

Problem 1:

Create a query displaying the employee_id, start_date, end_date, and department_name using the old SQL join syntax (Where clause). Alias the departments table with d and the job_history table with jh. Order it by employee_id and start_date.

SQL script:

```
SELECT jh.employee_id, jh.start_date, jh.end_date, d.department_name
FROM job_history jh, departments d
WHERE jh.department_id = d.department_id
ORDER BY jh.employee_id, jh.start_date;
```

Console Output:

```
SQL> SELECT jh.employee_id, jh.start_date, jh.end_date, d.department_name
  2  FROM job_history jh, departments d
  3 WHERE jh.department_id = d.department_id
  4* ORDER BY jh.employee_id, jh.start_date;
EMPLOYEE_ID START_DATE END_DATE DEPARTMENT_NAME
----- -----
 101 21-SEP-89 27-OCT-93 Accounting
 101 28-OCT-93 15-MAR-97 Accounting
 102 13-JAN-93 24-JUL-98 IT
 114 24-MAR-98 31-DEC-99 Shipping
 122 01-JAN-99 31-DEC-99 Shipping
 176 24-MAR-98 31-DEC-98 Sales
 176 01-JAN-99 31-DEC-99 Sales
 200 17-SEP-87 17-JUN-93 Executive
 200 01-JUL-94 31-DEC-98 Executive
 201 17-FEB-96 19-DEC-99 Marketing

10 rows selected.
```

Problem 2:

Rewrite the previous query using the new SQL join syntax (From clause).

SQL script:

```
SELECT jh.employee_id, jh.start_date, jh.end_date, d.department_name
FROM job_history jh
INNER JOIN departments d
```

```
ON d.department_id = jh.department_id  
ORDER BY jh.employee_id, jh.start_date;
```

Console Output:

```
SQL> SELECT jh.employee_id, jh.start_date, jh.end_date, d.department_name  
  2  FROM job_history jh  
  3  INNER JOIN departments d  
  4  ON d.department_id = jh.department_id  
  5* ORDER BY jh.employee_id, jh.start_date;  
EMPLOYEE_ID START_DATE END_DATE DEPARTMENT_NAME  
-----  
101 21-SEP-89 27-OCT-93 Accounting  
101 28-OCT-93 15-MAR-97 Accounting  
102 13-JAN-93 24-JUL-98 IT  
114 24-MAR-98 31-DEC-99 Shipping  
122 01-JAN-99 31-DEC-99 Shipping  
176 24-MAR-98 31-DEC-98 Sales  
176 01-JAN-99 31-DEC-99 Sales  
200 17-SEP-87 17-JUN-93 Executive  
200 01-JUL-94 31-DEC-98 Executive  
201 17-FEB-96 19-DEC-99 Marketing  
  
10 rows selected.
```

Problem 3:

Rewrite the previous query using the following syntax variations:

- A.) Using the Using keyword
- B.) Using the Natural Join keywords

SQL script:

A).

```
SELECT jh.employee_id, jh.start_date, jh.end_date, d.department_name  
FROM job_history jh  
INNER JOIN departments d  
USING (department_id)  
ORDER BY jh.employee_id, jh.start_date;
```

B).

```
SELECT jh.employee_id, jh.start_date, jh.end_date, d.department_name  
FROM job_history jh
```

```
NATURAL INNER JOIN departments d  
ORDER BY jh.employee_id, jh.start_date;
```

Console Output:

A).

```
SQL> SELECT jh.employee_id, jh.start_date, jh.end_date, d.department_name  
  2  FROM job_history jh  
  3  INNER JOIN departments d USING (department_id)  
[ 4* ORDER BY jh.employee_id, jh.start_date;  
EMPLOYEE_ID START_DATE END_DATE DEPARTMENT_NAME  
-----  
101 21-SEP-89 27-OCT-93 Accounting  
101 28-OCT-93 15-MAR-97 Accounting  
102 13-JAN-93 24-JUL-98 IT  
114 24-MAR-98 31-DEC-99 Shipping  
122 01-JAN-99 31-DEC-99 Shipping  
176 24-MAR-98 31-DEC-98 Sales  
176 01-JAN-99 31-DEC-99 Sales  
200 17-SEP-87 17-JUN-93 Executive  
200 01-JUL-94 31-DEC-98 Executive  
201 17-FEB-96 19-DEC-99 Marketing  
  
10 rows selected.
```

