.CUTOMER_ID;
```

Query Result

All Rows Fetched: 3 in 1.876 seconds

	CUSTOMERCOIN.FIRST_NAME||','||CUSTOMERCOIN.SURNAME	EMPLOYEE_ID	DESCRIPTION	PRODUCT	INVOICE_NUM	INVOICE_DATE
1	Sam,Clark	emp105	Double packaging requested	1oz Silver Kruger Rand	8118	19 May 2021
2	Lucy,Williams	emp103	Delivery to work address	1oz Gold Palaeontology	8116	18 May 2021
3	Julien,Hendricks	emp105	Double packaging requested	1oz Silver Kruger Rand	8117	19 May 2021

Creating a View to display the employee id, first name and surname. This query include the coin price and a 10% commission for the sales made by the employees.

```
CREATE VIEW EMPLOYEE_INFO AS
SELECT E.EMPLOYEE_ID, E.FIRST_NAME, E.SURNAME, CO.PRICE AS COIN_PRICE, CO.PRICE*10/100 as
COMMISIONER
FROM EMPLOYEE E,  JOIN
INVOICE
where E.EMPLOYEE_ID = IV.EMPLOYEE_ID AND CC.COIN_ID = IV.COIN_ID ; select
* FROM EMPLOYEE_INFO;
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.014 seconds

	EMPLOYEE_ID	FIRST_NAME	SURNAME	COIN_PRICE	COMMISSION
1	emp105	Xolani	Samson	R 25998	R 2599.8
2	emp102	Steven	Marks	R 17998	R 1799.8
3	emp101	Xander	Davis	R 20998	R 2099.8
4	emp103	Jessica	Andrews	R 17998	R 1799.8

Creating a PL/SQL query to display the combined customer name, coin purchased, coin price and

the delivery notes. In this query it only display the coins purchased with a price less than or equal

to R 8 000.

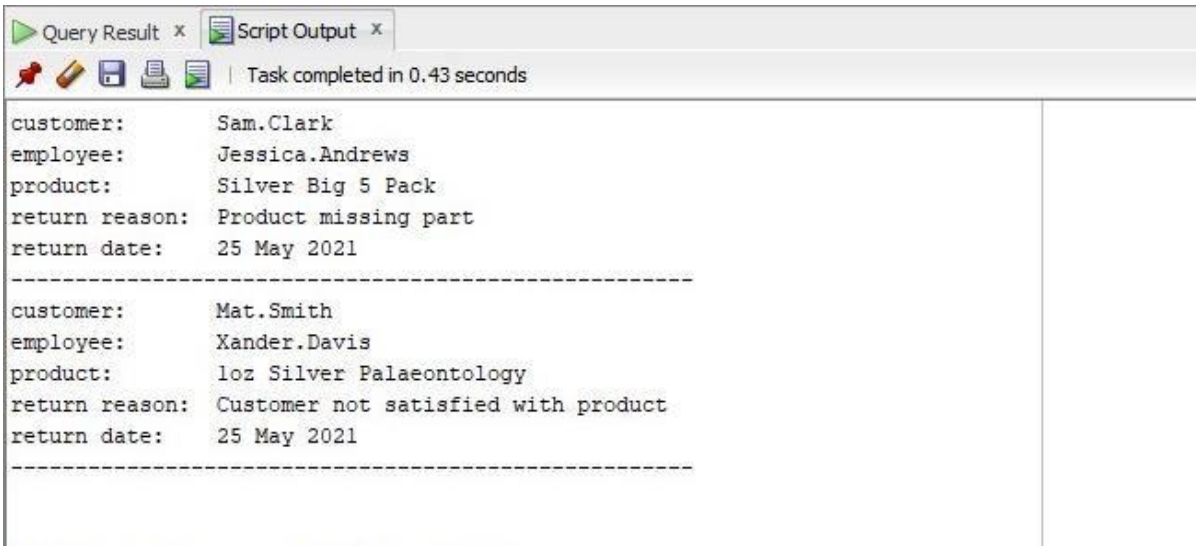
```
SET SERVEROUTPUT ON

DECLARE
C_CUSTOMER_NAME customer.first_name%type;
C_CUSTOMER_SURNAME customer.surname%type;
C_DESCRIPTION COIN_DELIVERY.DELIVERY_NOTES%type;
C_PRICE COIN.PRICE%TYPE;
C_PRODUCT COIN.PRODUCT%TYPE;

CURSOR C_COIN_CURSOR IS
SELECT FIRST_NAME, SURNAME, PRODUCT, PRICE, DELIVERY_NOTES
FROM INVOICE IV
JOIN CUSTOMER C ON C.CUSTOMER_ID = IV.CUSTOMER_ID
JOIN COIN C ON CC.COIN_ID = IV.COIN_ID
JOIN COIN_DELIVERY D  ON D.DELIVERY_ID = IV.DELIVERY_ID
WHERE P.PRICE <= 8000
ORDER BY P.PRICE ;
BEGIN

OPEN C_COIN_CURSOR;

LOOP
FETCH C_COIN_CURSOR
INTO C_CUSTOMER_NAME, C_CUSTOMER_SURNAME, C_PRODUCT, C_PRICE, C_DESCRIPTION;

EXIT WHEN C_COIN_CURSOR%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('CUSTOMER: ' || C_CUSTOMER_NAME || ' '|| C_CUSTOMER_SURNAME);
DBMS_OUTPUT.PUT_LINE('COIN: ' || C_PRODUCT);
DBMS_OUTPUT.PUT_LINE('PRICE: R' || C_PRICE);
DBMS_OUTPUT.PUT_LINE('NOTES: ' || C_DESCRIPTION);
DBMS_OUTPUT.PUT_LINE('------------------------');

END LOOP;

END;
```

```
Query Result x   Script Output x
Task completed in 0.082 seconds
customer:   :Mat,Smith
coin:       :7111
price:      r :5999
notes:      :Double packaging requested
-----------------------------------------------------
customer:   :Lucy,Williams
coin:       :7111
price:      r :5999
notes:      :No notes
-----------------------------------------------------
customer:   :Sam,Clark
coin:       :7116
price:      r :7999
notes:      :Delivery to work address
-----------------------------------------------------
```

