**CIS150/50 ORACLE**

**FINAL EXAM – FALL 2014**

**PART I:**

**Question #1:** Explain when a cursor is needed and the function it serves? Give a detailed description of what the cursor does, how you determine what goes in the cursor, how the information gets into the cursor, how you step through the information in the cursor, and how you know that you have completed processing all of the records in the cursor.

**Answer #1**

We need a cursor when we have group of records and we need to do some changes in each record individually.

Steps we follow to use a cursor.

1. Declare variable to store values from each row.
2. Declare a cursor with cursor name and put select query to get data from a table according to our needs.
3. We them open cursor by using OPEN command followed by cursor name in BEGIN portion.
4. Then we put FETCH to fetch one row at a time from cursor. The values we get are stored in the variables we defined.We need to put some loop like while or FOR loop so that we can get each record according to our conditions. We do some calculations or changes to the records in this part and and can update results in table or display the results.
5. Last step is to close the cursor using CLOSE command followed by the cursor name.

The loop we put take care that every record is checked.

**Question #2:** Explain the difference and similarities between a function, a procedure and a trigger. When is each one used?

**Answer #2**

**Function,** procedure and trigger are block of PL/SQL code that can be used with other codes to insert, update or delete from database tables. Both procedure and Functions are involved with the passing of arguments. The difference is that the function returns only one single value to the program but procedures can return multiple values.

Trigger like procedures and functions are block of PL/SQL code but it does not involve any arguments. A trigger is associated with some event like insert, update or delete.

**Question #3:** What is a trigger associated with, when does it "fire" and why would the developer choose to use one?

**ANSWER #3**

A trigger is associated with DML statement such as Insert, update or delete. It can fire before or after DML statement runs. The developer choose to use trigger to get notified or keep check on data if some changes are made in a table.

**Question #4:** Explain the differences in logic and execution between a WHILE loop involving a cursor, a LOOP with an EXIT involving a cursor and the special FOR loop involving a cursor.

Answer #4

**The while loop** will continue to execute while there is still data in cursor. So before we start the loop we need to fetch the data from cursor. This is tested with %FOUND. After the changes being made to the data next record is fetched at the end of the loop using FETCH. Loop will end when no data is fetched.

**A loop with an EXIT** will work differently. Data will be fetched after the loop starts and then tested with %not found . Loop will end if no data is found.

When using **FOR LOOP** you need not declare a record or variables to store the cursor values, need not open, fetch and close the cursor. These functions are accomplished by the FOR LOOP automatically. You just need to declare the cursor. The syntax is FOR followed by the record name IN followed by the cursor name and the word LOOP.

**Question #5:** How does a view work so that the data viewed is always current? Explain**.**

Views carries the information from base tables but information is not stored in them. All the data in view is derived from the base table so if we have made changes in a base table they are reflected in the view.

**Question #6**: With subqueries, what code is needed to allow the inner select to return multiple rows/records? Explain

Use of **IN** in outer query allows inner select to return multiple rows.

**Question #7:** Explain the concept of a primary and foreign key and how they can be used in a relational database.

Primary key allows you to put one and only record to the table which prevents from duplication of records.

Foreign Key links the table with the primary key of other table. It means that records in the table with foreign key depends on the table with primary key. It can only contain records which are in other table with primary key. They make a relation between two tables.

**Question #8:** Assume you are populating the following set of tables the first time. The tables are being created with information from your paper files so you have data to enter into all four tables. Underline and bold means a primary key. First determine what would be setup as foreign keys and then explain the order that you would use in populating the tables.

Library:

Customer (acctno, name, address)

Checkout Header (acctno, date)

Checkout Detail (acctno, date, ISBN #, due date)

Books(ISBN#, title, primary author, date published…)

**Answer #8**

Acctno in customer table is primary key.

Acctno in checkout header is foreign key.

Acctno in checkout Detail is foreign key.

ISBN# is foreign key in checkout detail table and primary key in Books table.

1. First populate **customer** table.
2. Populate **Books** table.
3. Populate **checkout detail** table.
4. Populate **checkout header**.