B).

```
SQL> SELECT jh.employee_id, jh.start_date, jh.end_date, d.department_name  
  2  FROM job_history jh  
  3  NATURAL INNER JOIN departments d  
[ 4* ORDER BY jh.employee_id, jh.start_date;  
EMPLOYEE_ID START_DATE END_DATE DEPARTMENT_NAME  
-----  
101 21-SEP-89 27-OCT-93 Accounting  
101 28-OCT-93 15-MAR-97 Accounting  
102 13-JAN-93 24-JUL-98 IT  
114 24-MAR-98 31-DEC-99 Shipping  
122 01-JAN-99 31-DEC-99 Shipping  
176 24-MAR-98 31-DEC-98 Sales  
176 01-JAN-99 31-DEC-99 Sales  
200 17-SEP-87 17-JUN-93 Executive  
200 01-JUL-94 31-DEC-98 Executive  
201 17-FEB-96 19-DEC-99 Marketing  
  
10 rows selected.
```

Problem 4:

Show the count of records of table employees and table job_history (2 queries). How many records do you receive when you join the two tables using a cartesian join (either show query or briefly explain)?

1. Now join the two tables using an inner join on column employee_id. Show the employee_id, last_name, start_date and end_date. Alias the table employees with e and job_history with jh.
2. Finally, join the two tables using a natural inner join using the columns in the SELECT clause.
3. Explain the differences in output between the queries in 1) and 2).

SQL script:

```
SELECT COUNT(*) count  
FROM employees;
```

Console Output:

```
[SQL]> SELECT COUNT(*) count  
[ 2 FROM employees;  
  
          COUNT  
-----  
         107
```

SQL script:

```
SELECT COUNT(*) count  
FROM job_history;
```

Console Output:

```
SQL> SELECT COUNT(*) count  
[ 2* FROM job_history;  
          COUNT  
-----  
         10
```

SQL script:

```
SELECT COUNT(*) count  
FROM employees  
CROSS JOIN job_history;
```

Console Output:

```
[SQL]> SELECT COUNT(*) count  
[ 2  FROM employees  
[ 3  CROSS JOIN job_history;  
  
COUNT  
-----  
1070
```

SQL script:

```
SELECT e.employee_id, e.last_name, jh.start_date, jh.end_date  
FROM employees e  
INNER JOIN job_history jh  
ON e.employee_id = jh.employee_id;
```

Console Output:

```

SQL> SELECT e.employee_id, e.last_name, jh.start_date, jh.end_date
  2  FROM employees e
  3  INNER JOIN job_history jh
  4  ON e.employee_id = jh.employee_id;

EMPLOYEE_ID LAST_NAME          START_DATE END_DATE
-----  -----
      101 Kochhar            21-SEP-89 27-OCT-93
      101 Kochhar            28-OCT-93 15-MAR-97
      102 De Haan           13-JAN-93 24-JUL-98
      114 Raphaely          24-MAR-98 31-DEC-99
      122 Kaufling          01-JAN-99 31-DEC-99
      176 Taylor             24-MAR-98 31-DEC-98
      176 Taylor             01-JAN-99 31-DEC-99
      200 Whalen            17-SEP-87 17-JUN-93
      200 Whalen            01-JUL-94 31-DEC-98
      201 Hartstein          17-FEB-96 19-DEC-99

10 rows selected.

```

SQL script:

```

SELECT employee_id, e.last_name, jh.start_date, jh.end_date
FROM employees e
NATURAL INNER JOIN job_history jh;

```

Console Output:

```

SQL> SELECT employee_id, e.last_name, jh.start_date, jh.end_date
  2  FROM employees e
  3* NATURAL INNER JOIN job_history jh;

EMPLOYEE_ID LAST_NAME          START_DATE END_DATE
-----  -----
      176 Taylor             24-MAR-98 31-DEC-98

```

Answer:

Using the “NATURAL” keyword filters entries using all common columns in both tables and only keeps entries with identical values in those columns, respectively. Since the employees table and the job_history table have more than one common columns, the output is much less than the output of the former SQL query.

Problem 5:

Create a query showing employee_id, start_date, end_date, department_name and job_title. Display the dates columns in the format of mm/dd/yyyy. Order it by department_name and job_title. (join Departments, Job_History, Jobs)

SQL script:

```
SELECT jh.employee_id,
       TO_CHAR(jh.start_date, 'MM/DD/YYYY') start_date,
       TO_CHAR(jh.end_date, 'MM/DD/YYYY') end_date,
       d.department_name, j.job_title
  FROM job_history jh
 INNER JOIN departments d
      USING (department_id)
 INNER JOIN jobs j
      USING (job_id)
 ORDER BY department_name, job_title;
```

