Q1)

CREATE OR REPLACE PROCEDURE modify\_sal(p\_deptID NUMBER) IS

V\_SALAVG NUMBER(20);

v\_deptId NUMBER(20);

V\_COUNTER NUMBER(20) :=0;

v\_emp NUMBER(20);

v\_salTot NUMBER(20);

emptyEmp EXCEPTION;

BEGIN

SELECT DEPARTMENT\_ID INTO v\_deptId FROM DEPARTMENTS WHERE DEPARTMENT\_ID = p\_deptID;

SELECT COUNT(\*) INTO v\_emp FROM EMPLOYEES WHERE DEPARTMENT\_ID = p\_deptID;

IF v\_emp = 0 THEN

RAISE EMPTYEMP;

END IF;

SELECT AVG(SALARY) INTO v\_salAvg FROM EMPLOYEES WHERE DEPARTMENT\_ID = p\_deptID;

FOR i IN (SELECT \* FROM EMPLOYEES WHERE DEPARTMENT\_ID = p\_deptID FOR UPDATE NOWAIT)

LOOP

IF i.SALARY < v\_salAvg THEN

v\_salTot := v\_salAvg - i.SALARY;

UPDATE EMPLOYEES SET SALARY = v\_salAvg WHERE EMPLOYEE\_ID = i.EMPLOYEE\_ID;

v\_counter := v\_counter + 1;

DBMS\_OUTPUT.PUT\_LINE('Employee ' || I.FIRST\_NAME || ' ' || I.LAST\_NAME || ' just got an increase of $' || V\_SALTOT);

-- ELSE DBMS\_OUTPUT.PUT\_LINE('no matching CASE found');

END IF;

END LOOP;

CASE

WHEN v\_counter > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Total # of employees who received salary increase is: ' || V\_COUNTER);

ELSE

DBMS\_OUTPUT.PUT\_LINE('No salary was modified in Department: ' || P\_DEPTID);

END CASE;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('This Department Id is invalid: ' || p\_deptID);

WHEN emptyEmp THEN

DBMS\_OUTPUT.PUT\_LINE('This Department is EMPTY: ' || p\_deptID);

END modify\_sal;

Q2)

CREATE OR REPLACE FUNCTION total\_cost(f\_studid NUMBER)

RETURN NUMBER IS

v\_total NUMBER;

v\_numenroll NUMBER;

v\_test NUMBER;

BEGIN

--To check if Student Exists

SELECT STUDENT\_ID INTO v\_test FROM STUDENT WHERE STUDENT\_ID = f\_studid;

--To check if student is enrolled at anything

SELECT COUNT(SECTION\_ID) INTO v\_numenroll FROM ENROLLMENT WHERE STUDENT\_ID = f\_studid;

IF(v\_numenroll = 0) THEN

RETURN 0;

END IF;

--Gets total cost for student

SELECT SUM(COST) INTO v\_total FROM COURSE WHERE COURSE\_NO IN(SELECT COURSE\_NO FROM SECTION WHERE SECTION\_ID IN (SELECT SECTION\_ID FROM ENROLLMENT WHERE STUDENT\_ID = f\_studid));

RETURN v\_total;

--Will trigger exception if Student id doesn't exist

EXCEPTION WHEN NO\_DATA\_FOUND THEN

RETURN -1;

END total\_cost;

VARIABLE b\_costtot NUMBER

EXECUTE :b\_costtot := total\_cost(494);

print b\_costtot

Q3)

CREATE OR REPLACE PACKAGE My\_pack AS

PROCEDURE modify\_sal(p\_deptID NUMBER);

FUNCTION total\_cost(f\_studid NUMBER)

RETURN NUMBER;

END My\_pack ;

CREATE OR REPLACE PACKAGE BODY My\_pack AS

PROCEDURE modify\_sal(p\_deptID NUMBER) IS

V\_SALAVG NUMBER(20);

v\_deptId NUMBER(20);

V\_COUNTER NUMBER(20) :=0;

v\_emp NUMBER(20);

v\_salTot NUMBER(20);

emptyEmp EXCEPTION;

BEGIN

SELECT DEPARTMENT\_ID INTO v\_deptId FROM DEPARTMENTS WHERE DEPARTMENT\_ID = p\_deptID;

