****

**UPA**

**Universidad Politécnica de Aguascalientes.**

**ISC06B**

**DATA BASE ADMINISTRATION**

# DP 11 PRACTICE

**TEACHER: Juan Carlos Herrera Hernández**

**STUDENTS:**

**Uriel Isaac Vazquez Martínez UP210934**

**Juan Pablo López González UP200053**

**José Miguel Escalera Rubalcava UP200667**

**Derek Gilberto Ramírez López UP200424**

**Aguascalientes, Ags, October 25th, 2022**

**Database Programming with SQL**

**11: Ensuring quality query results**

**Activities**

**Objectives**

* Create a query to produce specified data
* Modify a query to produce specified data

**Write the query**

1. **Page 5: Create a list of all tables whose first two characters in the name of the table is JO. The tables must be owned by the current Oracle User**

|  |  |
| --- | --- |
| Problem No: 1 page 5 | No. Rows in Result: |
| 1. **Page 5: Create a list of all tables whose first two characters in the name of the table is JO. The tables must be owned by the current Oracle User** | 3 |
| Text Code (No image) : | |
| --DP11  ----### Ej 5 Problem: ?Create a list of all tables whose first two characters in the name of the table is JO.  --The tables must be owned by the current Oracle User where table\_name like 'JO%';  select tname from tab  where tname LIKE 'JO%'; | |
| Image Result: | |
|  | |
|  | |

1. **Page 6: Create a list that includes the first initial of every employee's first name, a space, and the last name of the employee**

|  |  |
| --- | --- |
| Problem No: 2 page 6 | No. Rows in Result: |
| **2 Page 6: Create a list that includes the first initial of every employee's first name, a space, and the last name of the employee** | 20 |
| Text Code (No image) : | |
| --DP11  --6 Create a list that includes the first initial of every employee's first name, a space, and the last name of the employee  select (substr(first\_name,1,1)||' '|| last\_name) as "Employee Name" from employees ; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 7: Create a list of every employee's first name concatenated to a space and the employee's last name, and the email of all employees where the email address contains the string 'IN'**

|  |  |
| --- | --- |
| Problem No: 3 page 7 | No. Rows in Result: |
| **3 Page 7: Create a list of every employee's first name concatenated to a space and the employee's last name, and the email of all employees where the email address contains the string 'IN'** | 2 |
| Text Code (No image) : | |
| --DP11  -- --7 Create a list of every employee's first name concatenated to a space and the employee's last name,  --and the email of all employees where the email address contains the string 'IN'  select (first\_name||' '|| last\_name) as "Employee Name" , email from employees  where UPPER(email) like '%IN%'; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 8: Create a list of 'smallest' last name and the 'highest' last name from the employees table**

|  |  |
| --- | --- |
| Problem No: 4 page 8 | No. Rows in Result: |
| 1. **Page 8: Create a list of 'smallest' last name and the 'highest' last name from the employees table** | 1 |
| Text Code (No image) : | |
| --DP11  --8 Create a list of 'smallest' last name and the 'highest' last name from the employees table  select min(upper(last\_name)) as "First Last Name",max(upper(last\_name)) as "Last Last Name" from employees; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 9: Create a list of weekly salaries from the employees table where the weekly salary is between 700 and 3000. The salaries should be formatted to include a $-sign and have two decimal points like: $9999.99**

|  |  |
| --- | --- |
| Problem No: 5 page 9 | No. Rows in Result: |
| 1. **Page 9: Create a list of weekly salaries from the employees table where the weekly salary is between 700 and 3000. The salaries should be formatted to include a $-sign and have two decimal points like: $9999.99** | 15 |
| Text Code (No image) : | |
| --DP11  -- 9-- Create a list of weekly salaries from the employees table where the weekly salary is  --between 700 and 3000  --? The salaries should be formatted to include a $-sign and have two  --decimal points like: $9999.99  select to\_char((salary\*12)/52, '$9999.99') as weekly\_salary  from employees  where (salary\*12)/52 between 700 and 3000; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 10: Create a list of every employee and his related job title sorted by job\_title**

|  |  |
| --- | --- |
| Problem No: 6 page 10 | No. Rows in Result: |
| 1. **Page 10: Create a list of every employee and his related job title sorted by job\_title** | 20 |
| Text Code (No image) : | |
| --DP11  -- 10--Create a list of every employee and his related job title sorted by job\_title  select (SUBSTR(e.first\_name,1,1)|| ' ' ||last\_name) as "Employee Name", j.job\_title as "Job"  from employees e  inner join jobs j on e.job\_id = j.job\_id  order by (j.job\_title); | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 11: -Create a list of every employee’s job, the salary ranges within the job, and the employee's salary List the lowest and highest salary range within each job with a dash to separate the salaries like this: 100 – 200**

