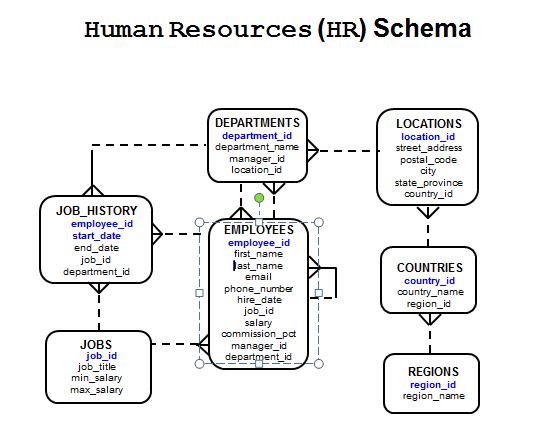
**CONSULTAS**

**-- CONSULTA DE MULTIPLES TABLAS**

*----------------------------------*

*-- 1. NATURAL JOINS*

*----------------------------------*

Combinación natural: realiza un join entre “**dos tablas”** cuando los campos por los cuales se enlazan tienen el mismo nombre y tipo de datos. Involucra claves primarias y foráneas.

SELECT DEPARTMENT\_ID, DEPARTMENT\_NAME, LOCATION\_ID, CITY

FROM HR.DEPARTMENTS

NATURAL JOIN HR.LOCATIONS

ORDER BY 1

SELECT COUNTRY\_NAME, REGION\_NAME

FROM HR.COUNTRIES

NATURAL JOIN HR.REGIONS

WHERE HR.REGIONS.REGION\_NAME = 'Europe'

SELECT FIRST\_NAME , LAST\_NAME , JOB\_TITLE

FROM EMPLOYEES

NATURAL JOIN JOBS

SELECT \*

FROM DEPARTMENTS

NATURAL JOIN LOCATIONS

NATURAL JOIN COUNTRIES

NATURAL JOIN REGIONS

*----------------------------------*

*-- 2. USING CLAUSE*

*----------------------------------*

Combinación empleando la cláusula "using": permite especificar el campo (o los campos) por el cual se enlazarán las tablas; los campos de ambas tablas DEBEN tener el mismo nombre y ser de tipos compatibles.

SELECT EMPLOYEE\_ID, LAST\_NAME, LOCATION\_ID, DEPARTMENT\_ID

FROM HR.EMPLOYEES

JOIN HR.DEPARTMENTS USING (DEPARTMENT\_ID)

WHERE DEPARTMENT\_ID = 100;

SELECT FIRST\_NAME , LAST\_NAME , JOB\_TITLE

FROM EMPLOYEES

JOIN JOBS USING ( JOB\_ID )

SELECT FIRST\_NAME , LAST\_NAME , JOB\_TITLE

FROM EMPLOYEES

JOIN JOBS USING ( JOB\_ID);

------EMP / DEP

SELECT FIRST\_NAME , LAST\_NAME , DEPARTMENT\_NAME

FROM EMPLOYEES

JOIN DEPARTMENTS USING ( DEPARTMENT\_ID);

*---------------------------------------------------------------------*

*-- 3. USANDO ALIAS DE TABLAS CON CAMPOS REFERENCIADOS EN CLAUSE USING*

*---------------------------------------------------------------------*

Las columnas utilizadas en USING no deben ser referenciadas con alias de tablas.

ERROR

SELECT L.CITY, D.DEPARTMENT\_NAME

FROM LOCATIONS L JOIN DEPARTMENTS D

USING (LOCATION\_ID)

WHERE D.LOCATION\_ID = 1400;

OK

SELECT L.CITY, D.DEPARTMENT\_NAME

FROM LOCATIONS L JOIN DEPARTMENTS D

USING (LOCATION\_ID)

WHERE LOCATION\_ID = 1400;

*----------------------------------*

*-- 4. JOINS CON CLAUSULA ON*

*----------------------------------*

Use la clusula ON para identificar las columnas que une la consulta. Con ello separa de la condición de filtro.