SELECT COUNT(\*) INTO v\_emp FROM EMPLOYEES WHERE DEPARTMENT\_ID = p\_deptID;

IF v\_emp = 0 THEN

RAISE EMPTYEMP;

END IF;

SELECT AVG(SALARY) INTO v\_salAvg FROM EMPLOYEES WHERE DEPARTMENT\_ID = p\_deptID;

FOR i IN (SELECT \* FROM EMPLOYEES WHERE DEPARTMENT\_ID = p\_deptID FOR UPDATE NOWAIT)

LOOP

IF i.SALARY < v\_salAvg THEN

v\_salTot := v\_salAvg - i.SALARY;

UPDATE EMPLOYEES SET SALARY = v\_salAvg WHERE EMPLOYEE\_ID = i.EMPLOYEE\_ID;

v\_counter := v\_counter + 1;

DBMS\_OUTPUT.PUT\_LINE('Employee ' || I.FIRST\_NAME || ' ' || I.LAST\_NAME || ' just got an increase of $' || V\_SALTOT);

-- ELSE DBMS\_OUTPUT.PUT\_LINE('no matching CASE found');

END IF;

END LOOP;

CASE

WHEN v\_counter > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Total # of employees who received salary increase is: ' || V\_COUNTER);

ELSE

DBMS\_OUTPUT.PUT\_LINE('No salary was modified in Department: ' || P\_DEPTID);

END CASE;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('This Department Id is invalid: ' || p\_deptID);

WHEN emptyEmp THEN

DBMS\_OUTPUT.PUT\_LINE('This Department is EMPTY: ' || p\_deptID);

END modify\_sal;

FUNCTION total\_cost(f\_studid NUMBER)

RETURN NUMBER IS

v\_total NUMBER;

v\_numenroll NUMBER;

v\_test NUMBER;

BEGIN

--To check if Student Exists

SELECT STUDENT\_ID INTO v\_test FROM STUDENT WHERE STUDENT\_ID = f\_studid;

--To check if student is enrolled at anything

SELECT COUNT(SECTION\_ID) INTO v\_numenroll FROM ENROLLMENT WHERE STUDENT\_ID = f\_studid;

IF(v\_numenroll = 0) THEN

RETURN 0;

END IF;

--Gets total cost for student

SELECT SUM(COST) INTO v\_total FROM COURSE WHERE COURSE\_NO IN(SELECT COURSE\_NO FROM SECTION WHERE SECTION\_ID IN (SELECT SECTION\_ID FROM ENROLLMENT WHERE STUDENT\_ID = f\_studid));

RETURN v\_total;

--Will trigger exception if Student id doesn't exist

EXCEPTION WHEN NO\_DATA\_FOUND THEN

RETURN -1;

END total\_cost;

END My\_pack ;

Q4)

CREATE OR REPLACE PACKAGE My\_pack AS

PROCEDURE modify\_sal(p\_deptID NUMBER);

FUNCTION total\_cost(f\_studid NUMBER)

RETURN NUMBER;

FUNCTION total\_cost(f\_fname VARCHAR2,f\_lname VARCHAR2)

RETURN NUMBER;

FUNCTION total\_cost(f\_zip VARCHAR2)

RETURN NUMBER;

END My\_pack ;

CREATE OR REPLACE PACKAGE BODY My\_pack AS

PROCEDURE modify\_sal(p\_deptID NUMBER) IS

V\_SALAVG NUMBER(20);

v\_deptId NUMBER(20);

V\_COUNTER NUMBER(20) :=0;

v\_emp NUMBER(20);

v\_salTot NUMBER(20);

emptyEmp EXCEPTION;

BEGIN

SELECT DEPARTMENT\_ID INTO v\_deptId FROM DEPARTMENTS WHERE DEPARTMENT\_ID = p\_deptID;

SELECT COUNT(\*) INTO v\_emp FROM EMPLOYEES WHERE DEPARTMENT\_ID = p\_deptID;

IF v\_emp = 0 THEN

RAISE EMPTYEMP;

END IF;

SELECT AVG(SALARY) INTO v\_salAvg FROM EMPLOYEES WHERE DEPARTMENT\_ID = p\_deptID;

FOR i IN (SELECT \* FROM EMPLOYEES WHERE DEPARTMENT\_ID = p\_deptID FOR UPDATE NOWAIT)

