

Name: Urooba Gohar

Roll No: 22P-9216

Section: BSCS-6A

Database Systems Lab 5

Task 1:

Write a query to list the name, job title, department name, and salary of the employees in ascending order of their department.

Ans:

To display the names, job id, department name, salary of employees of their department, we use inner join and for ascending order we use “order by”.

Following is the query:

```
--22P-9216
select e.first_name, e.job_id, d.department_name, e.salary from employees e
join departments d on e.department_id=d.department_id
order by d.department_name asc
```

Following is the result:

| | FIRST_NAME | JOB_ID | DEPARTMENT_NAME | SALARY |
|----|-------------|---------------|-----------------|--------|
| 1 | William | SENAC_ACCOUNT | Accounting | 8300 |
| 2 | Shelley | SENAC_MGR | Accounting | 12008 |
| 3 | Jennifer | SENAD_ASST | Administration | 4400 |
| 4 | Steven | SENAD_PRES | Executive | 24000 |
| 5 | Lex | SENAD_VP | Executive | 17000 |
| 6 | Neena | SENAD_VP | Executive | 17000 |
| 7 | Nancy | SENFI_MGR | Finance | 12008 |
| 8 | Daniel | SENFI_ACCOUNT | Finance | 9000 |
| 9 | Luis | SENFI_ACCOUNT | Finance | 6900 |
| 10 | Ismael | SENFI_ACCOUNT | Finance | 7700 |
| 11 | Jose Manuel | SENFI_ACCOUNT | Finance | 7800 |
| 12 | John | SENFI_ACCOUNT | Finance | 8200 |
| 13 | Susan | SENHR_REP | Human Resources | 6500 |
| 14 | Alexander | SENIT_PROG | IT | 9000 |
| 15 | Bruce | SENIT_PROG | IT | 6000 |
| 16 | Diana | SENIT_PROG | IT | 4200 |
| 17 | Valli | SENIT_PROG | IT | 4800 |
| 18 | David | SENIT_PROG | IT | 4800 |

Task 2:

Write a query to list the departments where at least two employees are working.

Ans:

To display the list of departments where at least 2 employees are working, we use inner join and then group them by the department name.

Following is the query:

```
--22P-9216
select d.department_name, count(e.employee_id) from employees e
join departments d on e.department_id=d.department_id
group by d.department_name
having count(e.employee_id)>= 2;
```

Following is the result:

| DEPARTMENT_NAME | COUNT(E.EMPLOYEE_ID) |
|-----------------|----------------------|
| 1 Accounting | 2 |
| 2 Purchasing | 6 |
| 3 IT | 5 |
| 4 Executive | 3 |
| 5 Shipping | 45 |
| 6 Sales | 34 |
| 7 Finance | 6 |
| 8 Marketing | 2 |

Task 3:

Fetch all records where the employee's salary is less than the lowest salary in the company.

Ans:

To find the records where the employees salary is less than the lowest salary, we use subquery.

Following is the query:

```
--22P-9216
select * from employees
where salary<(select min(salary) from employees);
```

Following is the result:

| EMPLOYEE... | FIRST_NA... | LAST_NAME | EMAIL | PHONE_N... | HIRE_DATE | JOB_ID | SALARY | COMMISS... | MANAGER... | DEPARTM... |
|-------------|-------------|-----------|-------|------------|-----------|--------|--------|------------|------------|------------|
|-------------|-------------|-----------|-------|------------|-----------|--------|--------|------------|------------|------------|

Task 4:

Write a query to list the name, job title, annual salary, department name, and city of employees who earn 60000 or more annually and are not working as ANALYST.

Ans:

To list the name, job id, annual salary, department name, and city of employees who earn 60k or more and are not working as analyst, we can use inner join.

