**Kgaogelo Sekhaolelo**

**Student no: ST10097375**

**Advanced Databases Assignment**

Table of Contents

[**--Question1** 1](#_Toc134091637)

[--Question2 2](#_Toc134091638)

[--Question3 8](#_Toc134091639)

[--Question4 8](#_Toc134091640)

[--Question5 9](#_Toc134091641)

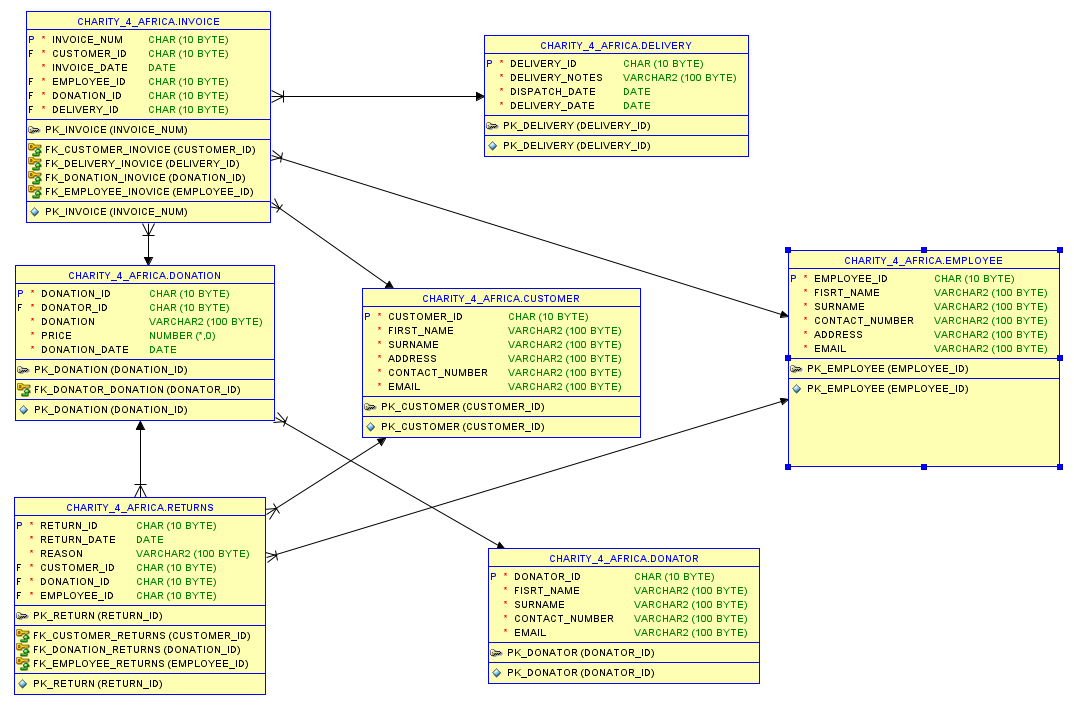
[--Question6 11](#_Toc134091642)

[--Question7 13](#_Toc134091643)

[--Question8 16](#_Toc134091644)

[--Question9 16](#_Toc134091645)

# **--Question1**



# --Question2

CREATE TABLE CUSTOMER(

CUSTOMER\_ID CHAR(10) NOT NULL,

FIRST\_NAME VARCHAR2(100) NOT NULL,

SURNAME VARCHAR2(100) NOT NULL,

ADDRESS VARCHAR2(100) NOT NULL,

CONTACT\_NUMBER VARCHAR2(100) NOT NULL,

EMAIL VARCHAR2(100) NOT NULL,

CONSTRAINT PK\_CUSTOMER PRIMARY KEY (CUSTOMER\_ID)

);

