

Manejo de usuarios en Oracle

Privilegios	Descripción
CREATE SESSION	Crear conexiones a la BD
CREATE TABLE	Crear tablas en tu esquema
CREATE VIEW	Crear vistas en tu esquema

Usuarios	Descripción
Creación y modificación	CREATE USER Nombre IDENTIFIED BY Contraseña ALTER USER Nombre IDENTIFIED BY Nueva_Contraseña

Revocar permisos REVOKE **Privilegio** ON **Objeto** TO **Nombre_Usuario**

Permitir acceso GRANT CREATE session TO **Nombre_Usuario** GRANT CONNECT to **Nombre_Usuario** GRANT UNLIMITED TABLESPACE TO **Nombre_Usuario** GRANT SELECT ON **Nombre_Tabla** TO **Nombre_Usuario** GRANT INSERT ON **Nombre_Tabla** TO **Nombre_Usuario**

Eliminar usuario DROP USER **Nombre** CASCADE

Al eliminar un usuario con el código DROP USER **Nombre** CASCADE este borra al usuario y a todo lo relacionado a este con el código "Cascade"

Ejemplo sobre Pivot

```
--Crear una vista
Create view PUESTO_DEPTNO as
Select * from
(select JOB, DEPTNO, SAL From emp)
Pivot(
Sum(Sal)
For Deptno
In(10 as "Deptno 10",20 As "Deptno 20",30 as "Deptno 30")
);
```

JOB	Deptno 10	Deptno 20	Deptno 30
1 PRESIDENT	5000	(null)	(null)
2 MANAGER	2450	2975	2850
3 SALESMAN	(null)	(null)	5600
4 CLERK	1300	1900	950
5 ANALYST	(null)	6000	(null)

Crear una vista y en usar un Pivot para dividir la información por Departamento "Deptno"

Ejemplo sobre Case

```
--Realizar una consulta, que muestre el código del departamento, el nombre del departamento, el nombre de empleado, salario y la cantidad de año que tiene que trabajar en la empresa.
--Agregar una columna para comparar la cantidad de años del empleado respecto al siguiente, en caso de que sea mayor ">>>" o si es menor "<<<".
```

Realizar una consulta, que muestre el código del departamento, el nombre del departamento, el nombre de empleado, salario y la cantidad de año que tiene que trabajar en la empresa.

Agregar una columna para comparar la cantidad de años del empleado respecto al siguiente, en caso de que sea mayor ">>>" o si es menor "<<<".

Ejemplo sobre Case

```
--Realizar una consulta, que muestre el código del departamento, el nombre del departamento, el nombre de empleado, salario y la cantidad de año que tiene que trabajar en la empresa.
--Agregar una columna para comparar la cantidad de años del empleado respecto al siguiente, en caso de que sea mayor ">>>" o si es menor "<<<".
```

Realizar una consulta, que muestre el código del departamento, el nombre del departamento, el nombre de empleado, salario y la cantidad de año que tiene que trabajar en la empresa.

Agregar una columna para comparar la cantidad de años del empleado respecto al siguiente, en caso de que sea mayor ">>>" o si es menor "<<<".

PL/SQL

Un bloque PL/SQL es una unidad de código que contiene una o más sentencias PL/SQL. La estructura general de un bloque PL/SQL es la siguiente:

Cursores

En SQL, un cursor es una estructura de control utilizada para recorrer y manipular filas de un resultado de consulta de forma secuencial. Los cursores proporcionan un mecanismo para procesar registros uno a uno y realizar operaciones específicas en cada uno de ellos.

Procedimientos y Funciones

Los procedimientos y las funciones son subprogramas que permiten agrupar y reutilizar bloques de código. Los procedimientos se utilizan para realizar acciones o tareas, mientras que las funciones devuelven un valor.

Ejemplo sobre First y Last Value

```
--Select deptno, empno, sal,
--First_Value(empno) Over(partition by Deptno order by Sal Desc) Primero,
--Last_Value(empno) Over(partition by Deptno order by Sal Desc) Ultimo
From emp;
```

DEPTNO	EMPNO	SAL	PRIMERO	ULTIMO
1	10 KING	5000 KING	KING	
2	20 CLARK	2450 CLARK	CLARK	
3	30 MILLER	1300 MILLER	MILLER	
4	40 FORD	3000 FORD	FORD	
5	50 JONES	2975 JONES	JONES	
6	60 SMITH	1200 SMITH	SMITH	
7	70 ADAMS	1100 ADAMS	ADAMS	
8	80 SCOTT	950 SCOTT	SCOTT	
9	90 BLAKE	2850 BLAKE	BLAKE	
10	10 ALLEN	1600 ALLEN	ALLEN	
11	11 TURNER	1500 TURNER	TURNER	
12	12 MARTIN	1250 MARTIN	MARTIN	
13	13 NWARD	1200 NWARD	NWARD	
14	14 JAMES	950 JAMES	JAMES	

NOTA: Tomar en cuenta que se está usando el Departamento(Deptno) para seleccionar el Primer y Último Nombre .