Following is the query:

```
--22P-9216
select e.first_name, e.job_id, e.salary*12, d.department_name, l.city
from employees e
join departments d on e.department_id=d.department_id
join locations l on d.location_id=l.location_id
where(e.salary*12)>=60000 and e.job_id not like 'analyst'
```

Following is the result:

| | FIRST_NAME | JOB_ID | E.SALARY*12 | DEPARTMENT_NAME | CITY |
|----|-------------|---------------|-------------|-----------------|---------------------|
| 1 | Alexander | SENIT_PROG | 108000 | IT | Southlake |
| 2 | Bruce | SENIT_PROG | 72000 | IT | Southlake |
| 3 | Payam | SENST_MAN | 94800 | Shipping | South San Francisco |
| 4 | Adam | SENST_MAN | 98400 | Shipping | South San Francisco |
| 5 | Matthew | SENST_MAN | 96000 | Shipping | South San Francisco |
| 6 | Shanta | SENST_MAN | 78000 | Shipping | South San Francisco |
| 7 | Kevin | SENST_MAN | 69600 | Shipping | South San Francisco |
| 8 | William | SENAC_ACCOUNT | 99600 | Accounting | Seattle |
| 9 | Shelley | SENAC_MGR | 144096 | Accounting | Seattle |
| 10 | John | SENFI_ACCOUNT | 98400 | Finance | Seattle |
| 11 | Ismael | SENFI_ACCOUNT | 92400 | Finance | Seattle |
| 12 | Luis | SENFI_ACCOUNT | 82800 | Finance | Seattle |
| 13 | Nancy | SENFI_MGR | 144096 | Finance | Seattle |
| 14 | Daniel | SENFI_ACCOUNT | 108000 | Finance | Seattle |
| 15 | Jose Manuel | SENFI_ACCOUNT | 93600 | Finance | Seattle |
| 16 | Neena | SENAD_VP | 204000 | Executive | Seattle |
| 17 | Steven | SENAD_PRES | 288000 | Executive | Seattle |
| 18 | Lex | SENAD_VP | 204000 | Executive | Seattle |

Task 5:

Write a query to print details of the employees who are also Managers.

Ans:

To find the details of employee who are also managers, we use inner join.

Following is the query:

```
--22P-9216
select distinct m.*
from employees e
join employees m on e.manager_id=m.employee_id
```

Following is the result:

| | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE | JOB_ID | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
|----|-------------|------------|-----------|----------|--------------------|-----------|------------|--------|----------------|------------|---------------|
| 1 | 101 | Neena | Kochhar | NKOCHHAR | 515.123.4568 | 21-SEP-05 | SENAD_VP | 17000 | (null) | 100 | 90 |
| 2 | 146 | Karen | Partners | KPARTNER | 011.44.1344.467268 | 05-JAN-05 | SENSA_MAN | 13500 | 0.3 | 100 | 80 |
| 3 | 100 | Steven | King | SKING | 515.123.4567 | 17-JUN-03 | SENAD_PRES | 24000 | (null) | (null) | 90 |
| 4 | 102 | Lex | De Haan | LDEHAAN | 515.123.4569 | 13-JAN-01 | SENAD_VP | 17000 | (null) | 100 | 90 |
| 5 | 114 | Den | Raphaely | DRAPHEAL | 515.127.4561 | 07-DEC-02 | SENFU_MAN | 11000 | (null) | 100 | 30 |
| 6 | 124 | Kevin | Mourgos | KMOURGOS | 650.123.5234 | 16-NOV-07 | SENST_MAN | 5800 | (null) | 100 | 50 |
| 7 | 145 | John | Russell | JRUSSEL | 011.44.1344.429268 | 01-OCT-04 | SENSA_MAN | 14000 | 0.4 | 100 | 80 |
| 8 | 149 | Eleni | Zlotkey | EZLOTKEY | 011.44.1344.429018 | 29-JAN-08 | SENSA_MAN | 10500 | 0.2 | 100 | 80 |
| 9 | 147 | Alberto | Errazuriz | AERRAZUR | 011.44.1344.429278 | 10-MAR-05 | SENSA_MAN | 12000 | 0.3 | 100 | 80 |
| 10 | 201 | Michael | Hartstein | MHARTSTE | 515.123.5555 | 17-FEB-04 | SENMK_MAN | 13000 | (null) | 100 | 20 |
| 11 | 103 | Alexander | Hunold | AHUNOLD | 590.423.4567 | 03-JAN-06 | SENIT_PROG | 9000 | (null) | 102 | 60 |
| 12 | 205 | Shelley | Higgins | SHIGGINS | 515.123.8080 | 07-JUN-02 | SENAC_MGR | 12008 | (null) | 101 | 110 |
| 13 | 108 | Nancy | Greenberg | NGREENBE | 515.124.4569 | 17-AUG-02 | SENFI_MGR | 12008 | (null) | 101 | 100 |
| 14 | 121 | Adam | Fripp | AFRIPP | 650.123.2234 | 10-APR-05 | SENST_MAN | 8200 | (null) | 100 | 50 |
| 15 | 122 | Payam | Kaufling | PKAUFLIN | 650.123.3234 | 01-MAY-03 | SENST_MAN | 7900 | (null) | 100 | 50 |
| 16 | 123 | Shanta | Vollman | SVOLLMAN | 650.123.4234 | 10-OCT-05 | SENST_MAN | 6500 | (null) | 100 | 50 |
| 17 | 148 | Gerald | Cambrault | GCAMBRAU | 011.44.1344.619268 | 15-OCT-07 | SENSA_MAN | 11000 | 0.3 | 100 | 80 |

