SQL

Lenguaje de Consulta Estructurado

Indice de contenidos

- Introducción
- Consultas de Selección
- Criterios de Selección
- Agrupamiento de Registros y funciones agregadas
- Consultas de Acción
- Subconsultas
- Tipos de Consultas

Introducción

- Componentes del SQL
- Comandos
- Cláusulas
- Operadores lógicos
- Operadores de Comparación
- Funciones de Agregado

Criterios de Selección

- Operadores Lógicos
- Intervalos de Valores
- El Operador Like
- El Operador In
- La cláusula WHERE

Agrupamiento de registros y funciones agregadas

- La cláusula GROUP BY
- La cláusula HAVING
- AVG (Media Aritmética)
- Count (Contar Registros)
- Max y Min (Valores Máximos y Mínimos)
- StDev y StDevP (Desviación Estándar)
- Sum (Sumar Valores)
- Var y VarP (Varianza)

Tipos de Consultas

- Consultas de Referencias Cruzadas
- Consultas de Unión Internas
- Consultas de Unión Externas

Componentes del SQL

- El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Comandos

- Existen dos tipos de comandos SQL:
- Los DLL que permiten crear y definir nuevas bases de datos, campos e índices.
- Los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos (select, insert, update y delete)

Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Cláusula	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

Operadores lógicos

Operador	Uso
AND	Es el "y" lógico. Evalua dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdar si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

Operadores de Comparación

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor ó Igual que
>=	Mayor ó Igual que
=	Igual que
BETWEEN	Utilizado para especificar un intervalo de valores.
LIKE	Utilizado en la comparación de un modelo
In	Utilizado para especificar los valores de los registros a seleccionar

Funciones de Agregado

Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

– Ver Tabla

Funciones de Agregado

Función	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

Consultas de Selección

Se utilizan para indicar al motor de datos que devuelva información de las bases de datos, esta información es devuelta en forma de un conjunto de registros que pueden modificarse.

- Consultas Básicas
- Ordenar los registros
- Consultas con Predicado
- Alias
- Bases de Datos Externas

Consultas Básicas

Sintaxis: SELECT Atributos FROM Tabla

atributos: lista de propiedades que se deseen presentar

tabla: nombre-s de la-s tabla-s que contiene los atributos a presentar

SELECT Nombre, Telefono FROM Clientes;

Esta consulta devuelve un *recordset* con el campo nombre y teléfono de la tabla clientes.

Ordenar los Registros

- Para especificar el orden en que se desean recuperar los registros de las tablas se emplea la claúsula ORDER BY lista de atributos a ordenar.
- Se pueden ordenar los registros por mas de un atributo.
- Se puede especificar el orden de los registros: ascendente ASC, ó descendente, DESC.

Consultas con Predicado

El predicado se incluye entre la claúsula y el primer nombre del atributo a recuperar, los posibles predicados son:

- ALL
- TOP
- DISTINCT
- DISTINCTROW

ALL

Por defecto siempre se asume ALL.

Esto origina la selección de todos los registros que cumplen las condiciones de la instrucción.

```
SELECT ALL FROM Empleados;
SELECT * FROM Empleados;
```

TOP

Devuelve un cierto número de registros que entran entre al principio o al final de un rango especificado por una cláusula ORDER BY.

Se puede utilizar la palabra reservada PERCENT para devolver un cierto porcentaje de registros que caen al principio o al final de un rango especificado por la cláusula ORDER BY.

DISTINCT

Omite los registros que contienen datos duplicados en los campos seleccionados...

DISTINCTROW

Devuelve los registros diferentes de una tabla; a diferencia del predicado anterior que sólo se fijaba en el contenido de los campos seleccionados, éste lo hace en el contenido del registro completo independientemente de los campo indicados en la cláusula SELECT.

Alias

La palabra reservada AS se encarga de asignar el nombre que deseamos a la columna deseada.

