PL/SQL Introduction

1. Count the number of Locations

Grade settings: Maximum grade: 25
Run: Yes **Evaluate**: Yes
Automatic grade: Yes
Write a PL/SQL to count the locations, where the various departments are located and print the number.

Table name : Department

Column name	Data type	Constraints
DEPARTMENT_ID	NUMBER(5)	PK
DEPARTMENT_NAME	VARCHAR2(25)	NOT NULL
LOCATION_ID	VARCHAR2(15)	

Sample Output:
Number of Locations are : xxx

Automatic evaluation[-]

Proposed grade: 25 / 25
Result Description
[-]**Summary of tests**

```
+-----------------------------+
| 2 tests run / 2 test passed  |
+-----------------------------+
```

count.sql

```
1 set serveroutput on;
2 DECLARE
3   cnt number;
4 BEGIN
5  select count(DISTINCT(LOCATION_ID))into cnt FROM DEPARTMENT;
6  DBMS_OUTPUT.PUT_LINE('Number of Locations are :'||cnt);
7 END;
8 /
```

Grade

Reviewed on Wednesday, 1 December 2021, 12:46 AM by Automatic grade
Grade 25 / 25
Assessment report
[+]**Summary of tests**

Cursors & Exception Handling

1. Count the number of Records

Grade settings: Maximum grade: 25
Run: Yes **Evaluate**: Yes
Automatic grade: Yes
Write a program that gives all employees in Mechanical department,

(i) Update the salary with 15% pay increase.

(ii) Display a message displaying how many Employees were awarded the increase. If no Employees found then print the message 'No Records found.'

EMPLOYEE:

Column name	Data type	Constraints
EMP_ID	NUMBER(5)	PK
EMP_NAME	VARCHAR2(25)	NOT NULL
SALARY	NUMBER(10,2)	
DEPT	VARCHAR2(25)	

EMP_ID	EMP_NAME	SALARY	DEPT
101	Tom	54000	MECH
102	William	43000	CSE
103	John	34560	MECH
104	Smith	56000	CSE
105	Steve	23450	IT

Sample Output:
2 Employees got increment.

Automatic evaluation[-]

Proposed grade: 25 / 25
Result Description
[-]**Summary of tests**

```
+-----------------------------+
| 2 tests run / 2 test passed  |
+-----------------------------+
```

rec.sql

```
1 set serveroutput on;
2 declare
3 tot_rec number;
4 begin
5 update employee set salary = (salary*0.15)+salary where dept='MECH';
```

```
 6 tot_rec:= sql%rowcount;
 7 if sql%found then
 8 dbms_output.put_line(tot_rec || ' Employees got increment.');
 9 else
10 dbms_output.put_line('No Records found.');
11 end if;
12 end;
13 /
14
```

Grade

Reviewed on Wednesday, 1 December 2021, 12:46 AM by Automatic grade
Grade 25 / 25
Assessment report
[+]**Summary of tests**

---

2. Copy Records

Grade settings: Maximum grade: 25
Run: Yes **Evaluate**: Yes
Automatic grade: Yes
Create a pl/sql program that should copy the department id, name and location id of all departments that located in 'CBE' (location_id should start with 'CBE') to a table called 'DEPARTMENT_CBE'. Do not use a cursor FOR loop. Display how many rows were copied else display the message 'No Records found.'.

DEPARTMENT:

Column name	Data type	Constraints
DEPARTMENT_ID	NUMBER(5)	PK
DEPARTMENT_NAME	VARCHAR2(25)	NOT NULL
LOCATION_ID	VARCHAR2(15)	

Note:
Assume, 'DEPARTMENT' table is already been created

Sample Output:
xx Record(s) got copied.

