



Problem 1:

```
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;
```

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_DAT	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:

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

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

EMPLOYEE_ID	START_DAT	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:

```
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;
```

```
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 USING(department_id)
5 ORDER BY jh.employee_id, jh.start_date;
```

EMPLOYEE_ID	START_DAT	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.



```
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;
```

The screenshot shows a window titled "Run SQL Command Line". The command entered is:

```
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;
```

The results are displayed in a table with the following columns: EMPLOYEE_ID, START_DAT, END_DATE, and DEPARTMENT_NAME. The data is as follows:

EMPLOYEE_ID	START_DAT	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:

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

```
SQL> SELECT COUNT(*) FROM employees;  
  
COUNT(*)  
-----  
107  
  
SQL> SELECT COUNT(*) FROM job_history;  
  
COUNT(*)  
-----  
10
```

```
SELECT COUNT(*) FROM employees, job_history;
```

```
SQL> SELECT COUNT(*) FROM employees, job_history;  
  
COUNT(*)  
-----  
1070
```



```
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;
```

```
Run SQL Command Line

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

EMPLOYEE_ID LAST_NAME          START_DAT  END_DATE
-----
101 Kochhar                28-OCT-93  15-MAR-97
101 Kochhar                21-SEP-89  27-OCT-93
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                01-JAN-99  31-DEC-99
176 Taylor                24-MAR-98  31-DEC-98
200 Whalen                01-JUL-94  31-DEC-98
200 Whalen                17-SEP-87  17-JUN-93
201 Hartstein            17-FEB-96  19-DEC-99

10 rows selected.
```

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

```
Run SQL Command Line

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

EMPLOYEE_ID LAST_NAME          START_DAT  END_DATE
-----
176 Taylor                24-MAR-98  31-DEC-98
```

NATURAL INNER JOIN joins on all common columns, in this case on employee_id, job_id, and department_id. This is more restrictive and hence results in fewer records compared to the join condition on just column employee_id.



Below is the query using explicit join conditions on those three columns:

```
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
AND e.job_id = jh.job_id
AND e.department_id = jh.department_id;
```

The screenshot shows a window titled "Run SQL Command Line" with a blue header bar. The window contains the following SQL query and its output:

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

The output is displayed in a table format with the following columns: EMPLOYEE_ID, LAST_NAME, START_DAT, and END_DATE. The data row shows employee 176, Taylor, with a start date of 24-MAR-98 and an end date of 31-DEC-98.

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



Problem 5:

```
SELECT jh.employee_id, TO_CHAR(jh.start_date,'mm/dd/yyyy') AS start_date,  
       TO_CHAR(jh.end_date, 'mm/dd/yyyy') AS end_date, d.department_name, j.job_title  
FROM departments d  
     INNER JOIN job_history jh ON d.department_id = jh.department_id  
     INNER JOIN Jobs j ON jh.job_ID = j.job_ID  
ORDER BY d.department_name, j.job_title;
```

Run SQL Command Line

```
SQL> SELECT jh.employee_id, TO_CHAR(jh.start_date,'mm/dd/yyyy') AS start_date,  
2      TO_CHAR(jh.end_date, 'mm/dd/yyyy') AS end_date, d.department_name, j.job_title  
3 FROM departments d  
4     INNER JOIN job_history jh ON d.department_id = jh.department_id  
5     INNER JOIN Jobs j ON jh.job_ID = j.job_ID  
6 ORDER BY d.department_name, j.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 Assistan
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 Representativ
176	01/01/1999	12/31/1999	Sales	Sales Manager
176	03/24/1998	12/31/1998	Sales	Sales Representative
114	03/24/1998	12/31/1999	Shipping	Stock Clerk
122	01/01/1999	12/31/1999	Shipping	Stock Clerk

10 rows selected.



Problem 6:

```
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 ON j.job_id = jh.job_id
ORDER BY j.job_title;
```

Run SQL Command Line

```
SQL> SELECT j.job_title, j.min_salary, j.max_Salary,   jh.employee_id, jh.start_date
2 FROM jobs j LEFT OUTER JOIN job_history jh ON j.job_id = jh.job_id
3 ORDER BY j.job_title;
```

JOB_TITLE	MIN_SALARY	MAX_SALARY	EMPLOYEE_ID	START_DAT
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	200	01-JUL-94
Public Accountant	4200	9000	101	21-SEP-89
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.