Ejemplo:

```
SELECT DISTINCTROW Apellido AS Empleado
FROM Empleados;
```

Bases de Datos Externas

Se emplea la palabra reservada IN para recuperar información de bases de datos externas o que no se encuentran abiertas. Ejemplo:

```
SELECT DISTINCTROW Apellido AS Empleado FROM
Empleados IN 'c:\databases\gestion.mdb';
```

(donde c:\databases\gestion.mdb es la base de datos que contiene la tabla Empleados)

Operadores Lógicos

Los operadores lógicos soportados por SQL son: AND, OR, XOR, Eqv, Imp, Is y Not. A excepción de los dos últimos todos poseen la siguiente sintaxis:

<expresión1> operador <expresión2>

expresión1 y expresión2 son las condiciones a evaluar, el resultado de la operación varía en función del operador lógico.

La tabla adjunta muestra los diferentes posibles resultados:

Tabla de resultados

<expresión1>	Operador	<expresión2>	Resultado
Verdad	AND	Falso	Falso
Verdad	AND	Verdad	Verdad
Falso	AND	Verdad	Falso
Falso	AND	Falso	Falso
Verdad	OR	Falso	Verdad
Verdad	OR	Verdad	Verdad
Falso	OR	Verdad	Verdad
Falso	OR	Falso	Falso

Ejemplos:

Ejemplos:

- SELECT * FROM Empleados WHERE
  Edad > 25 AND Edad < 50;
- SELECT * FROM Empleados WHERE
  (Edad > 25 **AND** Edad < 50) OR Sueldo = 100;
- SELECT * FROM Empleados WHERE
  NOT Estado = 'Soltero';
- SELECT * FROM Empleados WHERE
  (Sueldo > 100 **AND** Sueldo < 500) OR
  (Provincia = 'Madrid' **AND** Estado = 'Casado');

Intervalos de Valores

Para recuperar registros según el intervalo de valores de un campo se emplea Between cuya sintaxis:

campo [Not] Between valor1 And valor2

(la consulta devuelve los registros que contengan en "campo" un valor incluido en el intervalo valor1, valor2 (ambos inclusive). Si anteponemos Not devolverá aquellos valores no incluidos en el intervalo.)

Between

SELECT IIF(CodPostal Between 28000 And 28999, 'Provincial', 'Nacional') FROM Editores;

(Devuelve el valor 'Provincial' si el código postal se encuentra en el intervalo, 'Nacional' en caso contrario)

El Operador Like

Se utiliza para comparar una expresión de cadena con un modelo en una expresión SQL. Sintaxis: expresión Like modelo

- Se puede utilizar el operador Like para encontrar valores en los campos que coincidan con el modelo especificado.
- Por modelo se puede especificar un valor completo o utilizar caracteres comodín para encontrar un rango de valores.

- Like 'P[A-F]###'

  devuelve los datos que comienzan con la letra P seguido de cualquier letra entre A y F y de tres dígitos:

- Like '[A-D]*'

  devuelve los campos cuyo contenido empiece con una letra de la A a la D seguidas de cualquier cadena.

- En la tabla siguiente se muestra cómo utilizar el operador Like para comprobar expresiones con diferentes modelos.

Operador Like

Tipo de coincidencia	Modelo Planteado	Coincide	No coincide
Varios caracteres	'a*a'	'aa', 'aBa', 'aBBBa'	'aBC'
Carácter especial	'a[*]a'	'a*a'	'aaa'
Varios caracteres	'ab*'	'abcdefg', 'abc'	'cab', 'aab'
Un solo carácter	'a?a'	'aaa', 'a3a', 'aBa'	'aBBBa'
Un solo dígito	'a#a'	'a0a', 'a1a', 'a2a'	'aaa', 'a10a'
Rango de caracteres	'[a-z]'	'f', 'p', 'j'	'2', '&'
Fuera de un rango	'[!a-z]'	'9', '&', '%'	'b', 'a'
Distinto de un dígito	'[!0-9]'	'A', 'a', '&', '~'	'0', '1', '9'
Combinada	'a[!b-m]#'	'An9', 'az0', 'a99'	'abc', 'aj0'

El Operador In

Este operador devuelve aquellos registros cuyo campo indicado coincide con alguno de los incluidos en una lista.
Sintaxis:

expresión [Not] In(valor1, valor2, . . .)