Console Output:

```
|SQL> SELECT jh.employee_id,
| 2   TO_CHAR(jh.start_date, 'MM/DD/YYYY') start_date,
| 3   TO_CHAR(jh.end_date, 'MM/DD/YYYY') end_date,
| 4   d.department_name, j.job_title
| 5  FROM job_history jh
| 6  INNER JOIN departments d
| 7  USING (department_id)
| 8  INNER JOIN jobs j
| 9  USING (job_id)
| 10* ORDER BY department_name, job_title;
EMPLOYEE_ID START_DATE END_DATE  DEPARTMENT_NAME          JOB_TITLE
-----  -----  -----  -----
        101 10/28/1993 03/15/1997 Accounting          Accounting Manager
        101 09/21/1989 10/27/1993 Accounting          Public Accountant
        200 09/17/1987 06/17/1993 Executive           Administration Assistant
        200 07/01/1994 12/31/1998 Executive           Public Accountant
        102 01/13/1993 07/24/1998 IT                  Programmer
        201 02/17/1996 12/19/1999 Marketing          Marketing Representative
        176 01/01/1999 12/31/1999 Sales               Sales Manager
        176 03/24/1998 12/31/1998 Sales               Sales Representative
        122 01/01/1999 12/31/1999 Shipping            Stock Clerk
        114 03/24/1998 12/31/1999 Shipping            Stock Clerk
10 rows selected.
```

Problem 6:

Create a query showing job_title, min_salary, max_salary from the jobs table and employee_id, start_date from the job_history table. Make sure to display all records from the jobs table. Order it by job title.

Modify the previous query to find out how many job_titles are not used in the job_history table.

SQL script:

```
SELECT j.job_title, j.min_salary, j.max_salary, jh.employee_id, jh.start_date  
FROM jobs j  
LEFT OUTER JOIN job_history jh  
USING (job_id)  
ORDER BY j.job_title;
```

Console Output:

```
SQL> SELECT j.job_title, j.min_salary, j.max_salary, jh.employee_id, jh.start_date  
  2  FROM jobs j  
  3  LEFT OUTER JOIN job_history jh  
  4  USING (job_id)  
  5* ORDER BY j.job_title;  


| JOB_TITLE                       | MIN_SALARY | MAX_SALARY | EMPLOYEE_ID | START_DATE |
|---------------------------------|------------|------------|-------------|------------|
| Accountant                      | 4200       | 9000       |             |            |
| Accounting Manager              | 8200       | 16000      | 101         | 28-OCT-93  |
| Administration Assistant        | 3000       | 6000       | 200         | 17-SEP-87  |
| Administration Vice President   | 15000      | 30000      |             |            |
| Finance Manager                 | 8200       | 16000      |             |            |
| Human Resources Representative  | 4000       | 9000       |             |            |
| Marketing Manager               | 9000       | 15000      |             |            |
| Marketing Representative        | 4000       | 9000       | 201         | 17-FEB-96  |
| President                       | 20000      | 40000      |             |            |
| Programmer                      | 4000       | 10000      | 102         | 13-JAN-93  |
| Public Accountant               | 4200       | 9000       | 101         | 21-SEP-89  |
| Public Accountant               | 4200       | 9000       | 200         | 01-JUL-94  |
| Public Relations Representative | 4500       | 10500      |             |            |
| Purchasing Clerk                | 2500       | 5500       |             |            |
| Purchasing Manager              | 8000       | 15000      |             |            |
| Sales Manager                   | 10000      | 20000      | 176         | 01-JAN-99  |
| Sales Representative            | 6000       | 12000      | 176         | 24-MAR-98  |
| Shipping Clerk                  | 2500       | 5500       |             |            |
| Stock Clerk                     | 2000       | 5000       | 122         | 01-JAN-99  |
| Stock Clerk                     | 2000       | 5000       | 114         | 24-MAR-98  |
| Stock Manager                   | 5500       | 8500       |             |            |



21 rows selected.


```

SQL script:

```
SELECT j.job_title, j.min_salary, j.max_salary, jh.employee_id, jh.start_date
FROM jobs j
LEFT OUTER JOIN job_history jh
USING (job_id)
WHERE jh.employee_id IS NULL
ORDER BY j.job_title;
```

Console Output:

```
SQL> SELECT j.job_title, j.min_salary, j.max_salary, jh.employee_id, jh.start_date
  2  FROM jobs j
  3  LEFT OUTER JOIN job_history jh
  4  USING (job_id)
  5  WHERE jh.employee_id IS NULL
  6* ORDER BY j.job_title;

JOB_TITLE          MIN_SALARY MAX_SALARY EMPLOYEE_ID START_DATE
-----          -----      -----        -----
Accountant           4200       9000
Administration Vice President    15000      30000
Finance Manager        8200      16000
Human Resources Representative  4000       9000
Marketing Manager        9000      15000
President             20000      40000
Public Relations Representative  4500      10500
Purchasing Clerk         2500       5500
Purchasing Manager        8000      15000
Shipping Clerk            2500       5500
Stock Manager            5500       8500

11 rows selected.
```

Answer:

11 job titles are not used.

Problem 7:

Create a query showing employee_id, last_name, and salary from table employees and min_salary and max_salary from table jobs. Filter the data where the salary matches the midpoint of min_salary and max_salary. Sort the resulting data set by job_id and last_name.