|  |  |
| --- | --- |
| Problem No: 7 page 11 | No. Rows in Result: |
| 1. **Page 11: -Create a list of every employee’s job, the salary ranges within the job, and the employee's salary List the lowest and highest salary range within each job with a dash to separate the salaries like this: 100 – 200** | 20 |
| Text Code (No image) : | |
| --DP11  -- 11-- Create a list of every employee’s job, the salary ranges within the job, and the employee's salary  --List the lowest and highest salary range within each job with a dash to separate the salaries like this: 100 – 200  select (SUBSTR(e.first\_name,1,1)|| ' ' ||last\_name) as "Employee Name", j.job\_title Job,(j.min\_salary || ' - '|| j.max\_salary) as "Salary Range",  e.salary as "Employee's Salary"  from employees e  inner join jobs j on e.job\_id = j.job\_id  order by (j.job\_title); | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 12: Using an ANSII join method, create a list of every employee's first initial and last name, and department name. Make sure the tables are joined on all of the foreign keys declared between the two tables**

|  |  |
| --- | --- |
| Problem No: 8 page 12 | No. Rows in Result: |
| 1. **Page 12: Using an ANSII join method, create a list of every employee's first initial and last name, and department name. Make sure the tables are joined on all of the foreign keys declared between the two tables** | 12 |
| Text Code (No image) : | |
| --DP11  -- 12--Using an ANSII join method, create a list of every employee's first initial and last name, and department name  --? Make sure the tables are joined on all of the foreign keys declared between the two tables  select (SUBSTR(e.first\_name,1,1)||' '|| e.last\_name)as "Employee Name", d.department\_name  from employees e  natural jOIN departments d  natural join jobs j; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 13: Change the previous listing to join only on the department\_id column**

|  |  |
| --- | --- |
| Problem No: 9 page 13 | No. Rows in Result: |
| **9 Page 13: Change the previous listing to join only on the department\_id column** | 19 |
| Text Code (No image) : | |
| --DP11  -- 13--Change the previous listing to join only on the department\_id column  select (SUBSTR(e.first\_name,1,1)||' '|| e.last\_name) as Employee\_Name, d.department\_name  from employees e  INNER join departments d on d.department\_id = e.department\_id; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 14: Create a list of every employee's last name, and the word nobody or somebody depending on whether or not the employee has a manager Use the Oracle DECODE function to create the list**

|  |  |
| --- | --- |
| Problem No: 10 page 14 | No. Rows in Result: |
| **10 Page 14: Create a list of every employee's last name, and the word nobody or somebody depending on whether or not the employee has a manager Use the Oracle DECODE function to create the list** | 20 |
| Text Code (No image) : | |
| --DP11  -- 14-- Create a list of every employee's last name, and the word nobody or somebody depending on  --whether or not the employee has a manager  --? Use the Oracle DECODE function to create the list  select DECODE(manager\_id, '','Nobody','Somebody') as "Works For",last\_name  from employees; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 15: Create a list of every employee's first initial and last name, salary, and a yes or no to show whether or not an employee makes a commission. Fix this query to produce the result**

|  |  |
| --- | --- |
| Problem No: 11 page 15 | No. Rows in Result: |
| 1. **Page 15: Create a list of every employee's first initial and last name, salary, and a yes or no to show whether or not an employee makes a commission. Fix this query to produce the result** | 20 |
| Text Code (No image) : | |
| --DP11  -- 15-- Create a list of every employee's first initial and last name, salary, and a yes or no to show whether or not an  --employee makes a commission  --? Fix this query to produce the result  SELECT (SUBSTR(first\_name,1,1)||' '||last\_name) as "Employee Name", salary as "Salary",DECODE(COMMISSION\_PCT, NULL, 'No', 'Yes' ) AS "Commission"  from employees; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 16: Create a list of every employee's last name, department name, city, and state\_province Include departments without employees. An outer join is required**