SELECT * FROM Pedidos WHERE Provincia In ('Madrid', 'Barcelona', 'Sevilla');

La cláusula WHERE

Determina qué registros de las tablas enumeradas en la cláusula FROM aparecerán en los resultados de la instrucción SELECT.

- Si no se emplea esta cláusula, la consulta devolverá todas las filas de la tabla.

Ejemplos

Claúsula WHERE

- SELECT Apellidos, Salario FROM Empleados WHERE Salario > 21000;
- SELECT * FROM Pedidos WHERE Fecha_Envio = #5/10/94#;
- SELECT Apellidos, Nombre FROM Empleados WHERE Apellidos Like 'S*';
- SELECT Apellidos, Salario FROM Empleados WHERE Salario Between 200 And 300;

Claúsula WHERE

- SELECT Apellidos, Salario FROM Empl WHERE Apellidos Between 'Lon' And 'Tol';
- SELECT Id_Pedido, Fecha_Pedido FROM Pedidos WHERE Fecha_Pedido Between #1-1-98# And #30-6-99#;
- SELECT Apellidos, Nombre, Ciudad FROM Empleados WHERE Ciudad In('Sevilla', 'Los Angeles', 'Barcelona');

La cláusula GROUP BY

Combina los registros con valores idénticos, en la lista de campos especificados, en un único registro. Para cada registro se crea un valor sumario si se incluye una función SQL agregada, como por ejemplo Sum o Count, en la instrucción SELECT. Sintaxis:

SELECT campos FROM tabla WHERE criterio GROUP BY campos del grupo

Group By

Se utiliza la cláusula WHERE para excluir aquellas filas que no desea agrupar, y la cláusula HAVING para filtrar los registros una vez agrupados.

Ejemplo:

```
SELECT Id_Familia, Sum(Stock) FROM Productos
GROUP BY Id_Familia;
```

HAVING

HAVING es similar a WHERE, determina qué registros se seleccionan. Una vez que los registros se han agrupado utilizando GROUP BY, HAVING determina cuales de ellos se van a mostrar. Ejemplo:

```
SELECT Id_Familia Sum(Stock) FROM Productos
GROUP BY Id_Familia    HAVING Sum(Stock) > 100
AND NombreProducto Like “BOS*”;
```

Avg(Media Aritmética)

Calcula la media aritmética (promedio) de un conjunto de valores contenidos en un campo especificado de una consulta. Sintaxis: Avg(expr)

- expr representa el campo que contiene los datos numéricos para los que se desea calcular la media o una expresión que realiza un cálculo utilizando los datos de dicho campo.

Ejemplo:

```
SELECT Avg(Gastos) AS Promedio FROM Pedidos
WHERE Gastos > 100;
```

Count (Contar Registros)

Calcula el número de registros devueltos por una consulta. Sintaxis: Count(expr)

- expr contiene el nombre del campo que desea contar. Los operandos de expr pueden incluir el nombre de un campo de una tabla, una constante o una función (la cual puede ser intrínseca o definida por el usuario pero no otras de las funciones agregadas de SQL).

Count(expr)

Count cuenta el número de registros sin tener en cuenta qué valores se almacenan en los registros.

- La función Count no cuenta los registros que tienen campos null a menos que expr sea el carácter comodín asterisco (*).
- Count(*) cuenta todos los registros y es considerablemente más rápida que Count(Campo). Ejemplo:
- SELECT Count(*) AS Total FROM Pedidos;

Max y Min (Valores Máximos y Mínimos)

Devuelven el mínimo o el máximo de un conjunto de valores contenidos en un campo específico de una consulta. Sintaxis:

Min(expr) ; Max(expr)

expr es el campo sobre el que se desea realizar el cálculo. Expr puede incluir el nombre de un campo de una tabla, una constante o una función.