```
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 ON j.job_id = jh.job_id
WHERE jh.employee_id IS NULL
ORDER BY j.job_title;
```

Run SQL Command Line

```
SQL> SELECT j.job_title, j.min_salary, j.max_Salary, jh.employee_id, jh.start_date
2 FROM jobs j LEFT OUTER JOIN job_history jh ON j.job_id = jh.job_id
3 WHERE jh.employee_id IS NULL
4 ORDER BY j.job_title;
```

JOB_TITLE	MIN_SALARY	MAX_SALARY	EMPLOYEE_ID	START_DAT
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.

Alternatively:

```
SELECT count(*)
FROM jobs j LEFT OUTER JOIN
      job_history jh ON j.job_id = jh.job_id
WHERE jh.employee_id IS NULL
ORDER BY j.job_title;
```

Run SQL Command Line

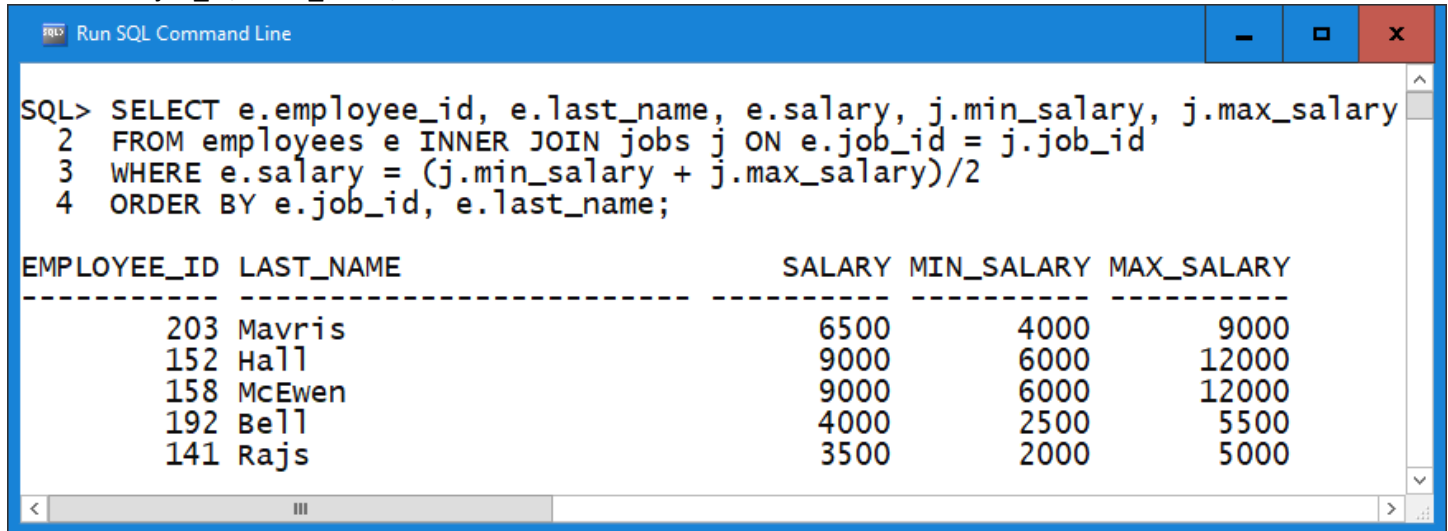
```
SQL> SELECT count(*)
2 FROM jobs j LEFT OUTER JOIN
3      job_history jh ON j.job_id = jh.job_id
4 WHERE jh.employee_id IS NULL
5 ORDER BY j.job_title;
```

COUNT(*)
11



Problem 7:

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

```
SELECT e.last_name,
SUBSTR(e.phone_number,1,3) || '-' || l.city || '-' || c.country_name area_code_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;
```



```
Run SQL Command Line

SQL> SELECT e.last_name,
2 SUBSTR(e.phone_number,1,3) || '-' || l.city || '-' || c.country_name area_code_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_CODE_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
```

```
Run SQL Command Line

LAST_NAME          AREA_CODE_CITY_COUNTRY
-----
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.
```