

Guía de Laboratorio 4

1. Diccionario de datos

Oracle 12c tiene más de 900 vistas con prefijo DBA, y más de 800 vistas ALL y USER

- a. Elabore una consulta que muestre la fecha de creación de la tabla HR.EMPLOYEES

```
SELECT object_name,created
FROM dba_objects
WHERE object_name='EMPLOYEES' and owner='HR';
```

- b. Elabore una consulta que muestre los índices de la tabla SCOTT.EMP

```
SELECT table_owner, table_name, index_name, column_name
FROM dba_ind_columns
WHERE table_owner='SCOTT'
AND table_name='EMP'
ORDER BY table_owner, table_name, column_name;
```

- c. Elabore una consulta que liste los propietarios de objetos y número de objetos por propietario

```
SELECT owner, count(owner) Numero
FROM dba_objects
GROUP BY owner
ORDER BY Numero desc;
```

- d. Elabore una consulta que muestre un listado de los espacios de tabla, su estado, tamaño en MB y cuál es el nombre del datafile asociado.

```
SELECT t.tablespace_name "Tablespace", t.status
"Estado", round(d.bytes/1024/1024,2)
"Tamaño (MB)", substr(d.file_name,1,80)
"Fichero de datos"
FROM dba_data_files d,
dba_tablespaces t
WHERE t.tablespace_name = d.tablespace_name;
```

2. Vistas de performance

Oracle 12c tiene más de 2500 vistas dinámicas de performance.

- a. Listar las vistas de performance ejecutando la siguiente consulta:

```
SQL> SELECT NAME,TYPE
2 FROM V$FIXED_TABLE
3 WHERE NAME LIKE 'V$%';
```

NAME	TYPE
V\$WAITSTAT	VIEW
V\$BH	VIEW
V\$GC_ELEMENT	VIEW
V\$CR_BLOCK_SERVER	VIEW
V\$CURRENT_BLOCK_SERVER	VIEW
V\$POLICY_HISTORY	VIEW
V\$ENCRYPTED_TABLESPACES	VIEW
V\$GC_ELEMENTS_WITH_COLLISIONS	VIEW
V\$FILE_CACHE_TRANSFER	VIEW
V\$TEMP_CACHE_TRANSFER	VIEW
V\$CLASS_CACHE_TRANSFER	VIEW

- b. Elabore una consulta que muestre el SID, Serial, el nombre de máquina, el usuario conectado a la base de datos Oracle y el nombre de usuario de sistema operativo de las sesiones activas en el servidor Oracle.

```
SELECT sid,
       serial#,
       machine,
       username,
       osuser,
       program
FROM v$session;
```

- c. Elabore una consulta que muestre que está ejecutando un usuario

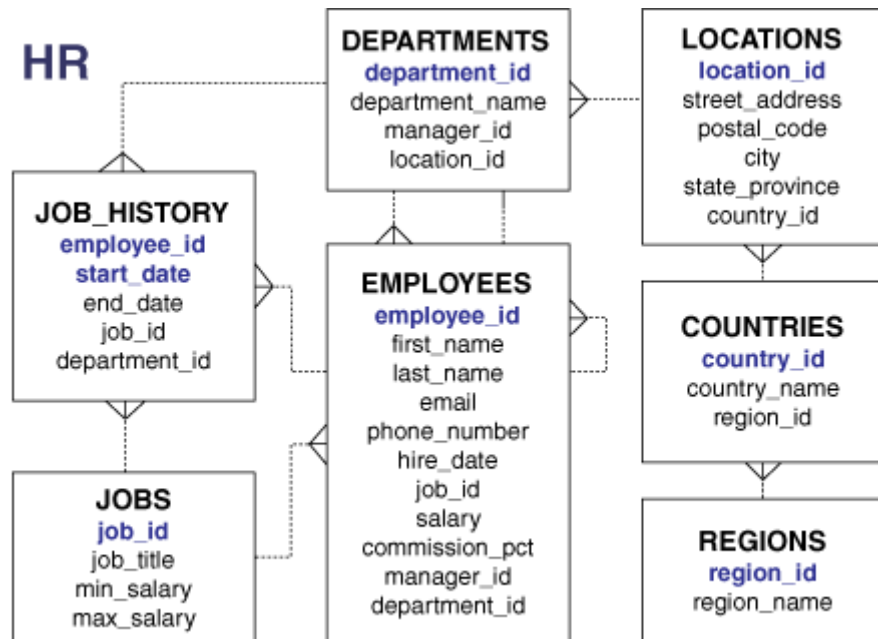
```
SELECT a.sid, a.serial#, b.sql_text
FROM v$session a, v$sqlarea b
WHERE a.sql_address=b.address
AND a.username='HR';
```