StDev y StDevP (Desv. Estándar)

Devuelve estimaciones de la desviación estándar para la población (el total de los registros de la tabla) o una muestra de la población representada (muestra aleatoria) .
Sintaxis: StDev(expr) ; StDevP(expr)

- expr representa el nombre del campo que contiene los datos que desean evaluarse o una expresión que realiza un cálculo utilizando los datos de dichos campos.
- StDevP evalúa una población, y StDev evalúa una muestra de la población.

Sum(Sumar Valores)

Devuelve la suma del conjunto de valores contenido en un campo especifico de una consulta. Sintaxis: Sum(expr)

- expr respresenta el nombre del campo que contiene los datos que desean sumarse o una expresión que realiza un cálculo utilizando los datos de dichos campos. Los operandos de expr pueden incluir el nombre de un campo, una constante o una función.

```
SELECT Sum(PrecioUnidad * Cantidad) AS Total
FROM DetallePedido;
```

Var y VarP (Varianza)

Devuelve una estimación de la varianza de una población (sobre el total de los registros) o una muestra de la población (muestra aleatoria de registros) sobre los valores de un campo. Sintaxis:Var(expr);VarP(expr)

- VarP evalúa una población, y Var evalúa una muestra de la población.
- Los operandos de expr pueden incluir el nombre de un campo, una constante o una función.

Consultas de Acción

Las consultas de acción son aquellas que no devuelven ningún registro, son las encargadas de ejecutar acciones como añadir, borrar y/o modificar registros.

- Consultas de Eliminación. DELETE
- Consultas de Datos Añadidos. INSERT INTO
- Consultas de Actualización. UPDATE

Consultas de Eliminación

DELETE elimina los registros de una o más de las tablas listadas en la cláusula FROM que satisfagan la cláusula WHERE. Esta consulta elimina los registros completos, no es posible eliminar el contenido de algún campo en concreto. Sintaxis:

DELETE Tabla.* FROM Tabla WHERE criterio

Una vez eliminados los registros, no se puede deshacer la operación.

Subconsultas

- Una subconsulta es una instrucción SELECT anidada dentro de una instrucción SELECT, o dentro de otra subconsulta.
- Hay tres tipos de sintaxis para crear una subconsulta:
  - comparación [ANY | ALL | SOME] (inst. sql)
  - expresión [NOT] IN (inst. sql)
  - [NOT] EXISTS (inst. sql)

Subconsultas (continuaciónI)

donde:

- **comparación:** es una expresión y un operador de comparación que compara la expresión con el resultado de la subconsulta.
- **Expresión:** expresión por la que se busca el conjunto resultante de la subconsulta.
- **instrucción sql** : es una instrucción SELECT, que sigue el mismo formato y reglas que cualquier otra instrucción SELECT. Debe ir entre paréntesis.

Subconsultas (continuaciónII)

- Se puede utilizar una subconsulta en lugar de una expresión en la lista de campos de una instrucción SELECT o en una cláusula WHERE o HAVING.
- En una subconsulta, se utiliza una instrucción SELECT para proporcionar un conjunto de uno o más valores especificados para evaluar en la expresión de la cláusula WHERE o HAVING.

El predicado ANY ó SOME

- Se puede utilizar el predicado ANY o SOME (sinónimos), para recuperar registros de la consulta principal, que satisfagan la comparación con cualquier otro registro recuperado en la subconsulta.

