

BITS Pilani (K.K. Birla Goa Campus)
Database Systems – Evaluation Lab-2

Instructions:

Execute queries for the following. Copy the queries and the results to a text file. Please make sure to number the question correctly.

Cartesian Product : BASIC-

The basic syntax for the Cartesian product is :

```
SELECT table1.column1, table2.column2...  
FROM table1, table2 [, table3 ]
```

Consider the following two tables, (a) CUSTOMERS table is as follows:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

(b) Another table is ORDERS as follows:

OID	DATE	CUSTOMER_ID	AMOUNT
102	2009-10-08 00:00:00	3	3000
100	2009-10-08 00:00:00	3	1500
101	2009-11-20 00:00:00	2	1560
103	2008-05-20 00:00:00	4	2060

The Cartesian product of the two tables is :

```
SQL> SELECT ID, NAME, AMOUNT, DATE  
FROM CUSTOMERS, ORDERS;
```

This would produce the following result:

ID	NAME	AMOUNT	DATE
1	Ramesh	3000	2009-10-08 00:00:00
1	Ramesh	1500	2009-10-08 00:00:00
1	Ramesh	1560	2009-11-20 00:00:00
1	Ramesh	2060	2008-05-20 00:00:00
2	Khilan	3000	2009-10-08 00:00:00
2	Khilan	1500	2009-10-08 00:00:00
2	Khilan	1560	2009-11-20 00:00:00
2	Khilan	2060	2008-05-20 00:00:00
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
3	kaushik	1560	2009-11-20 00:00:00
3	kaushik	2060	2008-05-20 00:00:00
4	Chaitali	3000	2009-10-08 00:00:00
4	Chaitali	1500	2009-10-08 00:00:00
4	Chaitali	1560	2009-11-20 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00
5	Hardik	3000	2009-10-08 00:00:00
5	Hardik	1500	2009-10-08 00:00:00
5	Hardik	1560	2009-11-20 00:00:00
5	Hardik	2060	2008-05-20 00:00:00
6	Komal	3000	2009-10-08 00:00:00
6	Komal	1500	2009-10-08 00:00:00
6	Komal	1560	2009-11-20 00:00:00
6	Komal	2060	2008-05-20 00:00:00
7	Muffy	3000	2009-10-08 00:00:00
7	Muffy	1500	2009-10-08 00:00:00
7	Muffy	1560	2009-11-20 00:00:00
7	Muffy	2060	2008-05-20 00:00:00

This gives $7 \times 4 = 28$ results. Suppose you would like to find the names of the customers who have made certain orders the query would look like:

```
SELECT ID, NAME, AMOUNT, DATE FROM CUSTOMERS, ORDERS WHERE  
CUSTOMERS.ID = ORDERS.CUSTOMER_ID
```

Sub-query : BASIC

SELECT within SELECT command (also called subquery: query inside query)

Subqueries are legal in a SELECT statement's FROM clause. The actual syntax is:

```
SELECT ... FROM (subquery) [AS] name ...
```

For the sake of illustration, assume that you have this table:

```
CREATE TABLE t1 (s1 INT, s2 CHAR(5), s3 FLOAT);
```

Here is how to use a subquery in the FROM clause, using the example table:

```
INSERT INTO t1 VALUES (1,'1',1.0);
```

```
INSERT INTO t1 VALUES (2,'2',2.0);
```

```
SELECT sb1,sb2,sb3  
FROM (SELECT s1 AS sb1, s2 AS sb2, s3*2 AS sb3 FROM t1) AS sb  
WHERE sb1 > 1;
```

```
+-----+-----+-----+  
| sb1 | sb2 | sb3 |  
+-----+-----+-----+  
|  2 | 2  |  4 |  
+-----+-----+-----+
```

Here is another example: Suppose that you want to know the average of a set of sums for a grouped table.

This does not work:

```
SELECT AVG(SUM(column1)) FROM t1 GROUP BY column1;
```

However, this query provides the desired information:

```
SELECT AVG(sum_column1)  
FROM (SELECT SUM(column1) AS sum_column1  
      FROM t1 GROUP BY column1) AS t1;
```

Notice that the column name used within the subquery (sum_column1) is recognized in the outer query.

Questions (45 marks)

Q1) Find the average of all the amounts of house loans.
e.g-hou1,hou2..

(3marks)

Q2) Display the account numbers of all even zone branch accounts with balance greater than 5000.
e.g-zone2,zone4..

(3 marks)

Q3) Display the 5 lexicographically largest city names from the BRANCH table. (3 marks)

Q4) Find male customers who are not depositors.

(3 marks)

Q5) Find the names of cities in alphabetical order where more than 1 loan has been issued.

(3 marks)

Q6) Find the name and amount of the customer(s) who have made maximum payment of their loans.

(6 marks)

Q7) Display the names and account numbers of all the female customers who have taken exactly three loans.

(6 marks)

Q8) Display the name and balance of all the customers who have accessed their accounts in the month of May and who also have taken more than one loan.

(9 marks)

Q9) For each payment date, give amount and cumulative amount made till that date. (9 marks)

The output should look like this :

```
+-----+-----+-----+
| P_NO | L_NO | DATE    | AMOUNT |
+-----+-----+-----+
| x    | x1   | 2011-01-09 | 100    |
| y    | y1   | 2000-01-09 | 90     |
| z    | z1   | 2011-01-01 | 3000   |
+-----+-----+-----+
| DATE    | AMOUNT | CUM_AMOUNT |
+-----+-----+-----+
| 2000-01-09 | 90    | 90        |
| 2011-01-01 | 100   | 190       |
| 2011-01-09 | 3000  | 3190      |
+-----+-----+-----+
```