|  |  |
| --- | --- |
| Problem No: 12 page 16 | No. Rows in Result: |
| 1. **Page 16: Create a list of every employee's last name, department name, city, and state\_province Include departments without employees. An outer join is required** | 20 |
| Text Code (No image) : | |
| --DP11  -- 16-- Create a list of every employee's last name, department name, city, and state\_province  --include departments without employees  select e.last\_name, d.department\_name, l.city,l.state\_province  from employees e  right outer join departments d on e.department\_id = d.department\_id  inner join locations l on l.location\_id = d.location\_id; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 17: Create a list of every employee's first and last names, and the first occurrence of: commission\_pct, manager\_id, or -1 If an employee gets commission, display the commission\_pct column; if no commission, then display his manager\_id; if he has neither commission nor manager, then the number -1**

|  |  |
| --- | --- |
| Problem No: 13 page 17 | No. Rows in Result: |
| **13 Page 17: Create a list of every employee's first and last names, and the first occurrence of: commission\_pct, manager\_id, or -1 If an employee gets commission, display the commission\_pct column; if no commission, then display his manager\_id; if he has neither commission nor manager, then the number -1** | 20 |
| Text Code (No image) : | |
| --DP11  -- 17 Create a list of every employee's first and last names, and the first occurrence  --of: commission\_pct, manager\_id, or -1 If an employee gets commission, display the commission\_pct column;  --if no commission, then display his manager\_id; if he has neither commission nor manager, then the number -1  select first\_name "First Name", last\_name "Last Name" ,COALESCE (COMMISSION\_PCT, manager\_id, -1) "Wich Function"  from employees; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 18: Create a list of every employee's last name, salary, and job\_grade for all employees working in departments with a department\_id greater than 50**

|  |  |
| --- | --- |
| Problem No: 14 page 18 | No. Rows in Result: |
| **14 Page 18: Create a list of every employee's last name, salary, and job\_grade for all employees working in departments with a department\_id greater than 50** | 11 |
| Text Code (No image) : | |
| --DP11  -- 18-- Create a list of every employee's last name, salary, and job\_grade for all employees working  -- in departments with a department\_id greater than 50  select e.last\_name,e.salary,j.grade\_level from employees e, job\_grades j  where department\_id >50 and (salary BETWEEN j.lowest\_sal AND j.highest\_sal)  order by salary; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 19: Produce a list of every employee's last name and department name Include both employees without departments, and department**

|  |  |
| --- | --- |
| Problem No: 15 page 19 | No. Rows in Result: |
| **15 Page 19: Produce a list of every employee's last name and department name Include both employees without departments, and department** | 21 |
| Text Code (No image) : | |
| --DP11  -- 19-- Produce a list of every employee's last name and department name  -- Include both employees without departments, and department  select NVL(e.last\_name,'-') Last\_Name, NVL(d.department\_name,'-') Department\_Name  from employees e  full outer join Departments d on e.department\_id = d.department\_id  order by d.Department\_Name,Last\_Name; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 20: Create a treewalking list of every employee's last name, his managers last name, and his position in the company. The top level manager has position 1, this manager's subordinates position 2, their subordinates position 3, and so on. Start the listing with employee number 100**

|  |  |
| --- | --- |
| Problem No: 16 page 20 | No. Rows in Result: |
| **16 Page 20: Create a treewalking list of every employee's last name, his managers last name, and his position in the company. The top level manager has position 1, this manager's subordinates position 2, their subordinates position 3, and so on. Start the listing with employee number 100** | 20 |
| Text Code (No image) : | |
| --DP11  -- 20-- Create a treewalking list of every employee's last name, his manager?s last name, and his position in the company  --The top level manager has position 1, this manager's subordinates position 2, their subordinates position 3, and so on  --Start the listing with employee number 100  SELECT LEVEL AS "Position", e.last\_name, PRIOR e.last\_name AS "Manager name"  FROM employees e START WITH e.employee\_id = 100  CONNECT BY PRIOR e.employee\_id = e.manager\_id; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 21: Produce a list of the earliest hire date, the latest hire date, and the number of employees from the employees table**

|  |  |
| --- | --- |
| Problem No: 17 page 21 | No. Rows in Result: |
| **17 Page 21: Produce a list of the earliest hire date, the latest hire date, and the number of employees from the employees table** | 1 |
| Text Code (No image) : | |
| --DP11  -- 21-- Produce a list of the earliest hire date, the latest hire date, and the number of employees from the employees table  select min(hire\_date) Lowest ,max(hire\_date) Highest ,count(last\_name) "No of Employees" from employees; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 22: Create a list of department names and the departmental costs (salaries added up) Include only departments whose salary costs are between 15000 and 31000, and sort the listing by the cost**