Subconsultas (continuaciónIII)

El ejemplo siguiente devuelve todos los productos cuyo precio unitario es mayor que el de cualquier producto vendido con un descuento igual o mayor al 25 por ciento:

```
SELECT * FROM Productos
WHERE PrecioUnidad > ANY (SELECT PrecioUnidad
FROM DetallePedido WHERE Descuento >= 0 .25);
```

El predicado ALL

- se utiliza para recuperar sólo aquellos registros de la consulta principal que satisfacen la comparación con todos los registros recuperados en la subconsulta.
- Si se cambia ANY por ALL en el ejemplo anterior, la consulta devolverá únicamente aquellos productos cuyo precio unitario sea mayor que el de todos los productos vendidos con un descuento igual o mayor al 25 por ciento. Esto es mucho más restrictivo.

El predicado IN

- Se emplea para recuperar sólo aquellos registros de la consulta principal para los que algunos registros de la subconsulta contienen un valor igual.

- El ejemplo siguiente devuelve todos los productos vendidos con un descuento igual o mayor al 25 por ciento:

Ejemplo:

```
SELECT * FROM Productos WHERE IDProducto IN
 (SELECT IDProducto FROM DetallePedido WHERE
    Descuento >=  0.25);
```

- Inversamente se puede utilizar NOT IN para recuperar únicamente aquellos registros de la consulta principal para los que no hay ningún registro de la subconsulta que contenga un valor igual

El predicado EXIST

Se utiliza en comparaciones de verdad/falso para determinar si la subconsulta devuelve algún registro.

- El ejemplo siguiente devuelve los nombres de los empleados cuyo salario es igual o mayor que el salario medio de todos los empleados con el mismo título. A la tabla Empleados se le ha dado el alias T1:

Ejemplos

```
SELECT Apellido, Nombre, Titulo, Salario FROM
Empleados AS T1
 WHERE Salario >= ( SELECT Avg(Salario) FROM
    Empleados WHERE T1.Titulo = Empleados.Titulo )
    ORDER BY Titulo;
```

```
SELECT Apellidos, Nombre, Cargo, Salario FROM
Empleados
 WHERE Cargo LIKE "Agente Ven*" AND
 Salario > ALL ( SELECT Salario FROM Empleados
    WHERE  ( Cargo LIKE "*Jefe*") OR
                (Cargo LIKE "*Director*‟ ) );
```

- Obtiene una lista con el nombre, cargo y salario de todos los agentes de ventas cuyo salario es mayor que el de todos los jefes y directores.

```
SELECT DISTINCTROW NombreProducto,
Precio_Unidad FROM Productos
 WHERE Precio_Unidad =( SELECT Precio_Unidad
FROM Productos WHERE Nombre_Producto="Almíbar
anisado" );
```

- Obtiene una lista con el nombre y el precio unitario de todos los productos con el mismo precio que el almíbar anisado.

```
SELECT DISTINCTROW Nombre_Contacto, Nombre_Compañia,
Cargo_Contacto, Telefono
FROM Clientes
WHERE (ID_Cliente IN (SELECT DISTINCTROW ID_Cliente
    FROM Pedidos
    WHERE Fecha_Pedido>=# 04/1/00#<#07/1/00#));
```

- Obtiene una lista de las compañías y los contactos de todos los clientes que han realizado un pedido en el segundo trimestre de 2000

```
SELECT Nombre, Apellidos FROM Empleados AS E
WHERE EXISTS
  (SELECT * FROM Pedidos AS O WHERE
     O.ID_Empleado = E.ID_Empleado);
```

- Selecciona el nombre de todos los empleados que han reservado al menos un pedido.

```
SELECT DISTINCTROW Pedidos.Id_Producto, Pedidos.Cantidad,
 (SELECT DISTINCTROW Productos.Nombre FROM Productos
    WHERE Productos.Id_Producto = Pedidos.Id_Producto)
     AS ElProducto FROM Pedidos
     WHERE Pedidos.Cantidad > 150
     ORDER BY Pedidos.Id_Producto;
```

Recupera el Código del Producto y la Cantidad pedida de la tabla pedidos, extrayendo el nombre del producto de la tabla de productos.