**QUESTION #9**

create table student

( studentid varchar2(2),

studentname varchar(20),

MAJORCODE varchar2(2),

optioncode varchar2(2),

numcredits number(4,2),

formsub varchar2(2),

gpa number(4,2),

gradstatus varchar2(2));

create table gradstatus

(studentid varchar2(2),

honor\_level varchar2(30));

insert into student

values('01', 'Elise','CI','NT', 46, 'Y', 3.4, 'N');

insert into student

values('02', 'Nick','BU','PA', 41, 'N', 3.1, 'N');

insert into student

values('03', 'Keema','CI','NT', 50, 'Y', 3.8, 'Y');

insert into student

values('04', 'Garvy','CI','WD', 47, 'Y', 3.6, 'Y');

insert into student

values('05', 'Aryana','BU','NT', 45, 'N', 3.4, 'N');

insert into student

values('06', 'Norman','CI','PA', 49, 'Y', 3.9, 'Y');

insert into student

values('07', 'Dev','BU','NT', 70, 'Y', 4.1, 'Y');

insert into student

values('08', 'Arnav','CI','WD', 60, 'Y', 4.0, 'Y');

insert into student

values('09', 'Varun','CI','NT', 68, 'Y', 4.4, 'Y');

insert into student

values('10', 'Ella','BU','NT', 48, 'N', 3.4, 'N');

SQL> select \* from student;

ST STUDENTNAME MA OP NUMCREDITS FO GPA GR

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

01 Elise CI NT 43 Y 3.1 N

02 Nick BU PA 41 N 3.1 N

03 Keema CI NT 50 Y 3.8 Y

04 Garvy CI WD 47 N 3.6 Y

05 Aryana BU NT 45 N 3.15 N

06 Norman CI PA 40 N 3 Y

07 Dev BU NT 70 Y 4.1 Y

08 Arnav CI WD 60 Y 4 Y

09 Varun CI NT 68 Y 4.4 Y

10 Ella BU NT 48 N 3.4 N

10 rows selected.

SQL> select \* from **gradstatus;**

ST HONOR\_LEVEL

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

01

02

03

04

05

06

07

08

09

10

**PL/SQL Program**

set verify off

set serveroutput on

declare

stu\_id student.studentid%type ;

stu\_maj student.majorcode%type;

stu\_gpa student.gpa%type;

num\_credits student.numcredits%type;

stu\_intent student.formsub%type;

stu\_level gradstatus.honor\_level%type;

CURSOR stu\_cursor is

select studentid, majorcode, gpa, numcredits, formsub

from student where majorcode ='CI' and gpa>=3.2 and numcredits >45and formsub ='Y';

begin

OPEN stu\_cursor;

FETCH stu\_cursor into stu\_id, stu\_maj, stu\_gpa, num\_credits, stu\_intent;

WHILE stu\_cursor %FOUND LOOP

IF ( stu\_gpa >= 3.2 and stu\_gpa <= 3.49)THEN

stu\_level := 'Cum Laude';

ELSIF ( stu\_gpa >= 3.5 and stu\_gpa <= 3.79)THEN

stu\_level := 'Magna Cum Laude';

ELSIF stu\_gpa >= 3.8 THEN

stu\_level := 'Summa Cum Laude';

END IF;

UPDATE gradstatus

set honor\_level = stu\_level

where studentid = stu\_id;

FETCH stu\_cursor into stu\_id, stu\_maj, stu\_gpa, num\_credits, stu\_intent;

END LOOP;

CLOSE stu\_cursor;

end;

/

set serveroutput off

set verify on

**QUESTION #10:**

SQL> **@final9**

PL/SQL procedure successfully completed.

SQL> select \* from **gradstatus;**

**ST HONOR\_LEVEL**

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

01 Cum Laude

02

03 Summa Cum Laude

04 Magna Cum Laude

05

06

07

08 Summa Cum Laude

09 Summa Cum Laude

10

10 rows selected.

**QUESTION #11**

create table **major**

(majorcode varchar2(2),

majorname varchar2(15),

advisor varchar2(15));

create table **statistics**

(majorcode varchar2(2),

numstudents number(4),

numcredits number(5,2));

insert into major

values( 'CI', 'Computer', 'Smith');

insert into major

values( 'BU', 'Business', 'Henry');

SQL> select \* from major;

MA MAJORNAME ADVISOR

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

CI Computer Smith

BU Business Henry

**PL/SQL Program**

set verify off

set serveroutput on

DECLARE

stu\_major student.majorcode%type;

num\_students statistics.numstudents%type;

num\_credits statistics.numcredits%type;

**CURSOR** major\_cursor is select majorcode from major

order by majorcode;