<u>Creating a PL/SQL query to display the customer's name, employee name and the coin product</u>
<u>returned. In this query it also display the reason why the customer returned the coin along with the</u>
<u>return date.</u>

```
SET SERVEROUTPUT ON

DECLARE
C_CUSTOMER_NAME CUSTOMER.FIRST_NAME%TYPE;
C_CUSTOMER_SURNAME CUSTOMER.SURNAME%TYPE;
C_EMPLOYEE_NAME EMPLOYEE.FIRST_NAME%TYPE;
C_EMPLOYEE_SURNAME EMPLOYEE.SURNAME%TYPE;
C_PRODUCT COIN.PRODUCT%TYPE;
C_RETURN_REASON COIN_RETURNS.REASON%TYPE;
C_RETURN_DATE COIN_RETURNS.RETURN_DATE%TYPE;

CURSOR C_COIN_RETURN IS
SELECT C.FIRST_NAME, C.SURNAME, E.FIRST_NAME, E.SURNAME, PRODUCT, REASON, RETURN_DATE
FROM COIN_RETURNS CR
JOIN CUSTOMER C ON C.CUSTOMER_ID = IV.CUSTOMER_ID
JOIN EMPLOYEE E ON E.EMPLOYEE_ID = IV.EMPLOYEE_ID
JOIN COIN C ON CC.COIN_ID = IV.COIN_ID;
```

```
BEGIN

OPEN C_COIN_RETURN;

LOOP
FETCH C_COIN_RETURN
INTO C_CUSTOMER_NAME, C_CUSTOMER_SURNAME, C_EMPLOYEE_NAME, C_EMPLOYEE_SURNAME,
C_PRODUCT, C_RETURN_REASON, C_RETURN_DATE;
EXIT WHEN C_COIN_RETURN%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('CUSTOMER: ' || C_CUSTOMER_NAME || ' '|| C_CUSTOMER_SURNAME);
DBMS_OUTPUT.PUT_LINE('EMPLOYEE: ' || C_EMPLOYEE_NAME || ' ' || C_EMPLOYEE_SURNAME);
DBMS_OUTPUT.PUT_LINE('PRODUCT: ' || C_PRODUCT);
DBMS_OUTPUT.PUT_LINE('REASON: ' || C_RETURN_REASON);
DBMS_OUTPUT.PUT_LINE('DATE: ' || C_RETURN_DATE);
DBMS_OUTPUT.PUT_LINE('------------------------');

END LOOP;

END;
```