Automatic evaluation[-]

Proposed grade: 25 / 25
Result Description
[-]**Summary of tests**

```
+-----------------------------+
| 2 tests run / 2 test passed  |
+-----------------------------+
```

copy.sql

```
 1 SET SERVEROUTPUT ON;
 2 DECLARE
 3    v_count number(5) := 0;
 4 BEGIN
```

```
 5        insert into DEPARTMENT_CBE SELECT * FROM DEPARTMENT where LOCATION_ID LIKE
'CBE%' ;
 6    IF SQL%NOTFOUND THEN
 7                    DBMS_OUTPUT.PUT_LINE('No Records found.');
 8    ELSIF SQL%FOUND THEN
 9            v_count := SQL%ROWCOUNT;
10            DBMS_OUTPUT.PUT_LINE(V_COUNT || ' Record(s) got copied.');
11        END IF;
12 END;
13 /
```

Grade

Reviewed on Thursday, 2 December 2021, 6:50 PM by Automatic grade
Grade 25 / 25
Assessment report
[+]**Summary of tests**

Procedures & Functions

1. Remove Employee records - Procedures

Grade settings: Maximum grade: 25
Run: Yes **Evaluate**: Yes
Automatic grade: Yes
Create a procedure that deletes rows from the Employee table. It should accept 1 parameter, department name; only delete the employee records belonging to that department. Display how many employees were deleted else raise "DeptNotFoundException" and print the message 'No Records found.'.

EMPLOYEE:

Column name	Data type	Constraints
EMP_ID	NUMBER(5)	PK
EMP_NAME	VARCHAR2(25)	NOT NULL
SALARY	NUMBER(10,2)	
DEPT	VARCHAR2(25)	

EMP_ID	EMP_NAME	SALARY	DEPT
101	Tom	54000	MECH
102	William	43000	CSE
103	John	34560	MECH
104	Smith	56000	CSE
105	Steve	23450	IT

Functional Requirements:

PROCEDURE DELETE_EMPLOYEE(v_dept IN EMPLOYEE.dept%TYPE)

Sample Output:
2 Employee record(s) got deleted.

Automatic evaluation[-]

Proposed grade: 25 / 25
Result Description
[-]**Summary of tests**

```
+-----------------------------+
| 2 tests run / 2 test passed  |
+-----------------------------+
```

remove.sql

```
1 CREATE PROCEDURE DELETE_EMPLOYEE(
2  v_dept IN EMPLOYEE.dept%TYPE)
3 IS
4   v_count number := 0;
5   DeptNotFoundException EXCEPTION;
6 BEGIN
7     delete from employee where dept=v_dept;
8     IF SQL%NOTFOUND THEN RAISE DeptNotFoundException;
9     ELSIF SQL%FOUND THEN
10            v_count := SQL%ROWCOUNT;
11            DBMS_OUTPUT.PUT_LINE(V_COUNT || ' Employee record(s) got deleted.');
12         END IF;
13         EXCEPTION
14            WHEN DeptNotFoundException THEN
15               DBMS_OUTPUT.PUT_LINE('No Records found.');
16 END DELETE_EMPLOYEE;
17 /
```

Grade

Reviewed on Thursday, 2 December 2021, 6:50 PM by Automatic grade
Grade 25 / 25
Assessment report
[+]**Summary of tests**

2. Concatenate Employee names - Functions

Grade settings: Maximum grade: 25
Run: Yes **Evaluate**: Yes
Automatic grade: Yes
Write a PL/SQL function to concatenate first name and last name of an employee. Pass employee id as an input to the function CONCAT_NAME.
EMPLOYEE:

Column name	Data type	Constraints
EMP_ID	NUMBER(5)	PK
FIRST_NAME	VARCHAR2(25)	NOT NULL
LAST_NAME	VARCHAR2(25)	
DEPT	VARCHAR2(25)	

Functional Requirement:

FUNCTION CONCAT_NAME(v_id employee.emp_id%type)

---

3. Withdraw Amount - Procedures

Grade settings: Maximum grade: 25
Run: Yes **Evaluate**: Yes
Automatic grade: Yes
Write a PL/SQL procedure to perform withdraw operation that only permits a withdrawal, if there is sufficient funds in the account then update the Account table and print the message, 'Transaction successful.' else print, 'Insufficient Amount.' . The procedure should take Account_id and withdrawal amount as input.