Ejemplo sobre Over (Partition by ...)

```
select Deptno, ename, extract (year from hiredate) año, sal,  
row number() over(partition by deptno order by to_char(hiredate, 'YYYY-MM')) as X from EMP;
```

	DE	ENAME	AGE	SAL	X
1		10 CLARK	1981 2450	1	
2		10 KING	1981 5000	2	
3		10 MILLER	1982 1300	3	
4		20 SMITH	1980 800	1	
5		50 JONES	1981 2975	2	
6		20 FORD	1981 3000	3	
7		20 SCOTT	1982 3000	4	
8		20 ADAMS	1985 1100	5	
9		30WARD	1981 1200	1	
10		30 ALLEN	1981 1600	2	
11		30 BLAKE	1981 2850	3	
12		30 MARTIN	1982 1200	4	
13		30 TURNER	1981 1500	5	
14		30 JAMES	1981 950	6	

Realizar una consulta que muestre el nombre del colaborador, el año en que ingreso a la empresa, el salario, y una clasificación por orden a los colaboradores por el año y mes en que ingresaron, segmento por departamento

Ejemplo de Group by (Rollup)

```
Select d.Dname, to_char(hiredate,'MM') Mes,count(*) as CantidadColaboradores, round(avg(sal)) as PromedioSalario
from Emp e inner join Dept d on e.deptno = d.deptno
Group by rollup(d.dname,to_char(hiredate,'MM'));
```

NAME	MES	CATEDACOLABORADORES	PROHEGOSALADO
1 ACCOUNTING 11		3	2450
2 ACCOUNTING 04		3	2450
3 ACCOUNTING 01		3	1500
4 RESEARCH 01		3	1100
5 RESEARCH 12		3	2267
6 RESEARCH 04		3	2976
7 SALES 02		2	1425
8 SALES 12		3	950
9 SALES 08		2	175
10 SALES 08		1	2556
11 ACCOUNTING (null)		3	2417
12 RESEARCH (null)		5	2175
13 SALES (null)		6	1647
14 (null)		24	2073

Consulta que muestre por Departamento y por mes, la cantidad de colaboradores y el promedio del salario.

Se verá por Deptno, mes, por deptno(nombre) y un total general.

Ejemplo de Group by (cube)

```
select to_char(hiredate,'YYYY')Anno,Job,sum(sal),max(sal)
from emp
group by cube( Job, to_char(hiredate,'YYYY'));
```

1	NAME	JOB	SAL	DEPT
2	SMITH	CLERK	2900	800
3	ALLEN	SALES	3100	900
4	WARD	SALES	2500	900
5	MARTIN	SALES	4000	900
6	MANEY	CLERK	1100	1000
7	DAVIS	CLERK	4100	1000
8	NEAL	CLERK	900	900
9	SMITH	CLERK	1300	1000
10	SMITH	CLERK	1100	1100
11	SMITH	ANALYST	6000	2000
12	SMITH	ANALYST	3000	2000
13	SMITH	ANALYST	3000	2000
14	SMITH	MANAGER	5275	2575
15	SMITH	MANAGER	5275	2575
16	SMITH	MANAGER	5400	1400
17	SMITH	MANAGER	5600	1400

Este Group by divide los resultados por "Job" y por "Año"

Ejemplo de Group by (cube)

```
SELECT equipo, SUM(pts) AS Puntos, SUM(GD) AS GD, SUM(GF) AS GF
FROM equipos
GROUP BY GROUPING SETS ( (equipo), (equipo, pts), (equipo, pts, GD), (equipo, pts, GD, GF) )
ORDER BY Puntos DESC, GD DESC, GF DESC, equipo;
```

El resultado es largo, pero, en resumen: El Grouping sets hará grupos por cada paréntesis dentro, en este caso hay 4 grupos diferentes como en el siguiente ejemplo:

Ejemplo (PL/SQL)

```
-- DECLARE
-- Declara variables de variables locales y tipos de datos personalizados
Source warchart(2019)
Salario number(10,2);
Fecha date := sysdate;

BEGIN
-- DECLARACION PL/SQL
-- Inicializado por el usuario
-- create, insert, update, delete, y llamadas a procedimientos y funciones
Salario := 'Fabian Soto';
Salario := 5000;

-- Para imprimir el resultado
dbms_output.put_line('Nombre:');
dbms_output.put_line('Salario:');
dbms_output.put_line('Fecha: ' || to_char(sysdate, 'DD-MM-YYYY') || ' Hora: ' || to_char(sysdate, 'HH24:MI:SS') || ' Salario: ' || Salario);

select first_name, salary
from employees;

EXCEPTION
-- Manejo de errores
END;
```

Ejemplo (Cursores)

[illegible]

Ejemplo Procedimiento:

```

CREATE PROCEDURE
imprimir_saludo(nombre IN VARCHAR2) IS
BEGIN
DBMS_OUTPUT.PUT_LINE( 'Hola, ' ||
nombre || ' ');
END;

-- Uso del proced imiento
BEGIN
imprimir_saludo ('Juan');
END;

```

Ejemplo Función:

```
CREATE FUNCTION
totalCustomers
RETURN number IS
total number(2) := 0;
BEGIN
SELECT count(*) into total
FROM customers;
RETURN total;
END;
```