Creating a PL/SQL query to display the invoice number, coin name, coin price and delivery notes. In this query it determines whether the coin product is a premium or standard product. All coins that are valued at R10 000 and above are considered Premium coins. In your query use invoice number 8115 only.

```
SET SERVEROUTPUT ON
DECLARE
C_INVOICE_NUMBER INVOICE.INVOICE_NUM%TYPE;
C_PRODUCT COIN.PRODUCT%TYPE;
C_PRICE COIN.PRICE%TYPE;
C_DELIVERY_NOTE COIN_DELIVERY.DELIVERY_NOTES%TYPE;
C_TYPE_A VARCHAR2(50);

CURSOR C_INVOICE_P IS
SELECT INVOICE_NUM, PRODUCT, PRICE, DELIVERY_NOTES
FROM COIN C
JOIN INVOICE I ON CO.COIN_ID = I.COIN_ID
JOIN COIN_DELIVERY CD ON D.DELIVERY_ID = IV.DELIVERY_ID
WHERE IV.INVOICE_NUM = 8115;

BEGIN

OPEN C_INVOICE_P;

 LOOP

 FETCH C_INVOICE_P
 INTO C_INVOICE_NUMBER, C_PRODUCT, C_PRICE, C_DELIVERY_NOTE;
 IF C_PRICE  >= 10000 THEN
 C_TYPE_A := 'PREMIUM';
 ELSE
 C_TYPE_A := 'STANDARD';

  END IF;
 EXIT WHEN C_INVOICE_P%NOTFOUND;


DBMS_OUTPUT.PUT_LINE('INVOICE NUMBER: ' || C_INVOICE_NUMBER);
DBMS_OUTPUT.PUT_LINE('COIN PRODUCT: ' || C_PRODUCT);
DBMS_OUTPUT.PUT_LINE('PRICE: ' || C_PRICE);
DBMS_OUTPUT.PUT_LINE('DELIVERY NOTE: ' || C_DELIVERY_NOTE);
DBMS_OUTPUT.PUT_LINE('COIN TYPE: ' || C_TYPE_A);
DBMS_OUTPUT.PUT_LINE('------------------------');


END LOOP;
 END;
```

```
Query Result x   Script Output x
Task completed in 0.493 seconds
invoice number:     8115
coin product:       1oz Gold Palaeontology
price:             R 11999
delivery note:      Birthday present wrapping required
coin type:     premium
----------------------------------------------
```

<u>Creating a PL/SQL query to display the customer name, coin id and coin price. In this query it display</u>
<u>a 25% discount for the invoice dates 18 May 2021 to 20 May 2021.</u>

```
SET SERVEROUTPUT ON

DECLARE
C_CUSTOMER_NAME CUSTOMER.FIRST_NAME%TYPE;
C_CUSTOMER_SURNAME CUSTOMER.SURNAME%TYPE;
C_COIN_ID COIN.COIN_ID%TYPE;
C_PRICE COIN.PRICE%TYPE;
C_DISCOUNT_PRICE NUMBER;
C_DELIVERY_DATE COIN_DELIVERY.DELIVERY_DATE%TYPE;
C_INVOICE_DATE INVOICE.INVOICE_DATE%TYPE;



CURSOR C_DISCOUNT IS
SELECT FIRST_NAME, SURNAME, CC.COIN_ID, CO.PRICE, CD.DELIVERY_DATE, I.INVOICE_DATE
FROM INVOICE I
JOIN CUSTOMER C ON C.CUSTOMER_ID = IV.CUSTOMER_ID
JOIN COIN_DELIVERY CD ON D.DELIVERY_ID = IV.DELIVERY_ID
JOIN COIN C ON CC.COIN_ID = IV.COIN_ID
WHERE IV.INVOICE_DATE BETWEEN '18 MAY 2021' AND '20 MAY 2021' ;

BEGIN

OPEN C_DISCOUNT;
```

```
LOOP

FETCH C_DISCOUNT
INTO C_CUSTOMER_NAME, C_CUSTOMER_SURNAME, C_COIN_ID, C_PRICE, C_DELIVERY_DATE,
C_INVOICE_DATE;


C_DISCOUNT_PRICE :=  C_PRICE * 25/100;

EXIT WHEN C_DISCOUNT%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('FIRST NAME: ' || C_CUSTOMER_NAME);
DBMS_OUTPUT.PUT_LINE('SURNAME: ' || C_CUSTOMER_SURNAME);
DBMS_OUTPUT.PUT_LINE('COIN ID: ' || C_COIN_ID);
DBMS_OUTPUT.PUT_LINE('PRICE: ' || C_PRICE);
DBMS_OUTPUT.PUT_LINE('DISCOUNT: ' || C_DISCOUNT_PRICE); DBMS_OUTPUT.PUT_LINE('DELIVERY DATE:
' || C_DELIVERY_DATE);
DBMS_OUTPUT.PUT_LINE('INVOICE DATE: ' || C_INVOICE_DATE);
DBMS_OUTPUT.PUT_LINE('------------------------');


END LOOP;

END;
```

Query Result × | Script Output ×

Task completed in 0.33 seconds

```
first name:Lucy
surname: Williams
product: 7115
price: R11999
discount: R14998.75
coin delivery: 25 May 2021
invoice date: 18 May 2021
----------------------------------------
first name:Sam
surname: Clark
product: 7112
price: R12999
discount: R16248.75
coin delivery: 27 May 2021
invoice date: 19 May 2021
----------------------------------------
first name:Julien
surname: Hendricks
product: 7112
price: R12999
discount: R16248.75
coin delivery: 27 May 2021
invoice date: 19 May 2021
----------------------------------------
```