LOOP

IF i.SALARY < v\_salAvg THEN

v\_salTot := v\_salAvg - i.SALARY;

UPDATE EMPLOYEES SET SALARY = v\_salAvg WHERE EMPLOYEE\_ID = i.EMPLOYEE\_ID;

v\_counter := v\_counter + 1;

DBMS\_OUTPUT.PUT\_LINE('Employee ' || I.FIRST\_NAME || ' ' || I.LAST\_NAME || ' just got an increase of $' || V\_SALTOT);

-- ELSE DBMS\_OUTPUT.PUT\_LINE('no matching CASE found');

END IF;

END LOOP;

CASE

WHEN v\_counter > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Total # of employees who received salary increase is: ' || V\_COUNTER);

ELSE

DBMS\_OUTPUT.PUT\_LINE('No salary was modified in Department: ' || P\_DEPTID);

END CASE;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('This Department Id is invalid: ' || p\_deptID);

WHEN emptyEmp THEN

DBMS\_OUTPUT.PUT\_LINE('This Department is EMPTY: ' || p\_deptID);

END modify\_sal;

FUNCTION total\_cost(f\_studid NUMBER)

RETURN NUMBER IS

v\_total NUMBER;

v\_numenroll NUMBER;

v\_test NUMBER;

BEGIN

--To check if Student Exists

SELECT STUDENT\_ID INTO v\_test FROM STUDENT WHERE STUDENT\_ID = f\_studid;

--To check if student is enrolled at anything

SELECT COUNT(SECTION\_ID) INTO v\_numenroll FROM ENROLLMENT WHERE STUDENT\_ID = f\_studid;

IF(v\_numenroll = 0) THEN

RETURN 0;

END IF;

--Gets total cost for student

SELECT SUM(COST) INTO v\_total FROM COURSE WHERE COURSE\_NO IN(SELECT COURSE\_NO FROM SECTION WHERE SECTION\_ID IN (SELECT SECTION\_ID FROM ENROLLMENT WHERE STUDENT\_ID = f\_studid));

RETURN v\_total;

--Will trigger exception if Student id doesn't exist

EXCEPTION WHEN NO\_DATA\_FOUND THEN

RETURN -1;

END total\_cost;

FUNCTION total\_cost(f\_fname VARCHAR2,f\_lname VARCHAR2)

RETURN NUMBER IS

v\_total NUMBER;

v\_numenroll NUMBER;

v\_test NUMBER;

BEGIN

--To check if Student Exists

SELECT STUDENT\_ID INTO v\_test FROM STUDENT WHERE FIRST\_NAME = f\_fname AND LAST\_NAME = f\_lname;

--To check if student is enrolled at anything

SELECT COUNT(SECTION\_ID) INTO v\_numenroll FROM ENROLLMENT WHERE STUDENT\_ID = v\_test;

IF(v\_numenroll = 0) THEN

RETURN 0;

END IF;

--Gets total cost for student

SELECT SUM(COST) INTO v\_total FROM COURSE WHERE COURSE\_NO IN(SELECT COURSE\_NO FROM SECTION WHERE SECTION\_ID IN (SELECT SECTION\_ID FROM ENROLLMENT WHERE STUDENT\_ID = v\_test));

RETURN v\_total;

--Will trigger exception if Student id doesn't exist

EXCEPTION WHEN NO\_DATA\_FOUND THEN

RETURN -1;

END total\_cost;

FUNCTION total\_cost(f\_zip VARCHAR2)

RETURN NUMBER IS

CURSOR c\_students IS

select STUDENT\_ID from STUDENT

where ZIP = f\_zip;

v\_total NUMBER :=0;

v\_sum NUMBER := 0;

v\_numenroll NUMBER;

v\_test NUMBER;

BEGIN

--To check if ZIP Exists

SELECT COUNT(ZIP) INTO v\_test FROM STUDENT WHERE ZIP = f\_zip;

IF v\_test = 0 THEN

return -1;

END IF;

FOR v\_rec IN c\_students LOOP

--To check if student is enrolled at anything

SELECT COUNT(SECTION\_ID) INTO v\_numenroll FROM ENROLLMENT WHERE STUDENT\_ID = v\_rec.STUDENT\_ID;