CREATE TABLE EMPLOYEE(

EMPLOYEE\_ID CHAR(10) NOT NULL,

FISRT\_NAME VARCHAR2(100) NOT NULL,

SURNAME VARCHAR2(100) NOT NULL,

CONTACT\_NUMBER VARCHAR2(100) NOT NULL,

ADDRESS VARCHAR2(100) NOT NULL,

EMAIL VARCHAR2(100) NOT NULL,

CONSTRAINT PK\_EMPLOYEE PRIMARY KEY (EMPLOYEE\_ID)

);

CREATE TABLE DONATOR(

DONATOR\_ID CHAR(10) NOT NULL,

FISRT\_NAME VARCHAR2(100) NOT NULL,

SURNAME VARCHAR2(100) NOT NULL,

CONTACT\_NUMBER VARCHAR2(100) NOT NULL,

EMAIL VARCHAR2(100) NOT NULL,

CONSTRAINT PK\_DONATOR PRIMARY KEY (DONATOR\_ID)

);

CREATE TABLE DONATION(

DONATION\_ID CHAR(10) NOT NULL,

DONATOR\_ID CHAR(10) NOT NULL,

DONATION VARCHAR2(100) NOT NULL,

PRICE INT NOT NULL,

DONATION\_DATE DATE NOT NULL,

CONSTRAINT PK\_DONATION PRIMARY KEY (DONATION\_ID),

CONSTRAINT FK\_DONATOR\_DONATION FOREIGN KEY (DONATOR\_ID) REFERENCES DONATOR(DONATOR\_ID)

);

CREATE TABLE DELIVERY(

DELIVERY\_ID CHAR(10) NOT NULL,

DELIVERY\_NOTES VARCHAR2(100) NOT NULL,

DISPATCH\_DATE DATE NOT NULL,

DELIVERY\_DATE DATE NOT NULL,

CONSTRAINT PK\_DELIVERY PRIMARY KEY (DELIVERY\_ID)

);

CREATE TABLE RETURNS(

RETURN\_ID CHAR(10) NOT NULL,

RETURN\_DATE DATE NOT NULL,

REASON VARCHAR2(100) NOT NULL,

CUSTOMER\_ID CHAR(10) NOT NULL,

DONATION\_ID CHAR(10) NOT NULL,

EMPLOYEE\_ID CHAR(10) NOT NULL,

CONSTRAINT PK\_RETURN PRIMARY KEY (RETURN\_ID),

CONSTRAINT FK\_CUSTOMER\_RETURNS FOREIGN KEY (CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID),

CONSTRAINT FK\_DONATION\_RETURNS FOREIGN KEY (DONATION\_ID) REFERENCES DONATION(DONATION\_ID),

CONSTRAINT FK\_EMPLOYEE\_RETURNS FOREIGN KEY (EMPLOYEE\_ID) REFERENCES EMPLOYEE(EMPLOYEE\_ID)

);

CREATE TABLE INVOICE(

INVOICE\_NUM CHAR(10) NOT NULL,

CUSTOMER\_ID CHAR(10) NOT NULL,

INVOICE\_DATE DATE NOT NULL,

EMPLOYEE\_ID CHAR(10) NOT NULL,

DONATION\_ID CHAR(10) NOT NULL,

DELIVERY\_ID CHAR(10) NOT NULL,

CONSTRAINT PK\_INVOICE PRIMARY KEY (INVOICE\_NUM),

CONSTRAINT FK\_CUSTOMER\_INOVICE FOREIGN KEY (CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID),

CONSTRAINT FK\_EMPLOYEE\_INOVICE FOREIGN KEY (EMPLOYEE\_ID) REFERENCES EMPLOYEE(EMPLOYEE\_ID),

CONSTRAINT FK\_DONATION\_INOVICE FOREIGN KEY (DONATION\_ID) REFERENCES DONATION(DONATION\_ID),

CONSTRAINT FK\_DELIVERY\_INOVICE FOREIGN KEY (DELIVERY\_ID) REFERENCES DELIVERY(DELIVERY\_ID)

);