- d. Elabore una consulta que muestra los usuarios de Oracle conectados y el número de sesiones por usuario

```
SELECT username Usuario_Oracle, count(username) Numero_Sesiones
FROM v$session
GROUP BY username
```

```
ORDER BY Numero_Sesiones desc;
```

Las siguientes consultas utilizan el esquema de ejemplo HR de la pdb XEPDB1.



Abrir una ventana de comandos de DOS y ejecutar el SQLPLUS conectándose a la pds XEPDB1 con el usuario SYS usando el comando ALTER SESSION (puede ejecutar SHOW PDBS para listar las pdbs creadas)

```
C:\Users\Administrator>sqlplus / as sysdba
```

```
SQL> alter session set container=XEPDB1;

Session altered.
```

3. Seleccionando Columnas

- El departamento de HR necesita una consulta que muestre el apellido, job ID, fecha de contratación y ID de empleado. El ID de empleado debe aparecer primero. Coloque un alias STARTDATE para la columna HIRE_DATE.

```
SELECT employee_id, last_name, job_id, hire_date StartDate
FROM hr.employees;
```

- b. El departamento de HR requiere columnas más descriptivas en su reporte. Nombre las columnas Emp #, Employee, Job, y Hire Date, respectivamente. Ejecute la consulta nuevamente.

```
SELECT employee_id "Emp #",  
       last_name "Employee",  
       job_id "Job",  
       hire_date "Hire Date"  
FROM hr.employees;
```

- c. El departamento de HR requiere un reporte de todos los empleados y su Job ID. Muestre el last name concatenado con el job ID separado por coma (,) y espacio () y nombrar la columna **Employee and Title**.

```
SELECT last_name||', '||job_id "Employee and Title"  
FROM hr.employees;
```

4. Aplicando filtros y ordenando datos

Debido a problemas de presupuesto, el departamento de HR le ha pedido su ayuda para obtener la siguiente información:

- a. Un reporte que liste el Last Name y Salary de todos los empleados que ganen más de \$12,000.

```
SELECT last_name, salary  
FROM hr.employees  
WHERE salary > 12000;
```

- b. Un reporte que liste el last name y salary de todos los empleados cuyo salario no se encuentre en el rango de \$5,000 a \$12,000.

```
SELECT last_name, salary  
FROM hr.employees  
WHERE salary NOT BETWEEN 5000 AND 12000;
```

- c. Listar el Last Name, Salary y Department ID de los empleados que ganen entre \$5,000 y \$12,000 y trabajen en los departamentos 20 o 50 ordenados ascendentemente por apellido.

```
SELECT last_name "Employee", salary "Monthly Salary", department_id  
FROM hr.employees  
WHERE salary BETWEEN 5000 AND 12000  
AND department_id IN (20, 50);
```

- d. Listar el last name y hire date de los empleados contratados en el año 2006

```
SELECT last_name, hire_date
FROM hr.employees
WHERE hire_date >= '01-JAN-06'
AND hire_date < '01-JAN-07';

SELECT last_name, hire_date
FROM hr.employees
WHERE to_char(hire_date,'YYYY') = '2006';
```