IF(v\_numenroll = 0) THEN

v\_total := 0;

ELSE

--Gets total cost for student

SELECT SUM(COST) INTO v\_total

FROM COURSE

WHERE COURSE\_NO IN

(SELECT COURSE\_NO

FROM SECTION

WHERE SECTION\_ID IN

(SELECT SECTION\_ID FROM ENROLLMENT WHERE STUDENT\_ID = v\_rec.STUDENT\_ID));

END IF;

v\_sum := v\_sum + v\_total;

--Will trigger exception if Student id doesn't exist

END LOOP;

RETURN v\_sum;

END total\_cost;

END My\_pack ;

Q5)

CREATE OR REPLACE PROCEDURE mod\_grade(p\_course NUMBER, p\_grade NUMBER) IS

CURSOR c\_students IS

SELECT STUDENT\_ID

FROM ENROLLMENT

WHERE SECTION\_ID IN (SELECT SECTION\_ID FROM SECTION WHERE COURSE\_NO = p\_course);

v\_total NUMBER := 0;

v\_test NUMBER;

BEGIN

SELECT COURSE\_NO INTO v\_test FROM COURSE WHERE COURSE\_NO = p\_course;

IF(p\_grade <= 100 AND p\_grade >= 0) THEN

FOR v\_rec IN c\_students LOOP

UPDATE GRADE

SET NUMERIC\_GRADE = p\_grade

WHERE STUDENT\_ID = v\_rec.STUDENT\_ID;

v\_total := v\_total + 1;

END LOOP;

IF(v\_total != 0) THEN

DBMS\_OUTPUT.PUT\_LINE('Total # of grades changed to '|| p\_grade ||' for course number 130 is ' || v\_total);

ELSE

DBMS\_OUTPUT.PUT\_LINE('This Course has NOBODY enrolled so far: ' || p\_course);

END IF;

ELSE

DBMS\_OUTPUT.PUT\_LINE('This Grade invalid: '||p\_grade||' It must be between 0 and 100. Try again.');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('This Course Number is invalid: '|| p\_course);

END mod\_grade;

EXECUTE mod\_grade(130, 101)

ROLLBACK;

OUTPUT

1)

Execute My\_pack.modify\_sal(99);

ROLLBACK;

This Department Id is invalid: 99

Execute My\_pack.modify\_sal(190);

ROLLBACK;

This Department is EMPTY: 190

Execute My\_pack.modify\_sal(10);

ROLLBACK;

No salary was modified in Department: 10

Execute My\_pack.modify\_sal(110);

ROLLBACK;

Employee William Gietz just got an increase of $1850

Total # of employees who received salary increase is: 1

Execute My\_pack.modify\_sal(60);

ROLLBACK;

Employee David Austin just got an increase of $960

Employee Valli Pataballa just got an increase of $960

Employee Diana Lorentz just got an increase of $1560

Total # of employees who received salary increase is: 3

2)

B\_COSTTOT

----

1195

B\_COSTTOT

-

0

B\_COSTTOT

--

-1

3)

Execute My\_pack.modify\_sal(99);

ROLLBACK;

This Department Id is invalid: 99

Execute My\_pack.modify\_sal(190);

ROLLBACK;

This Department is EMPTY: 190

Execute My\_pack.modify\_sal(10);

ROLLBACK;

No salary was modified in Department: 10

Execute My\_pack.modify\_sal(110);

ROLLBACK;

Employee William Gietz just got an increase of $1850

Total # of employees who received salary increase is: 1

Execute My\_pack.modify\_sal(60);

ROLLBACK;

Employee David Austin just got an increase of $960

Employee Valli Pataballa just got an increase of $960

Employee Diana Lorentz just got an increase of $1560

Total # of employees who received salary increase is: 3

B\_COSTTOT

----

1195

B\_COSTTOT

-

0

B\_COSTTOT

--

-1

4)

B\_COSTTOT

----

1195

B\_COSTTOT

-

0

B\_COSTTOT

--

-1

B\_COSTTOT

----

1195

B\_COSTTOT

----

7070

B\_COSTTOT

--

-1

5)

Total # of grades changed to 75 for course number 130 is 8

This Course has NOBODY enrolled so far: 144

This Course Number is invalid: 99

This Grade invalid: 101 It must be between 0 and 100. Try again.