Instructor: Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cgs2545/spr2012

Department of Electrical Engineering and Computer Science
Computer Science Division
University of Central Florida
SQL In Class Exercises

- Use the following database scheme for problems 1-9 in this exercise.

```
<table>
<thead>
<tr>
<th>Suppliers</th>
<th>Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>snum</td>
<td>pnum</td>
</tr>
<tr>
<td>sname</td>
<td>pname</td>
</tr>
<tr>
<td>status</td>
<td>color</td>
</tr>
<tr>
<td>city</td>
<td>weight</td>
</tr>
<tr>
<td></td>
<td>weight</td>
</tr>
<tr>
<td></td>
<td>city</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shipments</th>
<th>Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>snum</td>
<td>inum</td>
</tr>
<tr>
<td>pnum</td>
<td>jnum</td>
</tr>
<tr>
<td>inum</td>
<td>quantity</td>
</tr>
<tr>
<td></td>
<td>numworkers</td>
</tr>
<tr>
<td></td>
<td>city</td>
</tr>
</tbody>
</table>
```

- Develop SQL expressions for each of the following queries:
SQL In Class Exercises

- The schema version of the database for problems 1-9.
1. List only the names of those suppliers who ship a part that weighs more than 200.

Solutions

SELECT sname
FROM suppliers NATURAL JOIN shipments CROSS JOIN parts
WHERE weight > 200 AND shipments.pnum = parts.pnum;

- or -

SELECT sname
FROM suppliers
WHERE snum IN (SELECT snum
FROM shipments
WHERE pnum IN (SELECT pnum
FROM parts
WHERE weight > 200) ) );

Note that a second natural join won’t work here since the join would also occur on the city attribute, which would be a more restrictive query.
2. List the names of those cities in which both a supplier and a job are located.

**Solutions**

```sql
SELECT supplier.city
FROM suppliers NATURAL JOIN jobs;
```

- or -

```sql
SELECT supplier.city
FROM suppliers
WHERE city IN (SELECT city
               FROM jobs);
```
3. List the names of those jobs that receive a shipment from supplier number S1.

Solutions

```
SELECT jname
FROM jobs
WHERE jnum IN (SELECT jnum
               FROM shipments
               WHERE snum = "S1");

- or -

SELECT jname
FROM jobs NATURAL JOIN shipments
WHERE snum = "S1";
```
4. List the names of those parts that are not shipped to any job.

Solutions

SELECT pname
FROM parts
WHERE pnum NOT IN (SELECT pnum
    FROM shipments);

- or -

SELECT pname
FROM parts
WHERE NOT EXISTS (SELECT *
    FROM shipments
    WHERE shipments.pnum = parts.pnum);
5. List the names of those suppliers who ship part number P2 to any job.

Solutions

```
SELECT sname
FROM suppliers
WHERE snum IN (SELECT snum
    FROM shipments
    WHERE pnum = "P2");
```

- or -

```
SELECT sname
FROM suppliers NATURAL JOIN shipments
WHERE pnum = "P2";
```
6. List the names of those suppliers who do not ship part number P2 to any job.

Solutions

SELECT sname
FROM suppliers
WHERE snum NOT IN (SELECT snum
                      FROM shipments
                      WHERE pnum = "P2");

- or -

SELECT sname
FROM suppliers
WHERE NOT EXISTS (SELECT *
                  FROM shipments
                  WHERE shipments.snum = suppliers.snum AND shipments.pnum = "P2");

Note that neither of the following are correct!

SELECT sname
FROM suppliers
WHERE snum = (SELECT snum
              FROM shipments
              WHERE pnum ≠ "P2");

- or -

SELECT sname
FROM suppliers
WHERE snum IN (SELECT snum
                FROM shipments
                WHERE snum ≠ "P2");
7. List the names of those suppliers who ship part at least one red part to any job.

Solutions

SELECT sname
FROM suppliers
WHERE snum IN (SELECT snum
    FROM shipments
    WHERE pnum IN (SELECT pnum
        FROM parts
        WHERE color = "red" ));

- or -

SELECT sname
FROM suppliers NATURAL JOIN shipments
WHERE pnum IN (SELECT pnum
    FROM parts
    WHERE color = "red" );
8. List the part number for every part that is shipped by more than one supplier.