Task 6:

List department number and department name for all departments that have no employees.

Ans:

To find dept number and dept name for all departments with no employees, we use left join.

Following is the query:

```
--22P-9216
select d.department_id, d.department_name
from departments d
left join employees e on d.department_id=e.department_id
where e.department_id is null
```

Following is the result:

| | DEPARTMENT_ID | DEPARTMENT_NAME |
|----|---------------|----------------------|
| 1 | 120 | Treasury |
| 2 | 130 | Corporate Tax |
| 3 | 140 | Control And Credit |
| 4 | 150 | Shareholder Services |
| 5 | 160 | Benefits |
| 6 | 170 | Manufacturing |
| 7 | 180 | Construction |
| 8 | 190 | Contracting |
| 9 | 200 | Operations |
| 10 | 210 | IT Support |
| 11 | 220 | NOC |
| 12 | 230 | IT Helpdesk |
| 13 | 240 | Government Sales |
| 14 | 250 | Retail Sales |
| 15 | 260 | Recruiting |
| 16 | 270 | Payroll |

Task 7:

Display employee name, salary, and department name where all employees match their department, including employees with no assigned department.

Ans:

To list the employee names, salary and department name where all employees match their department having employees with no department, we use left join.

Following is the query:

```
--22P-9216
select e.first_name, d.department_name, e.salary from employees e
left join departments d on e.department_id=d.department_id
```

Following is the result:

| | FIRST_NAME | DEPARTMENT_NAME | SALARY |
|----|------------|-----------------|--------|
| 1 | Jennifer | Administration | 4400 |
| 2 | Pat | Marketing | 6300 |
| 3 | Michael | Marketing | 13000 |
| 4 | Karen | Purchasing | 2500 |
| 5 | Guy | Purchasing | 2600 |
| 6 | Sigal | Purchasing | 2800 |
| 7 | Shelli | Purchasing | 2900 |
| 8 | Alexander | Purchasing | 3100 |
| 9 | Den | Purchasing | 11000 |
| 10 | Susan | Human Resources | 6500 |
| 11 | Douglas | Shipping | 2600 |
| 12 | Donald | Shipping | 2600 |
| 13 | Kevin | Shipping | 3000 |
| 14 | Alana | Shipping | 3100 |
| 15 | Vance | Shipping | 2800 |
| 16 | Samuel | Shipping | 3200 |
| 17 | Britney | Shipping | 3900 |
| 18 | Sarah | Shipping | 4000 |

Task 8:

Display the name, job title, department name, and city of employees who are working in departments located in cities without a state province.

Ans:

To display the name, job title, dept name, city of employees who are working in departments located in cities without a state province, we use inner join.

Following is the query:

```
--22P-9216
select e.first_name, e.job_id, d.department_name, l.city
from employees e
join departments d on e.department_id=d.department_id
join locations l on d.location_id=l.location_id
where l.state_province is null
```

Following is the result:

| | FIRST_NAME | JOB_ID | DEPARTMENT_NAME | CITY |
|---|------------|-----------|-----------------|--------|
| 1 | Susan | SENHR_REP | Human Resources | London |

Task 9:

Write an SQL query to show records from one table that do not exist in another table.