**CURSOR** stu\_cursor is

select count(\*), sum(numcredits) from student

where majorcode = stu\_major

group by majorcode order by majorcode;

BEGIN

OPEN major\_cursor;

FETCH major\_cursor into stu\_major;

WHILE major\_cursor %FOUND LOOP

OPEN stu\_cursor;

FETCH stu\_cursor into num\_students, num\_credits;

insert into statistics

values(stu\_major,num\_students,num\_credits);

CLOSE stu\_cursor;

FETCH major\_cursor into stu\_major;

END LOOP;

CLOSE major\_cursor;

end;

/

set serveroutput off

set verify on

**QUESTION #12:**

SQL**> @final12**

PL/SQL procedure successfully completed

SQL> **select \* from statistics;**

MA NUMSTUDENTS NUMCREDITS

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

BU 4 204

CI 6 308

**PART II:**

**PROBLEM #1**

SQL> SELECT i.itemno, i.itemname, g.salesgoal

2 FROM inventory i

3 WHERE i.itemno IN

4 (SELECT g.itemno

5 FROM goalstable g

6 WHERE g.salesgoal > 50000);

This code is not valid . Table goalstable is missing from outer select.

SQL> SELECT i.itemno, i.itemname, g.salesgoal

2 FROM inventory i, **goalstable g (**corrected)

3 WHERE i.itemno IN

4 (SELECT g.itemno

5 FROM goalstable g

6 WHERE g.salesgoal > 50000);

In this code inner query will select multiple records of itemno with g.salesgoal >50000 and outer query will select only records wit itemno selected by inner query.

**PROBLEM #2**

SQL> UPDATE inventory

2 SET cost = 56.78 price = 75.99

3 where itemno = '22222';

This code is not valid. Comma is needed b/w set cost and price.

SQL> UPDATE inventory

2 SET cost = 56.78, price = 75.99

3 where itemno = '22222';

This code will update cost and price from table inventory with itemno 22222.

**PROBLEM #3**

SQL> alter table insurance\_account

2 insert (amtdue\_balance number(6,2));

This code is not valid . Use **MODIFY** in place of insert.

SQL> alter table insurance\_account

**MODIFY** (amtdue\_balance number(6,2));

This will change the size of field amtdue\_balance to number(6,2).

**PROBLEM #4**

SQL> CREATE VIEW sales\_view AS

2 SELECT \* FROM sales\_table

3 WHERE sales\_amt < 250000

4 WITH CHECK OPTION CONSTRAINT saleschk\_ck;

In this code a view sales\_view is created on table sales\_table and it will take only records with sale\_amt less than 250000.

WITH CHECK OPTION CONSTRAINT saleschk\_ck this constraint will not allow any sales\_amt in this view which is > 250000.

If you try to **update** any record in this view where sales\_amt is >= 250000 it will be rejected because we have a **CHECK OPTION CONSTRAINT** on this view**.**

**PROBLEM #5**

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

dbms\_output.put\_line('Not a unique response');

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Not a valid data entry');

WHEN OTHERS THEN

dbms\_output.put\_line('Undefined error occured');

**EXPLANATION:**

WHEN TOO\_MANY\_ROWS THEN

dbms\_output.put\_line('Not a unique response');

This error will appear when multiples rows are selected.

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Not a valid data entry');

This error will appear when no record is found.

WHEN OTHERS THEN

dbms\_output.put\_line('Undefined error occured');

This error will appear when there is any unspecified error in the program.

PL/SQL allows programmer to handle different kinds of errors they can be oracle errors or exceptions raised by the programmer according to the needs of a particular program.

The code above uses some of the predefined oracle errors. Errors are trapped in the EXCEPTION portion of the program and are written with the WHEN clause.

For non predefined errors EXCEPTION is first declared in declare portion and defined in BEGIN and then place in EXCEPTION portion.

**PROBLEM #6**

SQL> select studentid, majorcode, majorname, gpa, credits

2 from student, major;

This code will select the mentioned fields from student table and major table.

**PROBLEM #7**

**SQL> create table invenbk from inventory1;**

This code is not valid. Correct code is **create table invenbk as select \* from inventory1;**

This code will create a new table invenbk from table inventory1.

**SQL> delete \* from inventory1**;

This code will delete all records from inventory1 table.

