Oracle Assignment

**Section 1: Creating Tables**

1. CREATE TABLE department (department\_id NUMBER,

department\_name VARCHAR(30),

department\_block\_number NUMBER,

CONSTRAINT department\_pk PRIMARY KEY(department\_id));

2. CREATE TABLE student (student\_id NUMBER,

student\_name VARCHAR(30),

address VARCHAR(40),

city VARCHAR(30),

department\_id NUMBER,

CONSTRAINT student\_pk PRIMARY KEY (student\_id),

CONSTRAINT department\_id\_fk FOREIGN KEY (department\_id)

REFERENCES department(department\_id));

3. CREATE TABLE staff (staff\_id NUMBER,

staff\_name VARCHAR(30),

department\_id number,

CONSTRAINT staff\_pk PRIMARY KEY (staff\_id),

CONSTRAINT staff\_department\_id\_fk FOREIGN KEY (department\_id)

REFERENCES department(department\_id));

4. CREATE TABLE subject (subject\_id NUMBER,

subject\_name VARCHAR(30),

subject\_code varchar(10),

staff\_id NUMBER,

CONSTRAINT subject\_pk PRIMARY KEY (subject\_id),

CONSTRAINT subject\_staff\_id\_fk FOREIGN KEY (staff\_id)

REFERENCES staff(staff\_id));

5. CREATE TABLE mark (

value\_mark NUMBER,

subject\_id NUMBER,

student\_id NUMBER,

CONSTRAINT mark\_pk PRIMARY KEY (subject\_id, student\_id),

CONSTRAINT mark\_subject\_id\_fk FOREIGN KEY (subject\_id)

REFERENCES subject(subject\_id) ON DELETE CASCADE,

CONSTRAINT mark\_student\_id\_fk FOREIGN KEY (student\_id)

REFERENCES student(student\_id) ON DELETE CASCADE

);

6. ALTER TABLE staff MODIFY staff\_name NOT NULL;

7. ALTER TABLE student ADD emailid VARCHAR(20);

8. ALTER TABLE student MODIFY emailid VARCHAR(50);

9. ALTER TABLE student DROP COLUMN emailid;

**Section 3: Updating Records**

12. UPDATE subject

SET subject\_name = 'Computer Science', subject\_code = 1919

WHERE subject\_name = 'Sales';

**Section 4: Deleting Records**

13. DELETE FROM subject WHERE subject\_name = 'Accounting';

**Section 5: Basic Selection of Records**

14. SELECT subject\_name

FROM subject

ORDER BY subject\_name ASC;

15. SELECT department\_name

FROM department

WHERE department\_block\_number BETWEEN 3 AND 10;

16. SELECT student\_name FROM student ORDER BY student\_name ASC;

**Section 6: Selecting Single Rows**

17. SELECT student\_name

FROM student

WHERE city IN(‘Chicago’, ‘Taylor’, ‘San Jose’)

ORDER BY student\_id ASC;

18 SELECT address AS Address\_Student, CITY

FROM student;

19. SELECT student\_name

FROM student

WHERE LEN(student\_name) <= 6 ;

**Section 7: Selecting Groups**

20. SELECT department\_block\_number, COUNT(department\_name)

FROM department

GROUP BY department\_block\_number

ORDER BY department\_block\_number ASC;

21. SELECT COUNT(\*) AS stud\_count

FROM student;

**Section 8: SQL Joins**

22. SELECT department.department\_name, COUNT(student.student\_id) AS student\_count

FROM department INNER JOIN student

ON department.department\_id = student.department\_id

GROUP BY department.department\_name

ORDER BY department.department\_name ASC;

23. SELECT student.student\_name, subject.subject\_name

FROM student

CROSS JOIN subject

WHERE subject.subject\_code > 1600;

24. SELECT student\_name, subject\_name

FROM student

INNER JOIN mark ON student.student\_id = mark.student\_id

INNER JOIN subject ON subject.subject\_id = mark.subject\_id

WHERE mark.value\_mark < 3;