Ans:

We use left join to show records from one table that do not exist in another table. Following is the query:

```
--22P-9216
select e.employee_id, e.first_name, e.job_id
from employees e
left join employee_backup eb on e.employee_id=eb.employee_id
where eb.employee_id is null
```

Following is the result:

| | EMPLOYEE_ID | FIRST_NAME | JOB_ID |
|----|-------------|------------|--------------|
| 1 | 162 | Clara | SENSA_REP |
| 2 | 121 | Adam | SENST_MAN |
| 3 | 133 | Jason | SENST_CLERK |
| 4 | 136 | Hazel | SENST_CLERK |
| 5 | 154 | Nanette | SENSA_REP |
| 6 | 196 | Alana | SENSH_CLERK |
| 7 | 146 | Karen | SENSA_MAN |
| 8 | 104 | Bruce | SENIT_PROG |
| 9 | 184 | Nandita | SENSH_CLERK |
| 10 | 172 | Elizabeth | SENSA_REP |
| 11 | 197 | Kevin | SENSH_CLERK |
| 12 | 150 | Peter | SENSA_REP |
| 13 | 142 | Curtis | SENST_CLERK |
| 14 | 205 | Shelley | SENAC_MGR |
| 15 | 143 | Randall | SENST_CLERK |
| 16 | 103 | Alexander | SENIT_PROG |
| 17 | 191 | Randall | SENSH_CLERK |
| 18 | 119 | Karen | SENPHI_CLERK |

Task 10:

Display all employees who belong to the US but not to the state of Washington.

Ans:

We use inner join here to display the employees who belong to US but not to Washington. Following is the query:

```

--22P-9216
select e.* from employees e
join departments d on e.department_id=d.department_id
join locations l on d.location_id=l.location_id
join countries c on l.country_id=c.country_id
where c.country_name='United States of America' and l.state_province<>'Washington'

```

Following is the result:

| | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE | JOB_ID | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
|----|-------------|------------|-------------|----------|--------------|-----------|-------------|--------|----------------|------------|---------------|
| 1 | 103 | Alexander | Hunold | AHUNOLD | 590.423.4567 | 03-JAN-06 | SENIT_PROG | 9000 | (null) | 102 | 60 |
| 2 | 104 | Bruce | Ernst | BERNST | 590.423.4568 | 21-MAY-07 | SENIT_PROG | 6000 | (null) | 103 | 60 |
| 3 | 105 | David | Austin | DAUSTIN | 590.423.4569 | 25-JUN-05 | SENIT_PROG | 4800 | (null) | 103 | 60 |
| 4 | 106 | Valli | Pataballa | VPATABAL | 590.423.4560 | 05-FEB-06 | SENIT_PROG | 4800 | (null) | 103 | 60 |
| 5 | 107 | Diana | Lorentz | DLORENTZ | 590.423.5567 | 07-FEB-07 | SENIT_PROG | 4200 | (null) | 103 | 60 |
| 6 | 120 | Matthew | Weiss | MWEISS | 650.123.1234 | 18-JUL-04 | SENST_MAN | 8000 | (null) | 100 | 50 |
| 7 | 121 | Adam | Fripp | AFRIPP | 650.123.2234 | 10-APR-05 | SENST_MAN | 8200 | (null) | 100 | 50 |
| 8 | 122 | Payam | Kaufling | PKAUFLIN | 650.123.3234 | 01-MAY-03 | SENST_MAN | 7900 | (null) | 100 | 50 |
| 9 | 123 | Shanta | Vollman | SVOLLMAN | 650.123.4234 | 10-OCT-05 | SENST_MAN | 6500 | (null) | 100 | 50 |
| 10 | 124 | Kevin | Mourgos | KMOURGOS | 650.123.5234 | 16-NOV-07 | SENST_MAN | 5800 | (null) | 100 | 50 |
| 11 | 125 | Julia | Nayer | JNAYER | 650.124.1214 | 16-JUL-05 | SENST_CLERK | 3200 | (null) | 120 | 50 |
| 12 | 126 | Irene | Mikkilineni | IMIKKILI | 650.124.1224 | 28-SEP-06 | SENST_CLERK | 2700 | (null) | 120 | 50 |
| 13 | 127 | James | Landry | JLANDRY | 650.124.1334 | 14-JAN-07 | SENST_CLERK | 2400 | (null) | 120 | 50 |
| 14 | 128 | Steven | Markle | SMARKLE | 650.124.1434 | 08-MAR-08 | SENST_CLERK | 2200 | (null) | 120 | 50 |
| 15 | 129 | Laura | Bissot | LBISSOT | 650.124.5234 | 20-AUG-05 | SENST_CLERK | 3300 | (null) | 121 | 50 |
| 16 | 130 | Mozhe | Atkinson | MATKINSO | 650.124.6234 | 30-OCT-05 | SENST_CLERK | 2800 | (null) | 121 | 50 |
| 17 | 131 | James | Marlow | JAMRLOW | 650.124.7234 | 16-FEB-05 | SENST_CLERK | 2500 | (null) | 121 | 50 |