|  |  |
| --- | --- |
| Problem No: 18 page 22 | No. Rows in Result: |
| 1. **Page 22: Create a list of department names and the departmental costs (salaries added up) Include only departments whose salary costs are between 15000 and 31000, and sort the listing by the cost** | 5 |
| Text Code (No image) : | |
| --DP11  -- 22-- Create a list of department names and the departmental costs (salaries added up)  --Include only departments whose salary costs are between 15000 and 31000, and sort the listing by the cost  select d.department\_name,sum(e.salary) salaries from departments d  inner join employees e on e.department\_id = d.department\_id  having sum(e.salary) BETWEEN 15000 and 31000  group by d.department\_name  order by salaries; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 23: Create a list of department names, the manager id, manager name (employee last name) of that department, and the average salary in each department**

|  |  |
| --- | --- |
| Problem No: 19 page 23 | No. Rows in Result: |
| **19 Page 23: Create a list of department names, the manager id, manager name (employee last name) of that department, and the average salary in each department** | 7 |
| Text Code (No image) : | |
| --DP11  -- 23-- Create a list of department names, the manager id, manager name (employee last name) of that department, and the  --average salary in each department  select d.department\_name, d.manager\_id,e.last\_name Manager\_Name, T1.AVG\_Dept\_Salary from departments d  inner join (select department\_id,floor(avg(salary)) as AVG\_Dept\_Salary from employees  group by department\_id)T1  on d.department\_id = T1.department\_id  inner join employees e on d.manager\_id = e.employee\_id  order by (T1.AVG\_Dept\_Salary); | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 24: Show the highest average salary for the departments in the employees table Round the result to the nearest whole number**

|  |  |
| --- | --- |
| Problem No: 20 page 24 | No. Rows in Result: |
| 1. **Page 24: Show the highest average salary for the departments in the employees table Round the result to the nearest whole number** | 1 |
| Text Code (No image) : | |
| --DP11  -- 24--Show the highest average salary for the departments in the employees table  --? Round the result to the nearest whole number  select round(max(avg(salary) )) AS "Highest Avg Sal for Depts"  from employees  group by department\_id; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 25: Create a list of department names and their monthly costs (salaries added up)**

|  |  |
| --- | --- |
| Problem No: 21 page 25 | No. Rows in Result: |
| **21 Page 25: Create a list of department names and their monthly costs (salaries added up)** | 7 |
| Text Code (No image) : | |
| --DP11  -- 25--Create a list of department names and their monthly costs  --(salaries added up)  select d.department\_name,sum (e.salary) as "Monthly Cost"  from departments d, employees e  where d.department\_id = e.department\_id  group by d.department\_name  order by d.department\_name ; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 26:** **Create a list of department names, and job\_ids Calculate the monthly salary cost for each job\_id within department, for each department, and for all departments added together**

|  |  |
| --- | --- |
| Problem No: 22 page 26 | No. Rows in Result: |
| **22 Page 26:** **Create a list of department names, and job\_ids Calculate the monthly salary cost for each job\_id within department, for each department, and for all departments added together** | 20 |
| Text Code (No image) : | |
| --DP11  -- 26--Create a list of department names, and job\_ids  --Calculate the monthly salary cost for each job\_id within a department, for each department, and for all  --departments added together  select NVL(d.department\_name,'-') as "Department Name", NVL(e.job\_id, '-' ) as "JOB TITLE", sum(e.salary) AS "MONTHLY COST"  from employees e inner join departments d on e.department\_id = d.department\_id  where e.department\_id is not null  group by ROLLUP (d.department\_name,e.job\_id)  order by d.department\_name; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 27: Create a list of department names, and job\_ids. Calculate the monthly salary cost for each job\_id within a department, for each department, for each group of job\_ids irrespective of the department, and for all departments added together (Hint: Cube)**

|  |  |
| --- | --- |
| Problem No: 23 page 27 | No. Rows in Result: |
| **23 Page 27: Create a list of department names, and job\_ids. Calculate the monthly salary cost for each job\_id within a department, for each department, for each group of job\_ids irrespective of the department, and for all departments added together (Hint: Cube)** | 32 |
| Text Code (No image) : | |
| --DP11  -- 27--Create a list of department names, and job\_ids  --Calculate the monthly salary cost for each job\_id within a department, for each department, for each group of  --job\_ids irrespective of the department, and for all departments added together (Hint: Cube)  select NVL(d.department\_name,'-') as "Department Name", NVL(e.job\_id, '-' ) as "JOB TITLE", sum(e.salary) AS "MONTHLY COST"  from employees e inner join departments d using (department\_id)  where department\_id is not null  group by cube (d.department\_name,e.job\_id)  order by d.department\_name; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 28: Expand the previous list to also show if the department\_id or job\_id was used to create the subtotals shown in the output (Hint: Cube,Grouping)**

