Part 1. Simple statements.

Question 1. Write the script to allow user to enter the name, date of birth and confirm citizenship. IF the citizenship is Kazakhstan show the name and date of birth of the user, if he is from Russia show the name and date of birth of the user and ask to enter passport number (20%).

SET serveroutput ON

DECLARE

v\_name varchar2(30):=&enter\_name;

v\_date varchar2(20):=&enter\_birth\_date;

v\_citizenship varchar2(20):=&enter\_citizenship;

v\_passport number(30);

BEGIN

IF v\_citizenship LIKE 'Kazakhstan' THEN

dbms\_output.put\_line(v\_name || 'was born in' || v\_date || ', has' ||

v\_citizenship || ' citizenship');

ELSIF v\_citizenship like 'Russia' THEN

v\_passport:= &Enter\_passport\_number;

dbms\_output.put\_line(v\_name || 'was born in' || v\_date || ', has' ||

v\_citizenship || ' citizenship, and passport No' || v\_passport);

ELSE dbms\_output.put\_line(‘Incorrect input’);

END IF;

END;

Question 2. Write a query to display all the orders which values are greater than the average order value for 10th October 2012 (30%).

Part 2. IF-ELSE statement.

Question 1. Retrieve selected customer’s account and include status of account as ‘Extra pending’ if the ‘pending balance’ is more than 10000 or ‘Minimum Pending’;

SET serveroutput ON

DECLARE

accStatus varchar2(30);

cur account%rowtype;

CURSOR lecture IS SELECT \* FROM account;

BEGIN

OPEN lecture();

loop

fetch lecture into cur;

exit when lecture%NOTFOUND;

if cur.PENDING\_BALANCE > 10000

THEN accStatus := 'Extra Pending';

else

accStatus := 'Minimum Pending';

END IF;

dbms\_output.put\_line(cur.account\_id || ' ' || cur.pending\_balance || ' ' || accStatus);

END LOOP;

CLOSE lecture;

END;

Question 2. Show the positions of the employees and define is he/she is ‘executor’ or ‘governor’ or ‘no position’ regarding his/her position. For example: President is ‘governor’

SET serveroutput ON

DECLARE

position varchar2(30);

cur EMPLOYEE%rowtype;

CURSOR lecture IS SELECT \* FROM employee;

BEGIN

OPEN lecture;

LOOP

fetch lecture into cur;

exit when lecture%NOTFOUND;

IF cur.dept\_id = 1

THEN position := 'No Position';

elsif cur.dept\_id = 2

THEN position := 'Executor';

elsif cur.dept\_id = 3

THEN position := 'Governor';

END IF;

dbms\_output.put\_line(cur.first\_name || ' ' || cur.dept\_id || ' ' || position);

END LOOP;

CLOSE lecture;

END;

Question 3. For this task consider that there are unwritten three types of account holders such as ‘affluent’ (from 50000), ‘sufficiency’ (from 10000 to 50000) and ‘limited’ (less than 10000); and show how many customers belong to these groups.

SET serveroutput ON;

DECLARE

status varchar(30);

sufficiency number:=0;

affluent number:=0;

limited\_v number:=0;

temp account%rowtype;

CURSOR lecture IS

SELECT \* FROM account;

BEGIN

OPEN lecture();

LOOP

FETCH lecture INTO temp;

EXIT WHEN lecture%notfound;

IF temp.pending\_balance>=50000 THEN

status:='affluent';

affluent:=affluent+1;

ELSIF temp.pending\_balance<50000 AND temp.pending\_balance>10000 THEN

status:='sufficiency';

sufficiency:=sufficiency+1;

ELSIF temp.pending\_balance<=10000 THEN

status:='limited';

limited\_v:=limited\_v+1;

END IF;

dbms\_output.put\_line(temp.account\_id||' money is: '||temp.pending\_balance||' status is: '|| status);

END LOOP;

dbms\_output.put\_line('limited number is: '||limited\_v||' sufficiency number is: '||sufficiency||' affluent number is: '||affluent);

CLOSE lecture;

END;

Question 4. Add bonus in 10% to the account holders who opened their account more than 15 years ago.

SET serveroutput ON;

DECLARE

status varchar(30);

bonus number;

opened date:='23-SEP-02';

temp account%rowtype;

CURSOR lecture IS

SELECT \* FROM account;

BEGIN

OPEN lecture();

LOOP

FETCH lecture INTO temp;

EXIT WHEN lecture%notfound;

IF temp.open\_date<opened

THEN bonus:=temp.pending\_balance\*1.1;

ELSIF temp.open\_date>opened THEN

bonus:=temp.pending\_balance\*1;

END IF;

dbms\_output.put\_line(temp.account\_id||' money is: '|| bonus);

END LOOP;

CLOSE lecture;

END;

Part 3. CASE statement.