Task 11:

Write a query to list the name, job title, department name, and location of employees who have a salary higher than the average salary in their department.

Ans:

We use inner join to list the name, job id, department name, location of employees who have a salary higher than the average salary of their department.

Following is the query:

```

--22P-9216
select e.first_name, e.job_id, d.department_name, l.location_id
from employees e join departments d on e.department_id=d.department_id
join locations l on d.location_id=l.location_id
where e.salary>(
select avg(e1.salary) from employees e1 where e1.department_id=e.department_id)

```

Following is the result:

| SQL All Rows Fetched: 38 in 0.019 seconds | | | |
|---|---------------|-----------------|-------------|
| FIRST_NAME | JOB_ID | DEPARTMENT_NAME | LOCATION_ID |
| 1 Bruce | SENIT_PROG | IT | 1400 |
| 2 Alexander | SENIT_PROG | IT | 1400 |
| 3 Britney | SENSH_CLERK | Shipping | 1500 |
| 4 Sarah | SENSH_CLERK | Shipping | 1500 |
| 5 Jennifer | SENSH_CLERK | Shipping | 1500 |
| 6 Kelly | SENSH_CLERK | Shipping | 1500 |
| 7 Alexis | SENSH_CLERK | Shipping | 1500 |
| 8 Nandita | SENSH_CLERK | Shipping | 1500 |
| 9 Trena | SENST_CLERK | Shipping | 1500 |
| 10 Renske | SENST_CLERK | Shipping | 1500 |
| 11 Kevin | SENST_MAN | Shipping | 1500 |
| 12 Shanta | SENST_MAN | Shipping | 1500 |
| 13 Payam | SENST_MAN | Shipping | 1500 |
| 14 Adam | SENST_MAN | Shipping | 1500 |
| 15 Matthew | SENST_MAN | Shipping | 1500 |
| 16 Den | SENPU_MAN | Purchasing | 1700 |
| 17 Steven | SENAD_PRES | Executive | 1700 |
| 18 Daniel | SENEI_ACCOUNT | Finance | 1700 |

Task 12:

Write a query to list employees who have changed their job title at least once in their job history.

Ans:

We have used inner join to list employees who have changed their job id atleast once in their job history.

Following is the query:

```
--22P-9216
select e.employee_id, e.first_name from employees e
join job_history j on e.employee_id=j.employee_id
group by e.employee_id, e.first_name
having count(distinct j.job_id)>1;
```

Following is the result:

| EMPLOYEE_ID | FIRST_NAME |
|--------------|------------|
| 101 Neena | |
| 200 Jennifer | |
| 176 Jonathon | |

Task 13:

List employees who work in the same department as their managers.

Ans:

We have used self join to list employees who work in the same department as their managers.

Following is the query:

```
sheet Query Builder
--22P-9216
select e.employee_id, e.first_name from employees e
join employees m on e.manager_id=m.employee_id
where e.department_id=m.department_id
```

Following is the result:

| | EMPLOYEE_ID | FIRST_NAME |
|----|-------------|-------------|
| 1 | 102 | Lex |
| 2 | 101 | Neena |
| 3 | 107 | Diana |
| 4 | 106 | Valli |
| 5 | 105 | David |
| 6 | 104 | Bruce |
| 7 | 113 | Luis |
| 8 | 112 | Jose Manuel |
| 9 | 111 | Ismael |
| 10 | 110 | John |
| 11 | 109 | Daniel |
| 12 | 119 | Karen |
| 13 | 118 | Guy |
| 14 | 117 | Sigal |
| 15 | 116 | Shelli |
| 16 | 115 | Alexander |
| 17 | 183 | Girard |
| 18 | 182 | Martha |

Task 14:

Write a query to list the name, department name, and location of employees who work in the same country as their department location.