INSERT INTO CUSTOMER VALUES('11011','Jack','Smith','18 Water Rd','087 727 7521','jsmith@isat.com');

INSERT INTO CUSTOMER VALUES('11012','Pat','Hendricks','22 Water Rd','086 325 7857','ph@mcom.co.za');

INSERT INTO CUSTOMER VALUES('11013','Andre','Clark','101 Summer Lane','083 456 7891','aclack@mcom.co.za');

INSERT INTO CUSTOMER VALUES('11014','Kevin','Jones','55 Mountain way','061 254 7895','kj@isat.co.za');

INSERT INTO CUSTOMER VALUES('11015','Lucy','Williams','5 Main rd','082 723 8521','lw@mcal.co.za');

INSERT INTO EMPLOYEE VALUES('emp101','Jeff','Davis','087 727 7521','10 main road','jand@isat.com');

INSERT INTO EMPLOYEE VALUES('emp102','Kevin','Marks','083 737 7522','18 water road','km@isat.com');

INSERT INTO EMPLOYEE VALUES('emp103','Adanya','Andrews','081 711 7523','21 circle lane','aa@isat.com');

INSERT INTO EMPLOYEE VALUES('emp104','Adebayo','Dryer','079 721 5244','1 sea road ','aryer@isat.com');

INSERT INTO EMPLOYEE VALUES('emp105','Xolani','Samson','082 712 2255','12 main road','xosam@isat.com');

INSERT INTO DONATOR VALUES('20111','Jeff','Watson','082 717 2250','jwatson@ymail.com');

INSERT INTO DONATOR VALUES('20112','Stephen','Jones','083 786 5670','joness@ymail.com');

INSERT INTO DONATOR VALUES('20113','James','Joe','087 897 8650','jj@ymail.com');

INSERT INTO DONATOR VALUES('20114','Kelly','Ross','082 657 5650','kross@ymail.com');

INSERT INTO DONATOR VALUES('20115','Abraham','Clark','079 765 6430','aclark@ymail.com');

INSERT INTO DONATION VALUES('7111','20111','KIC Fridge','599','1 May 2022');

INSERT INTO DONATION VALUES('7112','20112','Samsung 42inch','1299','3 May 2022');

INSERT INTO DONATION VALUES('7113','20113','Sharp Mircowave','1599','3 May 2022');

INSERT INTO DONATION VALUES('7114','20115','6 Seat Dining room table','799','5 May 2022');

INSERT INTO DONATION VALUES('7115','20114','Lazyboy Sofa','1199','7 May 2022');

INSERT INTO DONATION VALUES('7116','20113','JVC Surround Sound System','179','9 May 2022');

INSERT INTO DELIVERY VALUES('511','Double packaging requested','10 May 2022','15 May 2022');

INSERT INTO DELIVERY VALUES('512','Delivery to work address','12 May 2022','15 May 2022');

INSERT INTO DELIVERY VALUES('513','Signature required','12 May 2022','17 May 2022');

INSERT INTO DELIVERY VALUES('514','No notes','12 May 2022','15 May 2022');

INSERT INTO DELIVERY VALUES('515','Birthday present wrapping required','18 May 2022','19 May 2022');

INSERT INTO DELIVERY VALUES('516','Delivery to work address','20 May 2022','25 May 2022');

INSERT INTO RETURNS VALUES('ret001','25 May 2022','Customer not satisfied with product','11011','7116','emp101');

INSERT INTO RETURNS VALUES('ret002','25 May 2022','Product had broken section','11013','7114','emp103');

INSERT INTO INVOICE VALUES('8111','11011','15 May 2022','emp103','7111','511');

INSERT INTO INVOICE VALUES('8112','11013','15 May 2022','emp101','7114','512');

INSERT INTO INVOICE VALUES('8113','11012','17 May 2022','emp101','7112','513');

INSERT INTO INVOICE VALUES('8114','11015','17 May 2022','emp102','7113','514');

