Department of Computer Science and Engineering
Islamic University of Technology (IUT)
A subsidiary organ of OIC

Lab Report 2
CSE 4508 :RDBMS

Name:A N M Zahid Hossain Milkan
Student ID:200041202
Section:2B
Semester:Winter (5th)
Academic Year:2022-23

Date of Submission:21.08.2023

Task 1:Creating Tablespace,one user and 4 objects (tables) and assigning data of one table to another tablespace :

SQL:

```
--creating table space

CREATE TABLESPACE myspace DATAFILE 'myspace.dbf' SIZE
100M;
CREATE TABLESPACE myspace2 DATAFILE 'myspace20.dbf' SIZE
100M;



SELECT name FROM v$pdbs;

NAME
-----------------------------------------------------------
----------------------
PDB$SEED
XEPDB1
-- Switch to the desired PDB
ALTER SESSION SET CONTAINER = XEPDB1;

--creating a user
CREATE USER lab2 IDENTIFIED BY anm123 DEFAULT TABLESPACE
myspace;

--Assign the tablespace to the user
```

```sql
ALTER USER lab2 QUOTA UNLIMITED ON myspace;

--creating 4 tables/objects

CREATE TABLE lab2.T1 (id INT, name VARCHAR2(10))
TABLESPACE myspace;
CREATE TABLE lab2.T2 (id INT, name VARCHAR2(10))
TABLESPACE myspace;
CREATE TABLE lab2.T3 (id INT, name VARCHAR2(10))
TABLESPACE myspace;
CREATE TABLE lab2.T4 (id INT, name VARCHAR2(10))
TABLESPACE myspace;

-- Inserting data into the T4 table

INSERT INTO lab2.T4 (id, name) VALUES (202, 'Zahid');

-- Moving the T4 table to the myspace2 tablespace

ALTER TABLE lab2.T4 MOVE TABLESPACE myspace2;
```

Result :

```
SQL> CREATE USER lab2 IDENTIFIED BY anm123 DEFAULT TABLESPACE myspace;

User created.

SQL> ALTER USER lab2 QUOTA UNLIMITED ON myspace;

User altered.

SQL> ALTER USER lab2 QUOTA UNLIMITED ON myspace2;

User altered.

SQL> CREATE TABLE lab2.T1 (id INT, name VARCHAR2(10)) TABLESPACE myspace;

Table created.

SQL> CREATE TABLE lab2.T2 (id INT, name VARCHAR2(10)) TABLESPACE myspace;

Table created.

SQL> CREATE TABLE lab2.T3 (id INT, name VARCHAR2(10)) TABLESPACE myspace;

Table created.

SQL> CREATE TABLE lab2.T4 (id INT, name VARCHAR2(10)) TABLESPACE myspace;

Table created.

SQL> INSERT INTO lab2.T4 (id, name) VALUES (202, 'Zahid');

1 row created.

SQL> LTER TABLE lab2.T4 MOVE TABLESPACE myspace2;
SP2-0734: unknown command beginning "LTER TABLE..." - rest of line ignored.
SQL> ALTER TABLE lab2.T4 MOVE TABLESPACE myspace2;

Table altered.
```

Task 2:Subquery and Inline-view example with related tables:

SQL:

```
CREATE TABLE students (
    student_id NUMBER PRIMARY KEY,
    student_name VARCHAR2(100)
);

CREATE TABLE grades (
    grade_id NUMBER PRIMARY KEY,
    student_id NUMBER,
    subject VARCHAR2(50),
    grade NUMBER
);

-- Insert sample data
INSERT INTO students VALUES (1, 'Alice');
INSERT INTO students VALUES (2, 'Bob');
INSERT INTO students VALUES (3, 'Charlie');

INSERT INTO grades VALUES (101, 1, 'Math', 85);
INSERT INTO grades VALUES (102, 1, 'Science', 92);
INSERT INTO grades VALUES (103, 2, 'Math', 78);
INSERT INTO grades VALUES (104, 2, 'Science', 89);
INSERT INTO grades VALUES (105, 3, 'Math', 94);
INSERT INTO grades VALUES (106, 3, 'Science', 88);


--nested subquery
SELECT student_id, student_name
```

```sql
FROM students
WHERE student_id IN (
    SELECT student_id
    FROM grades
    GROUP BY student_id
    HAVING AVG(grade) > 85
);


--inline view
SELECT s.student_name, max_grade.highest_grade
FROM students s
JOIN (
    SELECT student_id, MAX(grade) AS highest_grade
    FROM grades
    GROUP BY student_id
) max_grade ON s.student_id = max_grade.student_id;
```