Ans:

We use inner join to find the name, department name, location of employees who work in the same country as their department location.

Following is the query:

```
--22P-9216
select e.first_name, d.department_name, l.country_id from employees e
join departments d on e.department_id=d.department_id
join locations l on d.location_id=l.location_id
join countries c on l.country_id=c.country_id
where l.country_id=c.country_id
```

Following is the result:

| | FIRST_NAME | DEPARTMENT_NAME | COUNTRY_ID |
|----|------------|------------------|------------|
| 1 | Ellen | Sales | UK |
| 2 | Sundar | Sales | UK |
| 3 | Mozhe | Shipping | US |
| 4 | David | IT | US |
| 5 | Hermann | Public Relations | DE |
| 6 | Shelli | Purchasing | US |
| 7 | Amit | Sales | UK |
| 8 | Elizabeth | Sales | UK |
| 9 | Sarah | Shipping | US |
| 10 | David | Sales | UK |
| 11 | Laura | Shipping | US |
| 12 | Harrison | Sales | UK |
| 13 | Alexis | Shipping | US |
| 14 | Anthony | Shipping | US |
| 15 | Gerald | Sales | UK |
| 16 | Nanette | Sales | UK |
| 17 | John | Finance | US |
| 18 | Kelly | Shipping | US |

Task 15:

Write a query to find employees who work in departments with more than 5 employees.

Ans:

Inner join is used to find employees who work in departments with more than 5 employees.

Following is the command:

```
--22P-9216
select e.employee_id, e.first_name, e.department_id from employees e
join(
    select department_id from employees
    group by department_id
    having count(employee_id)>5
)dept_count on e.department_id=dept_count.department_id;
```

Following is the result:

| | EMPLOYEE_ID | FIRST_NAME | DEPARTMENT_ID |
|----|-------------|------------|---------------|
| 1 | 123 | Shanta | 50 |
| 2 | 134 | Michael | 50 |
| 3 | 135 | Ki | 50 |
| 4 | 147 | Alberto | 80 |
| 5 | 164 | Mattea | 80 |
| 6 | 193 | Britney | 50 |
| 7 | 110 | John | 100 |
| 8 | 113 | Luis | 100 |
| 9 | 114 | Den | 30 |
| 10 | 115 | Alexander | 30 |
| 11 | 118 | Guy | 30 |
| 12 | 120 | Matthew | 50 |
| 13 | 126 | Irene | 50 |
| 14 | 139 | John | 50 |
| 15 | 142 | Curtis | 50 |
| 16 | 176 | Jonathon | 80 |
| 17 | 181 | Jean | 50 |
| 18 | 182 | Martha | 50 |

Task 16:

Display a list of employees along with their managers' names.

Ans:

We use self join to display a list of employees and their manager names.

Following is the query:

```
--22P-9216
select e.employee_id, e.first_name as employee_name, m.employee_id as manager_id, m.first_name as manager_name
from employees e
left join employees m on e.manager_id=m.employee_id;
```

Following is the result:

Task 18:

Write a query to find employees who earn more than their department's average salary but less than the highest salary in the company.

Ans:

To find employees who earn more than their department's average salary but less than the highest salary of the company, we use inner join.

Following is the query:

```
--22P-9216
select e.first_name, e.salary, d.department_name from employees e
join departments d on e.department_id=d.department_id
where e.salary>(select avg(e1.salary) from employees e1 where e1.department_id=e.department_id)
and e.salary<(select max(salary) from employees)
```