Question 1. For this task consider that there are unwritten three types such as ‘affluent’ (from 50000), ‘sufficiency’ (from 10000 to 50000) and ‘limited’ (less than 10000); and show how many customers belong to these groups.

SET serveroutput ON;

DECLARE

status varchar(30);

sufficiency number:=0;

affluent number:=0;

limited\_v number:=0;

temp account%rowtype;

CURSOR lecture IS

SELECT \* FROM account;

BEGIN

OPEN lecture();

LOOP

FETCH lecture INTO temp;

EXIT WHEN lecture%notfound;

CASE WHEN temp.pending\_balance>=50000 THEN

status:='affluent';

affluent:=affluent+1;

WHEN temp.pending\_balance<50000 AND temp.pending\_balance>10000 THEN

status:='sufficiency';

sufficiency:=sufficiency+1;

WHEN temp.pending\_balance<=10000 THEN

status:='limited';

limited\_v:=limited\_v+1;

END CASE;

dbms\_output.put\_line(temp.account\_id||' money is: '||temp.pending\_balance||' status is: '|| status);

END LOOP;

dbms\_output.put\_line('limited number is: '||limited\_v||' sufficiency number is: '||sufficiency||' affluent number is: '||affluent);

CLOSE lecture;

END;

Question 2. Retrieve customers whose account was opened by employee, Paula Roberts, and also retrieve the bank’s product type which is served by this account. IF it is ‘Account’ update the close date of the account as Current date and change status to the ‘NOT ACTIVE’. IF it is not ‘Account’ keep it without change.

DECLARE

cursor curs is select \* from account where open\_emp\_id=10;

status varchar(30);

acc account%ROWTYPE;

BEGIN

OPEN curs;

LOOP

FETCH curs INTO acc;

EXIT WHEN curs%notfound;

IF acc.open\_date = acc.last\_activity\_date THEN

status := 'Not active';

ELSE

status := 'Active';

END IF;

dbms\_output.put\_line('id '|| acc.account\_id ||' status: '|| status);

END LOOP;

CLOSE curs;

END;

Question 3. There are two types of customers such as business and individuals. Please, find who is who and represent as ‘Business’ or ‘Individual’. Show the name of customer and individual or business.

set serveroutput on;

declare

status varchar(30);

temp customer%rowtype;

name\_v varchar(60);

cursor cus is

select \* from customer;

begin

open cus();

loop

fetch cus into temp;

exit when cus%notfound;

case when temp.cust\_type\_cd='I' then

status:='individual';

when temp.cust\_type\_cd='B' then

status:='business';

end case;

dbms\_output.put\_line(temp.cust\_id||' status is: '|| status);

end loop;

close cus;

end;

Question 4. Show the address and age of the customers and to which group they are belong; For Individual:

18-25 years old – Young Adult

26-40 years old – Adult

41-60 years old – Middle Aged Person

>60 – senior citizen

[the rest immature holders]

For Business:

Until 5 years – c

From 6 to 10 years – 2 Level of Maturity

From 11 to 15 years - 3 Level of Maturity

From 16 to 20 years - 4 Level of Maturity

More than 20 years - 5 Level of Maturity

set serveroutput on;

DECLARE

status varchar(30);

age varchar(30);

cus customer%rowtype;

ind individual%rowtype;

bus business%rowtype;

today date:='23-SEP-17';

res number;

name\_v varchar(60);

cursor curs1 is select \* from customer;

cursor curs2 is select \* from individual;

cursor curs3 is select \* from business;

BEGIN

open curs1();

open curs2();

open curs3();

LOOP

fetch curs1 into cus;

fetch curs2 into ind;

fetch curs3 into bus;

EXIT WHEN curs1%notfound;

CASE WHEN cus.cust\_type\_cd='I' THEN

status:='individual';

res:=months\_between(today,ind.birth\_date)/12;

CASE WHEN res>=18 AND res<=25 THEN

age:='Young Adult';

WHEN res>=26 AND res<=40 THEN

age:='Adult';

WHEN res>=41 AND res<=60 THEN

age:='Middle Aged Person';

WHEN res>=61 THEN

age:='senior citizen';

END CASE;

WHEN cus.cust\_type\_cd='B' THEN

status:='business';

res:=months\_between(today,bus.incorp\_date)/12;

CASE WHEN res>=6 AND res<=10 THEN

age:='2 Level of Maturity';

WHEN res>=11 AND res<=15 THEN

age:='3 Level of Maturity';

WHEN res>=16 AND res<=20 THEN

age:='4 Level of Maturity';

WHEN res>=21 THEN

age:='5 Level of Maturity';

END CASE;

END CASE;

dbms\_output.put\_line(cus.cust\_id||' status is: '|| status|| ' age is '|| age);

END LOOP;

close curs1;

close curs2;

close curs3;

END;