Results:

```
SQL> set pagesize 100 linesize 400;
SQL> SELECT student_id, student_name
  2  FROM students
  3  WHERE student_id IN (
  4      SELECT student_id
  5      FROM grades
  6      GROUP BY student_id
  7      HAVING AVG(grade) > 85
  8  );

STUDENT_ID STUDENT_NAME
---------- --------------------------------------------------------------------------------------------------------------------------------
         1 Alice
         3 Charlie

SQL> SELECT s.student_name, max_grade.highest_grade
  2  FROM students s
  3  JOIN (
  4      SELECT student_id, MAX(grade) AS highest_grade
  5      FROM grades
  6      GROUP BY student_id
  7  ) max_grade ON s.student_id = max_grade.student_id;

STUDENT_NAME                                                                                                                     HIGHEST_GRADE
-------------------------------------------------------------------------------------------------------------------------------- -------------
Alice                                                                                                                                      92
Bob                                                                                                                                        89
Charlie                                                                                                                                    94

SQL>
```

Task 3:Showing differences between Left(Outer), Right(Inner) and Natural Join with suitable examples:

```sql
-- Create Department table
CREATE TABLE department (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(50)
);

-- Insert data into Departments table
INSERT INTO department VALUES (1, 'CSE');
INSERT INTO department VALUES (2, 'EEE');
INSERT INTO department VALUES (3, 'MPE');
INSERT INTO department VALUES (4, 'BTM');

-- Create Student table
CREATE TABLE Student (
```

```
    student_id INT PRIMARY KEY,
    dept_id INT,
     batch INT,
    FOREIGN KEY (dept_id) REFERENCES department(dept_id)
);

-- Insert data into Student table
INSERT INTO Student VALUES (101, 1,20 );
INSERT INTO Student VALUES (102, 1,21 );
INSERT INTO Student VALUES (103, 2, 20);
INSERT INTO Student VALUES (104, 3,22 );


--LEFT OUTER JOIN
SELECT department.dept_name, Student.batch
FROM department
LEFT JOIN Student ON department.dept_id =
Student.dept_id;


--RIGHT INNER JOIN
SELECT department.dept_name, Student.batch
FROM department
RIGHT JOIN Student ON department.dept_id =
Student.dept_id;
--NATURAL JOIN
SELECT Student.student_id,department.dept_name,
Student.batch
FROM department
NATURAL JOIN Student;
```

Result :

```
SQL> SELECT department.dept_name, Student.batch
  2  FROM department
  3  LEFT JOIN Student ON department.dept_id = Student.dept_id;

DEPT_NAME                                                 BATCH
-------------------------------------------------- ----------
CSE                                                       20
CSE                                                       21
EEE                                                       20
MPE                                                       22
BTM

SQL> SELECT department.dept_name, Student.batch
  2  FROM department
  3  RIGHT JOIN Student ON department.dept_id = Student.dept_id;

DEPT_NAME                                                 BATCH
-------------------------------------------------- ----------
CSE                                                       20
CSE                                                       21
EEE                                                       20
MPE                                                       22

SQL> SELECT Student.student_id,department.dept_name, Student.batch
  2  FROM department
  3  NATURAL JOIN Student;

STUDENT_ID DEPT_NAME                                      BATCH
---------- -------------------------------------------- ----------
       101 CSE                                              20
       102 CSE                                              21
       103 EEE                                              20
       104 MPE                                              22

SQL>
```