Following is the result:

| | FIRST_NAME | SALARY | DEPARTMENT_NAME |
|----|------------|--------|-----------------|
| 1 | Daniel | 9000 | Finance |
| 2 | Nancy | 12008 | Finance |
| 3 | Den | 11000 | Purchasing |
| 4 | Michael | 13000 | Marketing |
| 5 | Shelley | 12008 | Accounting |
| 6 | Britney | 3900 | Shipping |
| 7 | Sarah | 4000 | Shipping |
| 8 | Jennifer | 3600 | Shipping |
| 9 | Kelly | 3800 | Shipping |
| 10 | Alexis | 4100 | Shipping |
| 11 | Nandita | 4200 | Shipping |
| 12 | Trenna | 3500 | Shipping |
| 13 | Renske | 3600 | Shipping |
| 14 | Kevin | 5800 | Shipping |
| 15 | Shanta | 6500 | Shipping |
| 16 | Payam | 7900 | Shipping |
| 17 | Adam | 8200 | Shipping |

Task 19:

Display a list of all employees who have worked in multiple departments, showing their job history and department names.

Ans:

We use inner join to display a list of all employees who have worked in multiple departments, showing their job history and department names.

Following is the query:

```
--22P-9216
select e.first_name, e.last_name, j.start_date, j.end_date, d.department_name
from job_history j
join employees e on j.employee_id=e.employee_id
join departments d on j.department_id=d.department_id
where j.employee_id in(
    select employee_id from job_history
    group by employee_id having count(distinct department_id)>1
)
order by e.employee_id, j.start_date
```

Following is the result:

| FIRST_NAME | LAST_NAME | START_DATE | END_DATE | DEPARTMENT_NAME |
|------------|-----------|------------|----------|-----------------|
|------------|-----------|------------|----------|-----------------|

Task 20:

Write a query to find employees who have worked in more than one region throughout their career.

Ans:

We use inner join to find employees who have worked in more than one region throughout their career.

Following is the query:

```
--22P-9216
select employee_id, count(distinct r.region_id) as region_count from job_history j
join departments d on j.department_id=d.department_id
join locations l on d.location_id=l.location_id
join countries c on l.country_id=c.country_id
join regions r on c.region_id=r.region_id
group by employee_id having count(distinct r.region_id)>1
```

Following is the result:

| EMPLOYEE_ID | REGION_COUNT |
|-------------|--------------|
|-------------|--------------|

Task 21:

List all employees and the region they are working in.

Ans:

To list all employees and the region they are working in, we use inner join.
Following is the query:

```
--22P-9216
select e.employee_id, e.first_name, e.last_name, r.region_name
from employees e
join departments d on e.department_id=d.department_id
join locations l on d.location_id=l.location_id
join countries c on l.country_id=c.country_id
join regions r on c.region_id=r.region_id
```

Following is the result:

| | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | REGION_NAME |
|----|-------------|-------------|-----------|-------------|
| 1 | 100 | Steven | King | Americas |
| 2 | 101 | Neena | Kochhar | Americas |
| 3 | 102 | Lex | De Haan | Americas |
| 4 | 103 | Alexander | Hunold | Americas |
| 5 | 104 | Bruce | Ernst | Americas |
| 6 | 105 | David | Austin | Americas |
| 7 | 106 | Valli | Pataballa | Americas |
| 8 | 107 | Diana | Lorentz | Americas |
| 9 | 108 | Nancy | Greenberg | Americas |
| 10 | 109 | Daniel | Faviet | Americas |
| 11 | 110 | John | Chen | Americas |
| 12 | 111 | Ismael | Sciarra | Americas |
| 13 | 112 | Jose Manuel | Urman | Americas |
| 14 | 113 | Luis | Popp | Americas |
| 15 | 114 | Den | Raphaely | Americas |
| 16 | 115 | Alexander | Khoo | Americas |
| 17 | 116 | Shelli | Baida | Americas |
| 18 | 117 | Sigal | Tobias | Americas |

Task 22:

Find employees who have the same last name but work in different departments.

Ans:

To find employees who have the same last name but work in different departments, we use inner join.
Following is the query:

```
--22P-9216
select e1.employee_id, e1.first_name, e1.last_name, e1.department_id, e2.employee_id as other_employee_id,
e2.first_name as other_first_name, e2.department_id as other_department_id
from employees e1
join employees e2
on e1.last_name=e2.last_name and e1.employee_id<>e2.employee_id and e1.department_id<>e2.department_id
```

Following is the result:

| | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | DEPARTMENT_ID | OTHER_EMPLOYEE_ID | OTHER_FIRST_NAME | OTHER_DEPARTMENT_ID |
|---|-------------|------------|-----------|---------------|-------------------|------------------|---------------------|
| 1 | 156 | Janette | King | 80 | 100 | Steven | 90 |
| 2 | 100 | Steven | King | 90 | 156 | Janette | 80 |
| 3 | 180 | Winston | Taylor | 50 | 176 | Jonathon | 80 |
| 4 | 176 | Jonathon | Taylor | 80 | 180 | Winston | 50 |

Task 23:

List employees who have changed job titles more than twice.

