TECNICAS DE ALMACENAMIENTO DE DATOS

Pertenencia a conjuntos

Clausula IN

- La conectiva IN comprueba la pertenencia a un conjunto, donde el conjunto es la colección de valores resultado de una cláusula select
- La conectiva NOT IN comprueba la no pertenencia a un conjunto
- Los operadores IN y NOT IN también se pueden usar sobre conjuntos enumerados

Cláusula IN – Ejemplos

- Mostrar los empleados que sean programadores o manager de marketing

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY
201	Michael	Hartstein	MK_MAN	13000
103	Alexander	Hunold	IT_PROG	9000
104	Bruce	Ernst	IT_PROG	6000
106	Valli	Pataballa	IT_PROG	4800
105	David	Austin	IT_PROG	4800
107	Diana	Lorentz	IT_PROG	4200

Cláusula IN – Ejemplos

- Mostrar los empleados que sean programadores o manager de marketing

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB_ID, SALARY
  FROM EMPLOYEES
 WHERE JOB_ID IN ( SELECT JOB_ID
                     FROM JOBS
                    WHERE JOB_TITLE = 'Programmer'
                       OR JOB_TITLE = 'Marketing Manager' )
 ORDER BY SALARY DESC;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY
201	Michael	Hartstein	MK_MAN	13000
103	Alexander	Hunold	IT_PROG	9000
104	Bruce	Ernst	IT_PROG	6000
106	Valli	Pataballa	IT_PROG	4800
105	David	Austin	IT_PROG	4800
107	Diana	Lorentz	IT_PROG	4200

Cláusula IN – Ejemplos

- Mostrar los empleados que sean programadores o manager de marketing

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB_ID, SALARY
  FROM EMPLOYEES
 WHERE JOB_ID IN ( 'IT_PROG', 'MK_MAN' ) -- CONJUNTO ENUMERADO
 ORDER BY SALARY DESC;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY
201	Michael	Hartstein	MK_MAN	13000
103	Alexander	Hunold	IT_PROG	9000
104	Bruce	Ernst	IT_PROG	6000
106	Valli	Pataballa	IT_PROG	4800
105	David	Austin	IT_PROG	4800
107	Diana	Lorentz	IT_PROG	4200

Cláusula NOT IN – Ejemplo

- Mostrar los empleados que NO sean programadores ni manager de marketing

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB_ID, SALARY
  FROM EMPLOYEES
 WHERE JOB_ID NOT IN ( 'IT_PROG', 'MK_MAN' )
 ORDER BY SALARY DESC;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY
100	Steven	King	AD_PRES	24000
101	Neena	Kochhar	AD_VP	17000
102	Lex	De Haan	AD_VP	17000
145	John	Russell	SA_MAN	14000
146	Karen	Partners	SA_MAN	13500
205	Shelley	Higgins	AC_MGR	12008
108	Nancy	Greenberg	FI_MGR	12008
147	Alberto	Errazuriz	SA_MAN	12000

Pertenencia a conjuntos

Clausula ANY

- Es usado para comparar un valor con una lista de valores o una subquery
- Debe estar precedida por =, !=, >, <, <=, >= y seguido por una lista de valores o una subquery
- Cuando la condición ANY es seguida por una lista de valores, el optimizador expande la condición inicial a todos los elementos de la lista y las encadena con operadores OR

Cláusula ANY – Ejemplos

- Mostrar los empleados que sean programadores o manager de marketing

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB_ID, SALARY
  FROM EMPLOYEES
 WHERE JOB_ID = ANY ( 'IT_PROG' , 'MK_MAN' )
 ORDER BY SALARY DESC;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY
201	Michael	Hartstein	MK_MAN	13000
103	Alexander	Hunold	IT_PROG	9000
104	Bruce	Ernst	IT_PROG	6000
106	Valli	Pataballa	IT_PROG	4800
105	David	Austin	IT_PROG	4800
107	Diana	Lorentz	IT_PROG	4200

