ISLAMIC UNIVERSITY OF TECHNOLOGY

Organization of Islamic Cooperation

Board Bazar, Gazipur

Lab 03

CSE 4508

Md. Mohsinul Kabir

# Answer 1

/\* create a new tablespace \*/  
CREATE TABLESPACE myspace DATAFILE 'myspace\_data.dbf' SIZE 1M;  
  
/\* create a new user \*/  
CREATE USER u1 IDENTIFIED BY 1234;  
  
/\* assign the tablespace to the user \*/  
ALTER USER u1 QUOTA UNLIMITED ON myspace;  
  
/\* creating 4 tables \*/  
CREATE TABLE t1 (  
 id NUMBER(9, 0),  
 name VARCHAR2(20),  
 cgpa NUMBER(3, 2)  
) TABLESPACE myspace;  
  
CREATE TABLE t2 (  
 id NUMBER(9, 0)  
) TABLESPACE myspace;  
  
CREATE TABLE t3 (  
 id NUMBER(9, 0)  
) TABLESPACE myspace;  
  
CREATE TABLE t4 (  
 id NUMBER(9, 0)  
) TABLESPACE myspace;  
  
/\* proves that created tables exist in created tablespace \*/  
SELECT table\_name, tablespace\_name FROM user\_tables WHERE table\_name = 'T1';  
  
/\* create another tablespace \*/  
CREATE TABLESPACE myspace2 DATAFILE 'myspace2\_data.dbf' SIZE 1M;  
  
/\* moving table to new tablespace \*/  
ALTER TABLE t4 MOVE TABLESPACE myspace2;  
  
/\* proves that table has been moved to new tablespace \*/  
SELECT table\_name, tablespace\_name FROM user\_tables WHERE table\_name = 'T4';

SQL

# Answer 2

INSERT INTO t1 VALUES (1, 'John', 3.75);  
INSERT INTO t1 VALUES (2, 'James', 3.50);  
INSERT INTO t1 VALUES (3, 'Olive', 3.25);  
  
/\* nested subquery \*/  
SELECT \* FROM t1 WHERE cgpa > (SELECT AVG(cgpa) FROM t1);  
  
/\* inline view; not the best example - redundant \*/  
SELECT AVG(cgpa) FROM (SELECT \* FROM t1 WHERE name LIKE 'J%');

SQL

# Answer 3

CREATE TABLE customers (  
 customer\_id INT,  
 last\_name VARCHAR2(20),  
 first\_name VARCHAR2(20)  
);  
  
INSERT INTO customers VALUES (4000, 'Jackson', 'Joe');  
INSERT INTO customers VALUES (5000, 'Smith', 'Jane');  
INSERT INTO customers VALUES (6000, 'Ferguson', 'Samantha');  
INSERT INTO customers VALUES (7000, 'Reynolds', 'Allen');  
INSERT INTO customers VALUES (8000, 'Anderson', 'Paige');  
INSERT INTO customers VALUES (9000, 'Johnson', 'Derek');  
  
CREATE TABLE orders (  
 order\_id INT,  
 customer\_id INT,  
 order\_date DATE  
);  
  
INSERT INTO orders VALUES (1, 7000, TO\_DATE(‘2016/04/18’, ‘YYYY/MM/DD’));  
INSERT INTO orders VALUES (2, 5000, TO\_DATE(‘2016/04/18’, ‘YYYY/MM/DD’));  
INSERT INTO orders VALUES (3, 8000, TO\_DATE(‘2016/04/19’, ‘YYYY/MM/DD’));  
INSERT INTO orders VALUES (4, 4000, TO\_DATE(‘2016/04/20’, ‘YYYY/MM/DD’));  
INSERT INTO orders VALUES (5, NULL, TO\_DATE(‘2016/05/01’, ‘YYYY/MM/DD’));

SQL

## Inner Join

SELECT order\_id, order\_date, orders.customer\_id, first\_name, last\_name

FROM orders

INNER JOIN customers ON orders.customer\_id = customers.customer\_id;

SQL

### Output

ORDER\_ID ORDER\_DAT CUSTOMER\_ID FIRST\_NAME LAST\_NAME  
---------- --------- ----------- -------------------- --------------------  
 4 20-APR-16 4000 Joe Jackson  
 2 18-APR-16 5000 Jane Smith  
 1 18-APR-16 7000 Allen Reynolds  
 3 19-APR-16 8000 Paige Anderson