Ans:

Inner join is used to list employees who have change job ids more than twice.

Following is the query:

```
--22P-9216
select j.employee_id, e.first_name, e.last_name, count(j.job_id) as job_changes
from job_history j
join employees e on j.employee_id=e.employee_id
group by j.employee_id, e.first_name, e.last_name having count(j.job_id)>2
```

Following is the result:

| EMPLOYEE... | FIRST_NA... | LAST_NAME | JOB_CHA... |
|-------------|-------------|-----------|------------|
|-------------|-------------|-----------|------------|

Task 24:

Show job titles that are not currently assigned to any employee.

Ans:

To show job titles that are not currently assigned to any employee, we use left join.

Following is the query:

```
--22P-9216
select j.job_id, j.job_title from jobs j
left join employees e on j.job_id=e.job_id where e.job_id is null
```

Following is the result:

| JOB_ID | JOB_TITLE |
|--------------|--------------------------------|
| 1 AD_PRES | President |
| 2 AD_VP | Administration Vice President |
| 3 AD_ASST | Administration Assistant |
| 4 FI_MGR | Finance Manager |
| 5 FI_ACCOUNT | Accountant |
| 6 AC_MGR | Accounting Manager |
| 7 AC_ACCOUNT | Public Accountant |
| 8 SA_MAN | Sales Manager |
| 9 SA_REP | Sales Representative |
| 10 PU_MAN | Purchasing Manager |
| 11 PU_CLERK | Purchasing Clerk |
| 12 ST_MAN | Stock Manager |
| 13 ST_CLERK | Stock Clerk |
| 14 SH_CLERK | Shipping Clerk |
| 15 IT_PROG | Programmer |
| 16 MK_MAN | Marketing Manager |
| 17 MK_REP | Marketing Representative |
| 18 HR_REP | Human Resources Representative |

Task 25:

Find the top 3 employees with the highest salaries in each department.

Ans:

We have used subqueries to find the top 3 employees with the highest salaries in each department.

Following is the query:

```
--22P-9216
select el.employee_id, el.first_name, el.last_name, el.department_id, el.salary
from employees el where 3>(
    select count(distinct e2.salary) from employees e2
    where e2.department_id=el.department_id and e2.salary>el.salary
) order by el.department_id, el.salary desc
```

Following is the result:

| | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | DEPARTMENT_ID | SALARY |
|----|-------------|------------|-----------|---------------|--------|
| 1 | 200 | Jennifer | Whalen | 10 | 4400 |
| 2 | 201 | Michael | Hartstein | 20 | 13000 |
| 3 | 202 | Pat | Fay | 20 | 6300 |
| 4 | 114 | Den | Raphaely | 30 | 11000 |
| 5 | 115 | Alexander | Khoo | 30 | 3100 |
| 6 | 116 | Shelli | Baida | 30 | 2900 |
| 7 | 203 | Susan | Mavris | 40 | 6500 |
| 8 | 121 | Adam | Fripp | 50 | 8200 |
| 9 | 120 | Matthew | Weiss | 50 | 8000 |
| 10 | 122 | Payam | Kaufling | 50 | 7900 |
| 11 | 103 | Alexander | Hunold | 60 | 9000 |
| 12 | 104 | Bruce | Ernst | 60 | 6000 |
| 13 | 105 | David | Austin | 60 | 4800 |
| 14 | 106 | Valli | Pataballa | 60 | 4800 |
| 15 | 204 | Hermann | Baer | 70 | 10000 |
| 16 | 145 | John | Russell | 80 | 14000 |
| 17 | 146 | Karen | Partners | 80 | 13500 |
| 18 | 147 | Alberto | Errazuriz | 80 | 12000 |