INSERT INTO INVOICE VALUES('8115','11011','17 May 2022','emp102','7115','515');

INSERT INTO INVOICE VALUES('8116','11015','18 May 2022','emp103','7116','516');

# --Question3

SELECT customer.FIRST\_NAME, employee.EMPLOYEE\_ID,donation.DONATION, delivery.DELIVERY\_NOTES, invoice.INVOICE\_NUM, invoice.INVOICE\_DATE

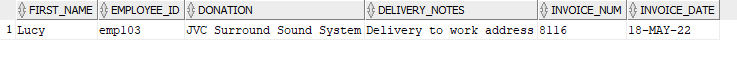
FROM employee INNER JOIN invoice ON employee.EMPLOYEE\_ID = invoice.EMPLOYEE\_ID

INNER JOIN customer ON invoice.CUSTOMER\_ID = customer.CUSTOMER\_ID

INNER JOIN delivery ON invoice.DELIVERY\_ID = delivery.DELIVERY\_ID

INNER JOIN donation ON invoice.DONATION\_ID = donation.DONATION\_ID

WHERE invoice.INVOICE\_DATE >= '18 May 2022';



----VARCHAR2: datatype that specifies a variable-length character string in the database character set.

----CHAR: datatype that stores fixed-length character strings.

# --Question4

--Q.4.1

CREATE OR REPLACE VIEW Question4

AS

SELECT employee.EMPLOYEE\_ID, employee.FISRT\_NAME, employee.SURNAME, invoice.INVOICE\_DATE

FROM employee, invoice

WHERE invoice.INVOICE\_DATE = '18 May 2022' AND employee.EMPLOYEE\_ID = 'emp103';

SELECT \* FROM Question4;



--Q.4.2

* View in Oracle: is a named, validated SQL query kept in the data dictionary of the Oracle Database (Staff, 2020:07).
* **Two benefits gained by CHARITY 4 AFRICA for using the View**

**-** (1) Views will provide an additional level of security by restricting access to predetermined set of rows and columns of a table (Staff, 2020:07).

- (2) View can be used to hide some of the columns from the table (Staff, 2020:07).

# --Question5

SET SERVEROUTPUT ON;

DECLARE

v\_Customer customer.FIRST\_NAME%TYPE;

v\_DonationP donation.DONATION%TYPE;

v\_Price donation.PRICE%TYPE;

v\_ReturnReason returns.REASON%TYPE;

CURSOR info

IS

SELECT customer.FIRST\_NAME||' '||customer.SURNAME AS CUSTOMER, donation.DONATION, donation.PRICE,returns.REASON

FROM returns INNER JOIN customer ON customer.CUSTOMER\_ID = returns.CUSTOMER\_ID

INNER JOIN donation ON returns.DONATION\_ID = donation.DONATION\_ID

ORDER BY customer.FIRST\_NAME;

BEGIN

OPEN info;

LOOP

FETCH info INTO v\_Customer, v\_DonationP, v\_Price, v\_ReturnReason ;

EXIT WHEN info%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER:  :'||v\_Customer);

DBMS\_OUTPUT.PUT\_LINE('DONATION PURCHASED :'||v\_DonationP);

DBMS\_OUTPUT.PUT\_LINE('DONATION PURCHASED :'||v\_Price);

DBMS\_OUTPUT.PUT\_LINE('DONATION PURCHASED :'||v\_ReturnReason);

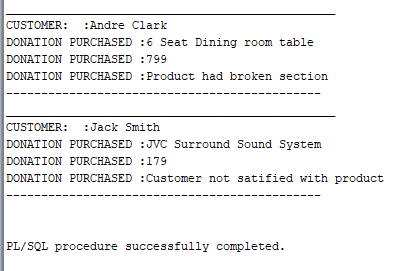
DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------');

END LOOP;

CLOSE info;

END;

/



# --Question6

--Q.6.1

SET SERVEROUTPUT ON;

DECLARE

v\_Customer customer.FIRST\_NAME%TYPE;

v\_Employee employee.FISRT\_NAME%TYPE;

v\_Donation donation.DONATION%TYPE;

v\_Dis DATE;

v\_Delivery DATE;

v\_Daystodelivery INT;

CURSOR info

IS

SELECT customer.FIRST\_NAME||','||customer.SURNAME AS CUSTOMER,employee.