Cláusula ANY – Ejemplos

- Mostrar los empleados que perciban un salario mayor al salario de algún manager

```
--SOLUCION
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB_ID, SALARY
  FROM EMPLOYEES
 WHERE SALARY > ANY( SELECT SALARY
                       FROM EMPLOYEES
                      WHERE JOB_ID LIKE '%MAN' )
 ORDER BY SALARY DESC;

--MINIMO SALARIO DE MANAGERS
SELECT MIN(SALARY)
  FROM EMPLOYEES
 WHERE JOB_ID LIKE '%MAN';
```

Pertenencia a conjuntos

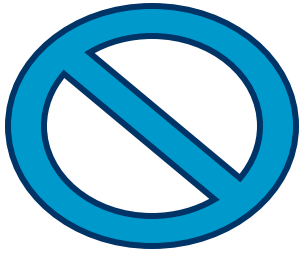
Clausula ALL

- Es usado para comparar un valor con una lista de valores o una subquery
- Debe estar precedido por =, !=, >, <, <=, >= y seguido por una lista de valores o una subquery
- Cuando la condición ALL es seguida por una lista de valores, el optimizador expande la condición inicial a todos los elementos de la lista y las encadena con operadores AND

Cláusula ALL – Ejemplos

- Mostrar los empleados que sean programadores y manager de marketing usando cláusula ALL

```
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB_ID, SALARY
  FROM EMPLOYEES
 WHERE JOB_ID = ALL ( 'IT_PROG' , 'MK_MAN' )
 ORDER BY SALARY DESC;
```



Cláusula ALL – Ejemplos

- Mostrar los empleados que perciban un salario mayor a cada salario de todos los empleados que son managers

```
--SOLUCION
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, JOB_ID, SALARY
  FROM EMPLOYEES
 WHERE SALARY > ALL( SELECT SALARY
                       FROM EMPLOYEES
                      WHERE JOB_ID LIKE '%MAN' )
 ORDER BY SALARY DESC;

--MAXIMO SALARIO DE MANAGERS
SELECT MAX(SALARY)
  FROM EMPLOYEES
 WHERE JOB_ID LIKE '%MAN';
```

Ejercicio

- Elaborar una consulta que muestre el empleado que tiene el mayor sueldo.

  - ✓ ORDER BY
  - ✓ MAX
  - ✓ ALL
  - ✓ …

Pruebas de relaciones vacías

- SQL incluye la posibilidad de comprobar si una subconsulta no produce ninguna tupla como resultado.
- La sentencia **EXISTS** devuelve el valor TRUE si la subconsulta argumento no es vacía. Devuelve FALSE si la subconsulta no retorna registros.
- Oracle prueba la existencia de datos que coinciden con la subconsulta, no retorna ningún registro. Termina la recuperación de registros cuando por lo menos un registro cumple la condición.

```
SELECT *
  FROM TABLA
 WHERE EXISTS ( ARGUMENTO );
```

Cláusula EXISTS – Ejemplos

- Elaborar una consulta que muestre las regiones que tienen algún país configurando.

```
SELECT *
  FROM REGIONS R
 WHERE EXISTS ( SELECT 1
                  FROM COUNTRIES C
                 WHERE C.REGION_ID = R.REGION_ID );
```

Cláusula EXISTS – Ejemplos

- Elaborar una consulta que muestre las regiones que NO tienen algún país configurando.

```
SELECT *
  FROM REGIONS R
 WHERE NOT EXISTS ( SELECT *
                      FROM COUNTRIES C
                     WHERE C.REGION_ID = R.REGION_ID );
```

Ejercicio – EXISTS

- Mostrar los datos de todos los empleados que tengan histórico de funciones creado (tabla JOB_HISTORY)

```
SELECT *
  FROM TABLA
WHERE EXISTS ( ARGUMENTO )
```

DER – ESQUEMA HR