TEXT

## Outer Join

SELECT order\_id, order\_date, orders.customer\_id, first\_name, last\_name

FROM orders FULL JOIN customers ON orders.customer\_id = customers.customer\_id;

SQL

### Output

ORDER\_ID ORDER\_DAT CUSTOMER\_ID FIRST\_NAME LAST\_NAME  
---------- --------- ----------- -------------------- --------------------  
 4 20-APR-16 4000 Joe Jackson  
 2 18-APR-16 5000 Jane Smith  
 Samantha Ferguson  
 1 18-APR-16 7000 Allen Reynolds  
 3 19-APR-16 8000 Paige Anderson  
 Derek Johnson  
 5 01-MAY-16

TEXT

## Left Join

SELECT order\_id, order\_date, orders.customer\_id, first\_name, last\_name

FROM orders LEFT JOIN customers ON orders.customer\_id = customers.customer\_id;

SQL

### Output

ORDER\_ID ORDER\_DAT CUSTOMER\_ID FIRST\_NAME LAST\_NAME  
---------- --------- ----------- -------------------- --------------------  
 4 20-APR-16 4000 Joe Jackson  
 2 18-APR-16 5000 Jane Smith  
 1 18-APR-16 7000 Allen Reynolds  
 3 19-APR-16 8000 Paige Anderson  
 5 01-MAY-16

TEXT

## Right Join

SELECT order\_id, order\_date, orders.customer\_id, first\_name, last\_name

FROM orders

RIGHT JOIN customers ON orders.customer\_id = customers.customer\_id;

SQL

### Output

ORDER\_ID ORDER\_DAT CUSTOMER\_ID FIRST\_NAME LAST\_NAME  
---------- --------- ----------- -------------------- --------------------  
 1 18-APR-16 7000 Allen Reynolds  
 2 18-APR-16 5000 Jane Smith  
 3 19-APR-16 8000 Paige Anderson  
 4 20-APR-16 4000 Joe Jackson  
 Derek Johnson  
 Samantha Ferguson

TEXT

## Left and Right Outer Joins

In a Left Outer Join, every record from the left-hand side table is shown. If any record does not have a value for one of the attributes, a blank space is shown in the place of the value. For the right-hand side table, only the records that meet the matching criteria (orders.customer\_id = customers.customer\_id in the above examples) are shown.

In a Right Outer Join, the opposite happens. Every record from the right-hand side table is shown. If any record does not have a value for one of the attributes, a blank space is shown in the place of the value. For the left-hand side table, only the records that meet the matching criteria are shown.

## Inner Joins and Natural Joins

A Natural Join joins records that have the same value for every attribute that has the same name and datatype in both tables.

For example, the following query gives us the same output that we got using the Inner Join example above.

SELECT order\_id, order\_date, customer\_id, first\_name, last\_name

FROM orders NATURAL JOIN customers;

SQL

Output

ORDER\_ID ORDER\_DAT CUSTOMER\_ID FIRST\_NAME LAST\_NAME  
---------- --------- ----------- -------------------- --------------------  
 4 20-APR-16 4000 Joe Jackson  
 2 18-APR-16 5000 Jane Smith  
 1 18-APR-16 7000 Allen Reynolds  
 3 19-APR-16 8000 Paige Anderson

TEXT

An Inner Join on the other hand, joins records that have the same value for the specified matching criteria. The example we saw earlier is the only sensible join to perform in this case, and it just so happened to give us the same results as a natural join. However, we could actually perform the join operation in less sensible ways as well. All the Inner Join cares about is that the values that it is being told to check match.

Say we add a new customer to the customers table.

INSERT INTO customers VALUES (1, 'John', 'Smith');

SQL

We can now perform an Inner Join like this:

SELECT order\_id, order\_date, orders.customer\_id, first\_name, last\_name

FROM orders INNER JOIN customers ON orders.order\_id = customers.customer\_id;

SQL

Output

ORDER\_ID ORDER\_DAT CUSTOMER\_ID FIRST\_NAME LAST\_NAME  
---------- --------- ----------- -------------------- --------------------  
 1 18-APR-16 7000 Smith John

TEXT

Of course, no sensible person would try to match order IDs with customer IDs, but the point is that Inner Joins allow us to do this whereas Natural Joins do not.