**CODING SNIPPET:**

-- 1. (a) create a student table with the following details (Assume your own datatypes for the data).

CREATE TABLE

STUDENT(STU\_ID INTEGER(6) NOT NULL PRIMARY KEY,

STU\_NAME VARCHAR(20) ,

SEX VARCHAR(10) ,

TOT\_MARKS INTEGER(3) );

INSERT INTO STUDENT VALUES(10,'Anu','Female',345),

(20,'Anbu','Male',347),

(30,'Malini','Female',456),

(40,'Sankar','Male',478),

(50,'Varshan','Male',324);

-- (a) Select the maximum marks of a student in the table

SELECT MAX(TOT\_MARKS) AS MAX\_MARK FROM STUDENT;

-- (b)Select the number of male and female students in the table.

SELECT COUNT(SEX) AS MALE\_COUNT FROM STUDENT WHERE SEX='Male';

SELECT COUNT(SEX) AS FEMALE\_COUNT FROM STUDENT WHERE SEX='Female';

-- (c) Select the average marks scored by male and female group of students in the table

SELECT AVG(TOT\_MARKS) AS AVG\_MARKS FROM STUDENT GROUP BY SEX;

-- (d) select all students who scored marks greater than the average mark of the students in the table

SELECT AVG(TOT\_MARKS) AS AVG\_MARKS FROM STUDENT;

--(e) Select the group of students who scored greater average marks than the average marks of the entire students in the table

SELECT STU\_NAME, AVG(TOT\_MARKS) FROM STUDENT WHERE TOT\_MARKS > (SELECT AVG(TOT\_MARKS) FROM STUDENT);

-- 2. (a) select all student details whose name value consists of exactly three characters.

SELECT \* FROM STUDENT WHERE STU\_NAME LIKE '\_\_\_';

-- (b) Display all students in ascending order of their sex and descending order of their marks.

SELECT \* FROM STUDENT ORDER BY SEX ASC,TOT\_MARKS DESC;

-- 3. Find the difference in marks between maximum and minimum in the class.

SELECT MAX(TOT\_MARKS) - MIN(TOT\_MARKS) DIFFERENCE FROM STUDENT;

-- 4. Create a view object for student table with name ‘mytab’ by assuming your own condition for the selection and demonstrate the role of ‘with check option’ for the created view.

CREATE VIEW MYTAB AS SELECT EMP\_ID,EMP\_NAME,DEPART\_ID,SALARY FROM EMPL WHERE SALARY >= 10000 WITH CHECK OPTION;

SELECT \* FROM MYTAB;

-- 5. Perform 3 insertions with student table and then set a savepoint s1. Again, perform any two deletion. Now check the execution effect of ROLLBACK to s1, commit followed by another Rollback.

START TRANSACTION;

INSERT INTO STUDENT VALUES(60,'Deepu','Female',345),

(70,'Keerthu','Female',378),

(80,'Mohan','Male',434);

SAVEPOINT S1;

DELETE FROM STUDENT WHERE STU\_ID IN (30,40);

ROLLBACK TO SAVEPOINT S1;

COMMIT;

ROLLBACK TO SAVEPOINT S1;

-- 6. Write a simple PL/SQL (Anonymous block) program to find the square and cube values for the given number;

DELIMITER //

CREATE PROCEDURE SAMPLE(IN X1 INT, IN X2 INT, OUT SQUARE INT, OUT CUBEE INT)

BEGIN

SET SQUARE= (X1\*X1);

SET CUBEE= (X2\*X2\*X2);

END//

CALL SAMPLE(2,4,@X,@Y);

SELECT @X;

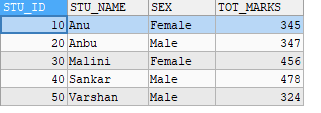
SELECT @Y;

**OUTPUT SCREENSHOTS:**

**1.(A)**

****

**(a)**

****

**(B)**

****

**(C)**

****

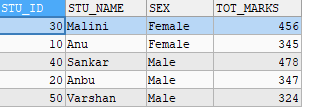
**(D)**

****

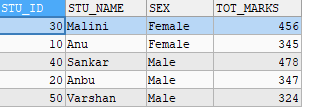
**2.(a)**

****

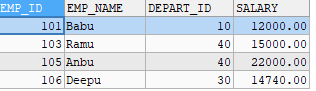
**(B)**

****

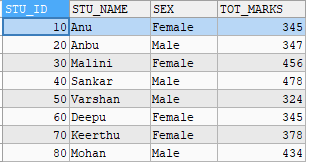
**3.**

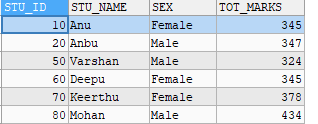
****

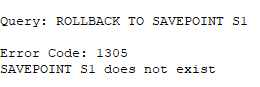
**4.**

****

**5.**

****

****

****

**6.**

** **

CREATE OR REPLACE PROCEDURE deleteprocedure AS BEGIN delete from depart where dept\_id=30; dbms\_output.put\_line('deleted successfully!'); END;