**Section 9: Selecting Sub-Queries**

25. SELECT department\_block\_number

FROM (SELECT department\_block\_number,

COUNT(department\_block\_number),

RANK() OVER(ORDER BY COUNT(department\_block\_number) DESC) rnk

FROM department

GROUP BY department\_block\_number)

WHERE rnk= 1;

26. SELECT staff\_name

FROM staff

WHERE staff.staff\_id NOT IN (SELECT subject.staff\_id

FROM subject)

ORDER BY staff\_name ASC;

**Section 10: Functions**

27. CREATE OR REPLACE FUNCTION find\_dept\_name (p\_department\_id NUMBER)

RETURN VARCHAR

AS

l\_department\_name department.department\_name%type;

BEGIN

SELECT department\_name INTO l\_department\_name

FROM department

WHERE department\_id = p\_department\_id;

RETURN l\_department\_name;

END find\_dept\_name;

/

28. CREATE OR REPLACE FUNCTION find\_dept\_block (p\_department\_id NUMBER)

RETURN NUMBER

AS

l\_department\_block department.department\_block\_number%TYPE;

BEGIN

SELECT department\_block\_number INTO l\_department\_block

FROM department

WHERE department\_id = p\_department\_id;

RETURN l\_department\_block;

END find\_dept\_block;

/

29. CREATE OR REPLACE FUNCTION find\_staff\_name(p\_staff\_id NUMBER)

RETURN VARCHAR

AS

l\_staff\_name staff.staff\_name%TYPE;

BEGIN

SELECT STAFF\_NAME INTO l\_staff\_name

FROM STAFF

WHERE STAFF\_ID = p\_staff\_id;

RETURN l\_staff\_name;

END find\_staff\_name;

/

**Section 11: Triggers**

30. CREATE OR REPLACE TRIGGER trigger\_department\_af\_update

AFTER UPDATE

ON department

BEGIN

IF UPDATING THEN

DBMS\_OUTPUT.PUT\_LINE('DEPARTMENTS table has been updated');

END IF;

END trigger\_department\_af\_update;

31. CREATE OR REPLACE TRIGGER trigger\_department\_bf\_delete

BEFORE DELETE

ON department

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE('A row has been deleted from DEPARTMENT');

END trigger\_department\_bf\_delete;

**Section 12: Views and Index**

32. CREATE INDEX stud\_name\_idx ON student (student\_name);

33. CREATE OR REPLACE VIEW staff\_name\_view AS

SELECT staff\_name

FROM staff;

**Section 13: Cursors**

36. DECLARE

CURSOR c1 is

SELECT \* FROM student;

l\_cur c1%rowtype;

BEGIN

OPEN c1;

LOOP

FETCH c1 INTO l\_cur;

EXIT WHEN c1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(l\_cur.student\_id || ' ' || l\_cur.student\_name || ' ' || l\_cur.address || ' ' || l\_cur.city || ' ' || l\_cur.department\_id);

END LOOP;

CLOSE C1;

END;

**Section 14: Packages and Procedures**

37. CREATE OR REPLACE PACKAGE College

AS

PROCEDURE select\_departments;

FUNCTION select\_student(p\_stud\_id NUMBER)

RETURN VARCHAR;

END College;

38. CREATE OR REPLACE PACKAGE BODY College

AS

PROCEDURE select\_departments

AS

l\_curs sys\_refcursor;

r1 department%rowtype;

BEGIN

OPEN l\_curs

FOR SELECT \* FROM department;

LOOP

FETCH l\_curs INTO r1;

EXIT WHEN l\_curs%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(r1.department\_id || ' ' || r1.department\_name || ' ' || r1.department\_block\_number);

END LOOP;

END select\_departments;

FUNCTION select\_student(l\_num NUMBER)

RETURN VARCHAR

AS

l\_student\_name student.student\_name%TYPE;

BEGIN

SELECT student\_name INTO l\_student\_name

FROM student

WHERE student\_id = l\_num;

RETURN l\_student\_name;

END select\_student;

END College;