|  |  |
| --- | --- |
| Problem No: 24 page 28 | No. Rows in Result: |
| **24 Page 28: Expand the previous list to also show if the department\_id or job\_id was used to create the subtotals shown in the output (Hint: Cube,Grouping)** | 32 |
| Text Code (No image) : | |
| --DP11  -- 28--Expand the previous list to also show if the department\_id or job\_id was used to create  --the subtotals shown in the output (Hint: Cube,Grouping)  select NVL(d.department\_name,'-') AS "Department Name", NVL(e.job\_id,'-') as "Job Title", sum(e.salary) "Monthly Cost",  DECODE (GROUPING(department\_name),'1','NO','YES') AS "Dept id used",  DECODE(GROUPING(job\_id),1,'NO','YES') AS "Job id used"  from employees e inner join departments d using (department\_id)  where department\_id is not null  group by cube (d.department\_name,e.job\_id)  order by d.department\_name; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 29: Create a list that includes the monthly salary costs for each job title within a department. In the same list, display the monthly salary cost per city.**

|  |  |
| --- | --- |
| Problem No: 25 page 29 | No. Rows in Result: |
| 1. **Page 29: Create a list that includes the monthly salary costs for each job title within a department. In the same list, display the monthly salary cost per city.** | 17 |
| Text Code (No image) : | |
| --DP11  -- 29-- Create a list that includes the monthly salary costs for each job title within a department  -- In the same list, display the monthly salary cost per city.  SELECT NVL(d.department\_name,'-'), nvl(e.job\_id,'-'),nvl(l.city,'-'), SUM(e.salary) DSum  FROM employees e  inner join departments d on e.department\_id = d.department\_id  inner join locations l on l.location\_id = d.location\_id  where e.department\_id is not null  GROUP BY Grouping Sets((d.department\_name,e.job\_id),(l.city))  order by d.department\_name,e.job\_id desc,l.city; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 30: reate a list of employee names as shown and department ids In the same report, list the department ids and department names. And finally, list the citiesThe rows should not be joined, just listed in the same report. (Hint: Union)**

|  |  |
| --- | --- |
| Problem No: 26 page 30 | No. Rows in Result: |
| 1. **Page 30: reate a list of employee names as shown and department ids In the same report, list the department ids and department names. And finally, list the citiesThe rows should not be joined, just listed in the same report. (Hint: Union)** | 33 |
| Text Code (No image) : | |
| --DP11  -- 30-- Create a list of employee names as shown and department ids  --?In the same report, list the department ids and department names. And finally, list the cities  -- The rows should not be joined, just listed in the same report. (Hint: Union)  select (substr(e.first\_name,1,1)||' '|| e.last\_name) as "Employee",to\_char(department\_id) as "Department ID" , '-' as "Department Name", '-' as city  from employees e  union  select '-' as "Employee",to\_char(department\_id) as "Department ID" , department\_name as "Department Name", '-' as city  from departments d  union  select '-' as "Employee",'-' as "Department ID" , '-' as "Department Name", city  from Locations; | |
| Image Result: | |
|  | |
|  | |
|  | |

1. **Page 31: Create a list of each employee's first initial and last name, salary, and department name for each employee earning more than the average for his department**

|  |  |
| --- | --- |
| Problem No: 27 page 31 | No. Rows in Result: |
| **27 Page 31: Create a list of each employee's first initial and last name, salary, and department name for each employee earning more than the average for his department** | 7 |
| Text Code (No image) : | |
| --DP11  -- 31-- Create a list of each employee's first initial and last name, salary,  --and department name for each employee earning more than the  --average for his department  select (substr(first\_name,1,1)||' '|| last\_name) as "Employee" ,e.salary,d.department\_name as "Department Name"  from employees e  inner join (select department\_id,avg(salary) average from employees  group by department\_id) av  on e.department\_id = av.department\_id  inner join departments d on e.department\_id = d.department\_id  where e.salary > av.average; | |
| Image Result: | |
|  | |
|  | |
|  | |