Account:

ACCNO	NUMBER	PK
CUSTOMER_NAME	VARCHAR2(30)	
BALANCE	NUMBER(15,2)	

12345Williams23455.6
23456Robert 43221
34521John 23449

Functional Requirement:
procedure withdraw(ano number , amt number)

Sample input:
withdraw(12345, 2000);

Sample output:

Transaction successful.

Automatic evaluation[-]

Proposed grade: 25 / 25
Result Description
[-]**Summary of tests**

```
+----------------------------+
| 2 tests run / 2 test passed  |
```

```
+-----------------------------+
```

account.sql

```
 1 create or replace procedure withdraw(ano number , amt number)
 2 is
 3   v_bal number;
 4 begin
 5  select balance into v_bal from account where accno = ano;
 6  if v_bal > amt then
 7     update account set balance = balance - amt where accno = ano;
 8     DBMS_OUTPUT.PUT_LINE('Transaction successful.');
 9  else
10     DBMS_OUTPUT.PUT_LINE('Insufficient Amount.');
11   end if;
12 end;
13 /
```

Grade

Reviewed on Wednesday, 1 December 2021, 12:46 AM by Automatic grade
Grade 25 / 25
Assessment report
[+]**Summary of tests**

Packages & Triggers

1. Delete a Record - Triggers

Grade settings: Maximum grade: 25
Run: Yes **Evaluate**: Yes
Automatic grade: Yes
An HR system has an Employee table that holds a row for each employee within the company. Each record in the table has a employee id, employee name and manager column, that holds the id for the employee's manager. Write a trigger so that when an employee record is deleted, the record details need to be inserted into an table called Employee_archive along with the deleted date.

EMPLOYEE:

EMPID	NUMBER	PRIMARY KEY
EMPNAME	VARCHAR2(25)	
MANAGERID	NUMBER	

EMPLOYEE_ARCHIVE:

EMPID	NUMBER	PRIMARY KEY
EMPNAME	VARCHAR2(25)	
MANAGERID	NUMBER	
DELETED_DATE	DATE	

Automatic evaluation[-]

Proposed grade: 25 / 25

Result Description

[-]**Summary of tests**

```
+-----------------------------+
| 2 tests run / 2 test passed  |
+-----------------------------+
```

mgr.sql

```
1 Create or replace trigger del_emp before delete on EMPLOYEE for each row
2 begin
3 insert into EMPLOYEE_ARCHIVE values(:old.EMPID,:old.EMPNAME,:old.MANAGERID,sysdate);
4 end;
5 /
```

Grade

Reviewed on Wednesday, 1 December 2021, 12:46 AM by Automatic grade

Grade 25 / 25

Assessment report

[-]**Summary of tests**

```
+-----------------------------+
| 2 tests run / 2 test passed  |
+-----------------------------+
```

2. Inventory alert

Grade settings: Maximum grade: 25
Run: Yes **Evaluate**: Yes
Automatic grade: Yes
Create a PL/SQL Trigger to display the message “Re-Order Level reached for the item - <item-name>”, whenever a item quantity reaches 10 and below while updating or inserting an item.

Stock:

ITEMID	NUMBER	PRIMARY KEY
DESCRIPTION	VARCHAR2(30)	
QUANTITY	NUMBER	

Sample output:
Re-order Level reached for the item - Keyboard

Automatic evaluation[-]

Proposed grade: 25 / 25

Result Description

[-]**Summary of tests**

```
+-----------------------------+
```

```
| 2 tests run / 2 test passed  |
+------------------------------+
```

inventory.sql

```
1 create or replace trigger t_Inventory after insert or update on stock for each row
2 when (new.quantity <= 10)
3 begin
4  dbms_output.put_line('Re-order Level reached for the item - '||:new.description);
5 end;
6 /
7
```

Grade

Reviewed on Thursday, 2 December 2021, 6:50 PM by Automatic grade
Grade 25 / 25
Assessment report
[-]**Summary of tests**

```
+------------------------------+
| 2 tests run / 2 test passed  |
+------------------------------+
```