SQL script:

```
SELECT e.employee_id, e.last_name, e.salary, j.min_salary, j.max_salary
FROM employees e
INNER JOIN jobs j
USING (job_id)
WHERE e.salary = (j.min_salary + j.max_salary) / 2
ORDER BY job_id, e.last_name;
```

Console Output:

```
SQL> SELECT e.employee_id, e.last_name, e.salary, j.min_salary, j.max_salary
  2  FROM employees e
  3  INNER JOIN jobs j
  4  USING (job_id)
  5  WHERE e.salary = (j.min_salary + j.max_salary) / 2
[ 6* ORDER BY job_id, e.last_name;
EMPLOYEE_ID LAST_NAME          SALARY MIN_SALARY MAX_SALARY
-----  -----
      203 Mavris            6500    4000      9000
      152 Hall              9000    6000     12000
      158 McEwen            9000    6000     12000
      192 Bell              4000    2500      5500
      141 Rajs              3500    2000      5000
```

Problem 8:

Create a query showing the last_name from table employees and the following derived expression:

Concatenate the area code (first 3 numbers of phone_number), the city, and the country_name using a hyphen in between: area_code-city-country_name

Sort the resulting dataset by country_name, city, and last_name.

SQL script:

```
SELECT e.last_name,
SUBSTR(e.phone_number, 1, 3) || '-' || l.city || '-' || c.country_name area_city_country
FROM employees e
INNER JOIN departments d ON d.department_id = e.department_id
INNER JOIN locations l ON l.location_id = d.location_id
INNER JOIN countries c ON c.country_id = l.country_id
```

```
ORDER BY c.country_name, l.city, e.last_name;
```

Console Output:

```
SQL> SELECT e.last_name,
  2  SUBSTR(e.phone_number, 1, 3) || '-' || l.city || '-' || c.country_name area_city_country
  3  FROM employees e
  4  INNER JOIN departments d ON d.department_id = e.department_id
  5  INNER JOIN locations l ON l.location_id = d.location_id
  6  INNER JOIN countries c ON c.country_id = l.country_id
  7* ORDER BY c.country_name, l.city, e.last_name;
LAST_NAME          AREA_CITY_COUNTRY
-----|-----
Fay                603-Toronto-Canada
Hartstein          515-Toronto-Canada
Baer               515-Munich-Germany
Mavris              515-London-United Kingdom
Abel                011-Oxford-United Kingdom
Ande                011-Oxford-United Kingdom
Banda               011-Oxford-United Kingdom
Bates               011-Oxford-United Kingdom
Bernstein           011-Oxford-United Kingdom
Bloom               011-Oxford-United Kingdom
Cambrault            011-Oxford-United Kingdom
Cambrault           011-Oxford-United Kingdom
Doran               011-Oxford-United Kingdom
Errazuriz            011-Oxford-United Kingdom
Fox                 011-Oxford-United Kingdom
Greene              011-Oxford-United Kingdom
Hall                011-Oxford-United Kingdom
Hutton              011-Oxford-United Kingdom
Johnson             011-Oxford-United Kingdom
King                011-Oxford-United Kingdom
Kumar               011-Oxford-United Kingdom
Lee                 011-Oxford-United Kingdom
```

Ladwig	650-South San Francisco-United States of America
Landry	650-South San Francisco-United States of America
Mallin	650-South San Francisco-United States of America
Markle	650-South San Francisco-United States of America
Marlow	650-South San Francisco-United States of America
Matos	650-South San Francisco-United States of America
McCain	650-South San Francisco-United States of America
Mikkilineni	650-South San Francisco-United States of America
Mourgos	650-South San Francisco-United States of America
Nayer	650-South San Francisco-United States of America
OConnell	650-South San Francisco-United States of America
Olson	650-South San Francisco-United States of America
Patel	650-South San Francisco-United States of America
Perkins	650-South San Francisco-United States of America
Philtanker	650-South San Francisco-United States of America
Rajs	650-South San Francisco-United States of America
Rogers	650-South San Francisco-United States of America
Sarchand	650-South San Francisco-United States of America
Seo	650-South San Francisco-United States of America
Stiles	650-South San Francisco-United States of America
Sullivan	650-South San Francisco-United States of America
Taylor	650-South San Francisco-United States of America
Vargas	650-South San Francisco-United States of America
Vollman	650-South San Francisco-United States of America
Walsh	650-South San Francisco-United States of America
Weiss	650-South San Francisco-United States of America
Austin	590-Southlake-United States of America
Ernst	590-Southlake-United States of America
Hunold	590-Southlake-United States of America
Lorentz	590-Southlake-United States of America
Pataballa	590-Southlake-United States of America

106 rows selected.