Ejemplo1 : 2 tablas consultadas

SELECT E.EMPLOYEE\_ID, E.LAST\_NAME, E.DEPARTMENT\_ID,

D.DEPARTMENT\_ID, D.LOCATION\_ID

FROM HR.EMPLOYEES E

JOIN HR.DEPARTMENTS D ON (E.DEPARTMENT\_ID = D.DEPARTMENT\_ID);

Ejemplo2 : 3 tablas consultadas

SELECT EMPLOYEE\_ID, CITY, DEPARTMENT\_NAME

FROM EMPLOYEES E

JOIN DEPARTMENTS D ON D.DEPARTMENT\_ID = E.DEPARTMENT\_ID

JOIN LOCATIONS L ON D.LOCATION\_ID = L.LOCATION\_ID;

Ejemplo3 : 2 tablas consultadas con filtro use AND o WHERE con los mismos resultados

SELECT E.EMPLOYEE\_ID, E.LAST\_NAME, E.DEPARTMENT\_ID,

D.DEPARTMENT\_ID, D.LOCATION\_ID

FROM HR.EMPLOYEES E JOIN HR.DEPARTMENTS D

ON (E.DEPARTMENT\_ID = D.DEPARTMENT\_ID)

AND E.MANAGER\_ID = 149 ;

SELECT E.EMPLOYEE\_ID, E.LAST\_NAME, E.DEPARTMENT\_ID,

D.DEPARTMENT\_ID, D.LOCATION\_ID

FROM HR.EMPLOYEES E JOIN HR.DEPARTMENTS D

ON (E.DEPARTMENT\_ID = D.DEPARTMENT\_ID)

WHERE E.MANAGER\_ID = 149 ;

*----------------------------------*

*-- 5. JOINS A LA MISMA TABLA ( RECURSIVIDAD )*

*----------------------------------*

Utilizada para casos de tablas recursivas

SELECT WORKER.LAST\_NAME EMP, MANAGER.LAST\_NAME MGR

FROM EMPLOYEES WORKER JOIN EMPLOYEES MANAGER

ON (WORKER.MANAGER\_ID = MANAGER.EMPLOYEE\_ID);

*----------------------------------*

*-- 6. LEFT OUTER JOIN*

*----------------------------------*

Esta consulta recupera todas las filas de la tabla EMPLEADOS, que es la tabla de la izquierda, incluso si no hay ninguna coincidencia en la tabla DEPARTAMENTOS.

SELECT E.LAST\_NAME, E.DEPARTMENT\_ID, D.DEPARTMENT\_NAME

FROM EMPLOYEES E

LEFT OUTER JOIN DEPARTMENTS D ON (E.DEPARTMENT\_ID = D.DEPARTMENT\_ID) ;

*----------------------------------*

*-- 7. RIGHT OUTER JOIN*

*----------------------------------*

Esta consulta recupera todas las filas de la tabla DEPARTAMENTOS, que es la tabla de la derecha, incluso si no hay ninguna coincidencia en la tabla EMPLEADOS.

SELECT e.last\_name, d.department\_id, d.department\_name

FROM employees e RIGHT OUTER JOIN departments d

ON (e.department\_id = d.department\_id) ;

*----------------------------------*

*-- 8. FULL OUTER JOIN*

*----------------------------------*

Esta consulta recupera todas las filas de la tabla EMPLEADOS, incluso si no hay ninguna coincidencia en la tabla DEPARTAMENTOS. También recupera todas las filas en la tabla de departamentos, incluso si no hay ninguna coincidencia en la tabla EMPLEADOS.

SELECT e.last\_name, d.department\_id, d.department\_name

FROM employees e FULL OUTER JOIN departments d

ON (e.department\_id = d.department\_id) ;

*----------------------------------*

*-- 9. ROWNUM*

*----------------------------------*

Es una pseudocolumna ( no es real ), e**s un número virtual que se le asigna a los registros devueltos por una select.**

SELECT DEPTNO, DNAME , ROWNUM

FROM SCOTT.DEPT;

SELECT DEPTNO, DNAME , ROWNUM

FROM SCOTT.DEPT

WHERE DEPTNO > 20 AND ROWNUM < 3

SELECT \* FROM (

SELECT ENAME , SAL

FROM SCOTT.EMP ORDER BY SAL )

WHERE ROWNUM < 5

*----------------------------------*

*-- 10. CASE*

*----------------------------------*

*SELECT LAST\_NAME , SALARY,*

*CASE JOB\_ID*

*WHEN 'SH\_CLERK' THEN SALARY \* 1.1*

*WHEN 'HR\_REP' THEN SALARY \* 1.2*

*ELSE SALARY*

*END*

*FROM EMPLOYEES;*