FISRT\_NAME||','||employee.SURNAME AS EMPLOYEE,

donation.DONATION,delivery.DISPATCH\_DATE,delivery.DELIVERY\_DATE,round(delivery.DELIVERY\_DATE - delivery.DISPATCH\_DATE) AS\_DAYS\_TO\_DELIVERY

FROM invoice

INNER JOIN customer ON invoice.CUSTOMER\_ID = customer.CUSTOMER\_ID

INNER JOIN employee ON invoice.EMPLOYEE\_ID = employee.EMPLOYEE\_ID

INNER JOIN donation ON invoice.DONATION\_ID = donation.DONATION\_ID

INNER JOIN delivery ON invoice.DELIVERY\_ID = delivery.DELIVERY\_ID

WHERE customer.CUSTOMER\_ID = '11013';

BEGIN

OPEN info;

LOOP

FETCH info INTO v\_Customer,v\_Employee,v\_Donation,v\_Dis,v\_Delivery, v\_Daystodelivery;

EXIT WHEN info%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('-------------------------------------------------');

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER :'||v\_Customer);

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE :'||v\_Employee);

DBMS\_OUTPUT.PUT\_LINE('DONATION :'||v\_Donation);

DBMS\_OUTPUT.PUT\_LINE('DISPATCH DATE :'||v\_Dis);

DBMS\_OUTPUT.PUT\_LINE('DELIVERY DATE :'||v\_Delivery);

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER :'||v\_Daystodelivery);

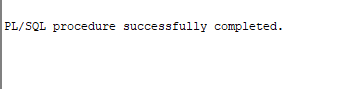
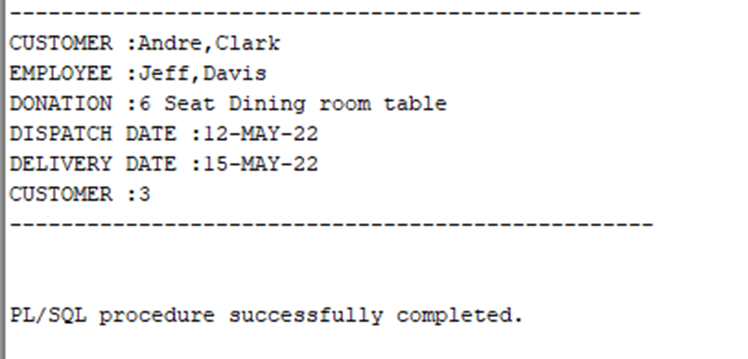
DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------');

END LOOP;

CLOSE info;

END;

/



--Q.6.2

---(a) DATE datatype: is used to store fixed-length datetimes, which include the time of day in seconds since midnight (Johnston,2000:08)

---(b) Character: Alphanumeric values that represent single characters or strings of characters (Johnston,2000:08).

# --Question7

--Q.7.1

SET SERVEROUTPUT ON;

DECLARE

v\_CustomerID customer.CUSTOMER\_ID%TYPE;

v\_DonationID donation.DONATION\_ID%TYPE;

v\_DonatorID donator.DONATOR\_ID%TYPE;

v\_DeliveryID delivery.DELIVERY\_ID%TYPE;

v\_ReturnID returns.RETURN\_ID%TYPE;

v\_ReturnReason returns.REASON%TYPE;

v\_Price INT;