Solution

```
SELECT pnum
FROM shipments
GROUP BY pnum
HAVING COUNT(snum) > 1;
```

*WHERE clause restricts by rows*
*HAVING clause restricts by groups*
9. List the names of those suppliers who ship every part.

Solutions

SELECT sname
FROM suppliers
WHERE NOT EXISTS (SELECT *
    FROM parts
    WHERE NOT EXISTS (SELECT *
        FROM shipments
        WHERE shipments.snum = suppliers.snum
            AND shipments.pnum = parts.pnum ) );

- or -

SELECT sname
FROM suppliers
WHERE (SELECT COUNT (shipments.pnum)
    FROM shipments
    WHERE shipments.snum = suppliers.snum
        AND shipments.pnum = parts.pnum ) =
    (SELECT COUNT (parts.pnum)
        FROM parts);
SQL In Class Exercises

• Use the following database scheme for problems 10- in this exercise.

• Develop SQL expressions for each of the following queries:
SQL In Class Exercises

- The schema version of the database.

```
customer
  | cust_id | name | address | . . . |
```

```
order
  | order_id | date | cust_id |
```

```
order_line
  | order_id | product_id | quantity |
```

```
product
  | product_id | description | finish | price |
```
10. List the date of every order placed by customer 5.

Solutions

SELECT date
FROM order
WHERE cust_id = 5;

- or -

SELECT DISTINCT date
FROM order
WHERE cust_id = 5;
11. List all the cities from which a customer placed an order on March 29th.

Solutions

SELECT DISTINCT city
FROM customer NATURAL JOIN order
WHERE date = "March 29";

- or -

SELECT DISTINCT city
FROM customer
WHERE cust_id IN (SELECT cust_id
FROM order
WHERE date = "March 29");
12. List the dates for every order placed that included part number 6.

Solutions

SELECT DISTINCT date
FROM order NATURAL JOIN order_line
WHERE product_id = 6;

- or -

SELECT DISTINCT date
FROM order
WHERE order_id IN (SELECT order_id
FROM order_line
WHERE product_id = 6);
13. List the names of those customers who have not placed any orders.

Solution

SELECT name
FROM customer
WHERE cust_id NOT IN (SELECT cust_id
FROM order);
14. List the names of those customers who have never ordered part number 6.

Solution

SELECT DISTINCT name
FROM customer
WHERE cust_id NOT IN (SELECT cust_id
    FROM order
    WHERE order_id IN (SELECT order_id
        FROM order_line
        WHERE product_id = 6)
    );
15. List the names of those customers who have ordered both part number 5 and part number 6.

Solution

```sql
SELECT DISTINCT name
FROM customer
WHERE (cust_id IN (SELECT cust_id
    FROM order
    WHERE order_id IN (SELECT order_id
        FROM order_line
        WHERE product_id = 5) )
    )
    AND
    (cust_id IN (SELECT cust_id
        FROM order
        WHERE order_id IN (SELECT order_id
            FROM order_line
            WHERE product_id = 6) )
    );
```
16. List the names of those customers who have ordered part number 5 and not ordered part number 6.

**Solution**

```sql
SELECT DISTINCT name
FROM customer
WHERE  (cust_id IN (SELECT cust_id
               FROM order
               WHERE order_id IN (SELECT order_id
               FROM order_line
               WHERE product_id = 5)  )
               )
AND
(cust_id NOT IN (SELECT cust_id
               FROM order
               WHERE order_id IN (SELECT order_id
               FROM order_line
               WHERE product_id = 6)  )
               );
```
17. List the names of those customers who have ordered either part number 5 or part number 6.

Solution

```
SELECT DISTINCT name
FROM customer
WHERE cust_id IN (SELECT cust_id
                  FROM order
                  WHERE order_id IN (SELECT order_id
                                      FROM order_line
                                      WHERE product_id = 5
                                      OR product_id = 6)
                  );
```
18. List the names of those customers who have ordered only part number 6.

**Solution**

```
SELECT DISTINCT name
FROM customer
WHERE (cust_id IN (SELECT cust_id
                   FROM order
                   WHERE order_id IN (SELECT order_id
                                       FROM order_line
                                       WHERE product_id = 6)
                   )
     )
AND
(cust_id NOT IN (SELECT cust_id
                  FROM order
                  WHERE order_id IN (SELECT order_id
                                      FROM order_line
                                      WHERE product_id <> 6)
                  )
  );
```