**PROBLEM #8**

SQL> SELECT idno, stuname, stugpa, stumajor

2 FROM student

3 WHERE stumajor IN

4 (SELECT stugpa

5 FROM student

6 WHERE stugpa > 3.5);

In this code first inner query will select all stugpa from student table where stugpa is >3.5. The outer query will then select records from student table where stumajor in( stugpa selected by inner query).

**PROBLEM #9**

SQL> edit

Wrote file afiedt.buf

1 select itemno, itemname, dept, price, newprice

2 decode(dept, 'first', price \* 1.1,

3 'misc', price \* 1.5,

4 'other', price \* 1.3,

5 price)

6 newprice

7\* from inventory1

This code tells us that there was an error in the code and we use edit to fix the error in editor.

**ERROR:** Need comma after newprice in select command.

**Correction:**

SQL> edit

Wrote file afiedt.buf

1 select itemno, itemname, dept, price**, newprice,(comma was missing after newprice )**

2 decode(dept, 'first', price \* 1.1,

3 'misc', price \* 1.5,

4 'other', price \* 1.3,

5 price)

6 newprice

7\* from inventory1;

This code will decode the dept field. If dept is first price is multiplied by 1.1, if dept is misc price is multiplied by 1.5 and if dept is other price is multiplied by 1.3 and in any other case price nothing is done and all the calculated results are stored in newprice.

**PROBLEM #10:**

SQL> select empno, ename, emp.deptno, dname

2 from emp join dept

3 on emp.deptno = dept.deptno;

This code will select empno,ename,emp.deptno and dname from emp table and dept table. ON with JOIN will make a relationship b/w two tables on deptno. Both tables have common field deptno. The records with deptno in both tables will be displayed.

select empno, ename, emp.deptno, dname

from emp, dept

where emp.deptno = dept.deptno;

**PROBLEM #11**

SQL> select inveno, invenname, vendorno, vendorname

from inven join vendor

using (vendorno);

This code will select mentioned fields from inven table and vendor table. USING vendorno will join two tables.

**Code using where:**

SQL> select inveno, invenname, invent.vendorno, vendorname

from inven, vendor

where inven.vendorno = vendor.vendorno;

**PROBLEM #12:**

SQL> select inveno, invenname, inven.vendorno, vendorname

2 from inven natural join vendor;

This code will give error because with natural join table name is not used with field name.

**Correct code:**

SQL> select inveno, invenname, vendorno, vendorname

from inven natural join vendor;

**PROBLEM #13**

SQL> select \* from payroll where proj in

2 (select proj from payroll where budget >=500000)

3 and projclass in

4 (select projclass from payroll where dept > 10);

This code will select records from payroll table where the proj is selected from first inner query(all projects where budget >= 500000) and projclass is selected from second inner query( all projclass of dept >10). The result will display the records where selected proj matches the record from selected projclass .

**PROBLEM #14:**

SQL> SELECT inv.itemno, inv.price, recpt.recept

2 FROM inventory inv, receipts recpt

3 WHERE inv.itemno(+) = recpt.itemno;

This code will select records from inventory table and receipts table where itemno matches in both tables but with(+) sign it will also display the records from inventory table where itemno is not in receipt table.

**PROBLEM #15**

1 SELECT itemno, itemname, dept, manager, onhand, onorder, reorder, price, cost

2 FROM inventory

3 WHERE cost IN

4 (SELECT cost

5 FROM inventory

6 WHERE onhand > 100)

7 AND dept =

8 (SELECT manager

9 FROM department

10 WHERE dept = 'books')

Assuming that we have two tables inventory and department table we need to join both tables with the common field **dept** and there is problem in second inner query also DEPT cannot be equal to manager.I assume department table contains two fields dept and manager.

**CORRECTION:**

1 SELECT itemno, itemname, inventory.dept, manager, onhand, onorder, reorder, price, cost

2 **FROM inventory,department**

**3 Where inventory.dept = department.dept**

4 AND cost IN

5 (SELECT cost

6 FROM inventory

7 WHERE onhand > 100)

8 AND dept =

9 (SELECT **dept**

10 FROM department

11 WHERE dept = 'books')

This code will select itemno, itemname, inventory.dept, manager, onhand, onorder, reorder, price, cost from inventory and department table where dept is in both tables. First inner query will select the cost(multiple records) from inventory where onhand >10 and second inner query will select the Book department.

This code is copied from SQL PROMPT.