- e. Listar el Last Name de los empleados que no tienen un Manager

```
SELECT last_name, job_id
FROM hr.employees
WHERE manager_id IS NULL;
```

- f. Un reporte que liste el Last Name, Salary y Commission para todos los empleados que ganan una comisión. Ordenar los resultados descendientemente por Salary y Commission.

```
SELECT last_name, salary, commission_pct
FROM hr.employees
WHERE commission_pct IS NOT NULL
ORDER BY 2 DESC, 3 DESC;
```

5. Funciones

- a. El departamento de HR desea un reporte que muestre el Employee ID, Last Name, Salary, Salary incrementado en 15% y la diferencia de los salarios.

```
SELECT employee_id, last_name, salary, ROUND(salary
* 1.155, 0) "New Salary", ROUND(salary *
1.155, 0) - salary "Increase"
FROM hr.employees;
```

- b. Generar un reporte que muestre el Last Name (con la primera letra en mayúscula y las demás en minúsculas) y la longitud del Last Name de los empleados cuyo apellido inicie con las letras "J," "A," o "M". Ordenas los resultados por Last Name.

```
SELECT INITCAP(last_name) "Name",
LENGTH(last_name) "Length"
FROM hr.employees
WHERE last_name LIKE 'J%'
OR last_name LIKE 'A%'
OR last_name LIKE 'M%';
```

```
OR last_name LIKE 'A%'
ORDER BY last_name;
```

- c. Generar un reporte que muestre el Last Name y el tiempo que un empleado tiene contratado en meses como números enteros. Nombre la columna MONTHS_WORKED y ordene los resultados por el número de meses contratados.

```
SELECT last_name,
       ROUND(MONTHS_BETWEEN(SYSDATE, hire_date)) MONTHS_WORKED
FROM hr.employees
ORDER BY months_worked;
```

- d. Genere un reporte que muestre el Last Name y el Salary expresado en asteriscos (*). Cada asterisco representa \$1,000. Ordene los resultados por Salary en forma descendente y nombre la columna EMPLOYEES_AND_THEIR_SALARIES.

```
SELECT last_name,
       rpad(' ', salary/1000, '*') EMPLOYEES_AND_THEIR_SALARIES
FROM hr.employees
ORDER BY salary DESC;
```

6. Agregación

- a. Listar el Salary mayor, menor, suma y promedio de todos los empleados. Nombre las columnas Maximum, Minimum, Sum, y Average, respectivamente. Redondee los resultados al entero más cercano.

```
SELECT ROUND(MAX(salary),0) "Maximum",
       ROUND(MIN(salary),0) "Minimum",
       ROUND(SUM(salary),0) "Sum",
       ROUND(AVG(salary),0) "Average"
FROM hr.employees;
```

- b. Listar el Salary mayor, menor, suma y promedio de los empleados agrupados por Job Id. Nombre las columnas Maximum, Minimum, Sum, y Average, respectivamente. Redondee los resultados al entero más cercano.

```
SELECT job_id, ROUND(MAX(salary),0) "Maximum",
       ROUND(MIN(salary),0) "Minimum",
       ROUND(SUM(salary),0) "Sum",
       ROUND(AVG(salary),0) "Average"
FROM hr.employees
GROUP BY job_id;
```

- c. Listar la cantidad de empleados por Departamento. Ordene por cantidad de empleados de forma descendente.

```
SELECT department_id, COUNT(*)  
FROM hr.employees  
GROUP BY department_id;
```

- d. Cree un reporte que liste el Manager Id y el Salary del empleado menos pagado para ese Manager. Excluya a todos cuyo Manager es desconcido. Excluya todos los grupos cuyo salario mínimo sea \$6000 o menor. Ordene los resultados descendentemente por Salary.

```
SELECT manager_id, MIN(salary)
FROM hr.employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
HAVING MIN(salary) > 6000
ORDER BY MIN(salary) DESC;
```