

Práctica 3. Recuperación de datos con subconsultas y subconsultas multi-registro

Índice

- Ampliación conceptos (alias, ordenación, manipulación cadenas, etc)
- Subconsultas
- Subconsultas multi-registro

Índice

- **Ampliación conceptos (alias, ordenación, manipulación cadenas, etc)**
- Subconsultas
- Subconsultas multi-registro

Ampliación de conceptos: Comando IN

- El **comando IN** se usa para localizar varios **valores dentro de una lista**
 - `SELECT * FROM votantes WHERE estudiossuperiores IN ('Basicos', 'Superiores', 'Doctorado');`
 - `SELECT * FROM votantes WHERE estudiossuperiores NOT IN ('Ninguno');`
- Igualmente se puede aplicar sobre **números**
 - `SELECT * FROM votantes WHERE localidad IN (1, 2, 3, 4, 5);`

Ampliación de conceptos: Ordenación

- Los **resultados de una consulta** puede **ordenarse** en base a algún criterio (ascendente o descendente).
 - **Ordenación ascendente:**
 - SELECT nombrecompleto, localidad FROM votantes ORDER BY localidad;
 - SELECT nombrecompleto, localidad FROM votantes ORDER BY localidad ASC;
 - SELECT nombrecompleto, localidad FROM votantes ORDER BY localidad ASC, nombrecompleto ASC;
 - **Ordenación descendente:**
 - SELECT nombrecompleto, localidad FROM votantes ORDER BY localidad DESC;
 - **Combinación:**
 - SELECT nombrecompleto, localidad FROM votantes ORDER BY localidad DESC, nombrecompleto ASC;

Ampliación de conceptos: Alias

- Las tablas pueden renombrarse para simplificar la consulta
 - Antes de usar alias:
 - `SELECT votantes.nombrecompleto, localidades.nombre FROM votantes, localidades WHERE votantes.localidad=localidades.idlocalidad;`
 - Después de usar alias:
 - `SELECT a.nombrecompleto, b.nombre FROM votantes a, localidades b WHERE a.localidad=b.idlocalidad;`
- Las columnas pueden renombrarse y usar su alias
 - Usando AS para definir el alias
 - `SELECT a.nombrecompleto, b.nombre AS nom FROM votantes a, localidades b WHERE a.localidad=b.idlocalidad ORDER BY nom;`
 - Sin usar AS para definir el alias
 - `SELECT a.nombrecompleto, b.nombre nom FROM votantes a, localidades b WHERE a.localidad=b.idlocalidad ORDER BY nom;`

Ampliación de conceptos: Manipulación de cadenas

- Funciones específicas de manipulación de cadenas:
 - **Concatenación:** CONCAT('Ho','la') devuelve Hola
 - SELECT CONCAT(CONCAT(a.nombrecompleto,' es de '),b.nombre) FROM votantes a, localidades b WHERE a.localidad=b.idlocalidad;
 - **Posición letra:** INSTR('Hola', 'o') devuelve 2
 - **Subcadena:** SUBSTR('Hola',1,3) devuelve Hol
 - SELECT SUBSTR(nombrecompleto,1,INSTR(nombrecompleto,' ')) FROM votantes;
 - **Longitud:** LENGTH('Hola') devuelve 4
 - SELECT SUBSTR(nombrecompleto,1,INSTR(nombrecompleto,' ')-1) FROM votantes WHERE LENGTH(SUBSTR(nombrecompleto,1,INSTR(nombrecompleto,' ')-1))<=4;

Ampliación de conceptos: Consultas condicionales

- La **función DECODE**, que hace las veces de una sentencia CASE para facilitar **consultas condicionales**
 - **Usando AS** para definir el alias de la nueva columna
 - SELECT nombrecompleto, DECODE(estudiossuperiores, 'Basicos', 1500, 'Superiores', 2000, 'Doctorado', 2500, 1000) salario FROM votantes;
 - **Sin usar AS** para definir el alias de la nueva columna
 - SELECT nombrecompleto, DECODE(estudiossuperiores, 'Basicos', 1500, 'Superiores', 2000, 'Doctorado', 2500, 1000) AS salario FROM votantes;

Índice

- Ampliación conceptos (alias, ordenación, manipulación cadenas, etc)
- **Subconsultas**
- Subconsultas multi-registro

Subconsultas

- Las **subconsultas se ejecutan una vez y antes de la consulta principal**
- El resultado de la subconsulta es usado por la consulta principal externa
 - `SELECT nombre FROM localidades WHERE numerohabitantes > (SELECT numerohabitantes FROM localidades WHERE nombre='Cordoba');`
 - La subconsulta se ejecuta primero y devuelve el número de habitantes que tiene Córdoba. Justo después se ejecuta la consulta principal. Sería lo mismo que decir:
 - `SELECT nombre FROM localidades WHERE numerohabitantes>326609;`

Subconsultas

- Las **subconsultas se pueden aplicar en cualquier zona de la consulta principal**, como por ejemplo en el FROM
 - SELECT nombre FROM (SELECT * FROM localidades WHERE provincia=1) WHERE numerohabitantes>20000;
- El anterior es sólo un ejemplo, que realmente podría hacerse también sin subconsultas
 - SELECT nombre FROM localidades WHERE provincia=1 AND numerohabitantes>20000;

Índice

- Ampliación conceptos (alias, ordenación, manipulación cadenas, etc)
- Subconsultas
- **Subconsultas multi-registro**

Subconsultas multi-registro

- Las subconsultas multi-registro **devuelven más de un registro**, por lo que **hay que usar comparadores multi-registro**
- **Tipos de comparadores multi-registro:**
 - **ANY:** compara los valores con cada valor devuelto por la subconsulta
 - **ALL:** compara los valores con cada uno de los valores devueltos por la subconsulta
- Ejemplo donde se busca el nombre de las localidades cuyo numero de habitantes es mayor que todas las localidades de la provincia numero 1
 - `SELECT nombre FROM localidades WHERE numerohabitantes > ALL (SELECT numerohabitantes FROM localidades WHERE provincia=1);`