CURSOR info

IS

SELECT customer.CUSTOMER\_ID,donation.DONATION\_ID,donator.DONATOR\_ID,returns.RETURN\_ID, delivery.DELIVERY\_ID, returns.REASON,

donation.PRICE \*0.2 AS DELIVERY\_PRICE

FROM customer

INNER JOIN invoice ON customer.CUSTOMER\_ID = invoice.CUSTOMER\_ID

INNER JOIN donation ON invoice.DONATION\_ID = donation.DONATION\_ID

INNER JOIN donator ON donator.DONATOR\_ID = donation.DONATOR\_ID

INNER JOIN delivery ON invoice.DELIVERY\_ID = delivery.DELIVERY\_ID

INNER JOIN returns ON returns.DONATION\_ID = donation.DONATION\_ID

WHERE delivery.DELIVERY\_ID = '512';

BEGIN

OPEN info;

LOOP

FETCH info INTO v\_CustomerID, v\_DonationID, v\_DonatorID, v\_DeliveryID, v\_ReturnID, v\_ReturnReason,v\_Price;

EXIT WHEN info%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER :'||v\_CustomerID);

DBMS\_OUTPUT.PUT\_LINE('DONATION ID:'||v\_DonationID);

DBMS\_OUTPUT.PUT\_LINE('DONATOR ID :'||v\_DonatorID);

DBMS\_OUTPUT.PUT\_LINE('DELIVERY ID: :'||v\_DeliveryID);

DBMS\_OUTPUT.PUT\_LINE('RETURN ID: :'||v\_ReturnID);

DBMS\_OUTPUT.PUT\_LINE('RETURN REASON: :'||v\_ReturnReason);

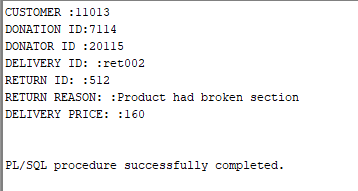
DBMS\_OUTPUT.PUT\_LINE('DELIVERY PRICE: :'||v\_Price);

END LOOP;

CLOSE info;

END;

/



--Q.7.2

--(a) Multiplication: is one of the four basic arithmetic operations, alongside addition, subtraction, and division (Interview, 2017:05).

--(b) Attribute indicator: property which let the user reference the data type and structure of a present item without having to repeat its definition (Interview, 2017:05).

# --Question8

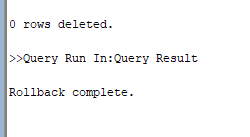
--Q.8.1

DELETE FROM employee

WHERE employee.EMPLOYEE\_ID = ' emp104';

SELECT \* FROM employee;

ROLLBACK;



--Q.8.2 I have used the rollblock statement for the delete transaction to be reversed.

# --Question9

--Q.9.1

CREATE TABLE FUNDING(

FUNDING\_ID INT NOT NULL,

FUNDER\_NAME VARCHAR2(100) NOT NULL,

FUNDING\_AMOUNT INT NOT NULL,

CONSTRAINT PK\_FUNDING PRIMARY KEY (FUNDING\_ID)

);

INSERT INTO FUNDING(FUNDER\_NAME,FUNDING\_AMOUNT)

VALUES('Charlotte Paul',40000);

CREATE SEQUENCE seq\_funding\_id;

CREATE TRIGGER Fund

BEFORE INSERT ON FUNDING

FOR EACH ROW

BEGIN

SELECT seq\_funding\_id.nextval

INTO :new.FUNDING\_ID

FROM dual;

END;

SELECT FUNDING\_ID,FUNDER\_NAME,FUNDING\_AMOUNT

FROM FUNDING

ORDER BY FUNDING\_ID;



--Q.9.2 Record id tool: creates a new column in the data and assigns a unique identifier, that increases sequentially, for each record in the data (Version, 2023:11).

**Reference list**

https://community.alteryx.com

[https://www.tutorialspoint.com](https://www.tutorialspoint.com/plsql/plsql_data_types.htm)

[https://docs.oracle.com](https://docs.oracle.com )