Consultas Simples

Unidad 4
Bases de Datos - 1º DAW

Contenidos

1. El lenguaje DML de SQL
2. **La declaración SELECT**
3. Modificadores, cláusulas y operadores
4. Funciones de Agregación
5. Tratamiento de los valores nulos

2. La declaración SELECT

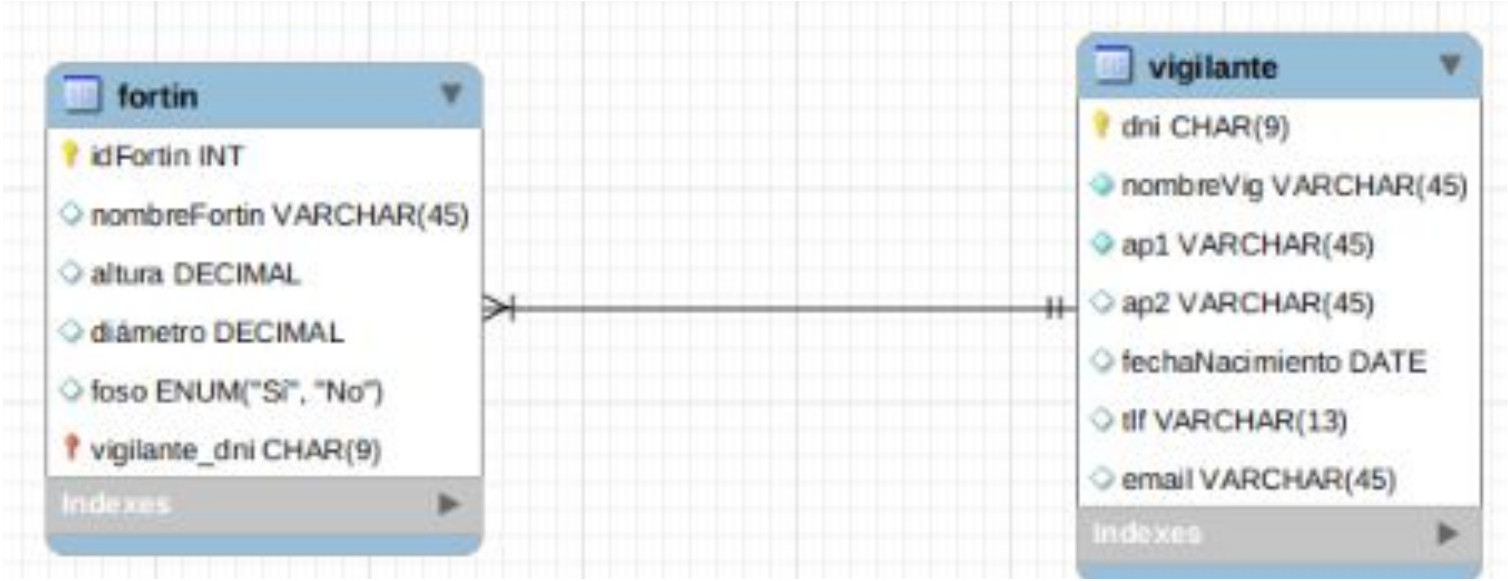
U4. Consultas Simples

Bases de Datos
1º DAW

Francisco Javier Castillo Jiménez

2.1. La sintaxis de la declaración SELECT.
2.2. Cláusulas.
2.3. Ejemplos de consultas simples.

2.0 Tablas para realizar las consultas de ejemplo



2.1 La sintaxis de la declaración SELECT

La documentación oficial de MySQL establece la sintaxis de la declaración **SELECT**. Básicamente se reduce a

SELECT [ALL | DISTINCT | DISTINCTROW] columnas
[**FROM** tablas]
[**WHERE** condiciones]
[**GROUP BY** columnas [ASC | DESC]]
[**HAVING** condición]
[**ORDER BY** columnas [ASC | DESC]]
[**LIMIT** filas];

Una consulta sólo necesita un SELECT pero una consulta simple, sobre una tabla, al menos se necesita **SELECT * FROM fortin;**

2.2 Cláusulas

Las cláusulas son las palabras reservadas que se pueden emplear en la declaración SELECT (que a la vez también forma una cláusula).

Así, se tienen las siguientes cláusulas:

- SELECT, indica las columnas.
- FROM, indica las tablas a utilizar
- WHERE, indica las condiciones que va a tener la consulta.
- GROUP BY - HAVING, empleada para consultas resumen.
- ORDER BY, va a ordenar los resultados de la consulta.
- LIMIT, limitará los resultados que ofrezca la consulta.

2.2.1 Cláusula SELECT

La cláusula SELECT permite indicar a la consulta cuáles serán las columnas que van a intervenir en la ejecución de ésta.

En teoría, serán las columnas que formen el resultado de la consulta.

Las opciones que se pueden indicar son:

- El **nombre de la columna** de la tabla sobre la que se realiza la columna. Tiene que ser una columna existente en la tabla que se indique en la cláusula FROM. Se pueden utilizar alias (**AS)** o ***.**
- Una **constante** que puede aparecer en todas las filas del resultado.
- Una **expresión** que permita calcular nuevos valores o mediante el uso de funciones.

2.2.2 Alias de las columnas

Los **alias** son pseudónimos para denominar a las columnas (también a las tablas de la cláusula FROM) que se crean con la palabra reservada **AS**

Pueden ser útiles para denominar a columnas creadas a partir de una o varias columnas.

Por ejemplo,

SELECT CONCAT_WS (“ “; nombre, ap1, ap2) **AS** nombreCompleto **FROM** vigilante_fortin **ORDER BY** nombreCompleto **ASC**;

Establecería un alias denominado *nombreCompleto* por el que despues se ordena la consulta de forma ascendente.

2.2.3 Cláusula FROM

En la cláusula FROM se establecerán todas las tablas que van a intervenir en la consulta.

A estas tablas se les puede otorgar un alias con la palabra reservada AS para darle un pseudónimo dentro de la consulta en cuestión.

Por ejemplo, para utilizar la tabla **fortin** se escribiría:

```
SELECT *
FROM fortin;
```

Se podrían anidar más tablas separadas por una coma (,) o empleando los diferentes operadores JOIN.

2.2.4 Cláusula WHERE

La cláusula WHERE se utiliza para establecer los diferentes filtros que tendrá la consulta.

Estos filtros harán que la consulta realice una búsqueda más exacta conforme a unas condiciones que se le establecerán en esta cláusula.

Para elaborar estas condiciones se utilizarán una serie de operadores que se verán en los próximos capítulos del tema.

```
// Muestra los fortines que tengan un diámetro mayor o igual a cien.
SELECT *
FROM fortin
WHERE diametro >= 100;
```

2.2.5 Cláusulas GROUP BY y HAVING

Estas dos cláusulas se van a emplear para las consultas que den un resultado de tipo resumen.

En la cláusula GROUP BY se le indicará la columna por la que se debe agrupar la información.

Por otro lado, en la cláusula HAVING se le indicará una condición para que la consulta agrupe posteriormente el resultado.

Normalmente, es necesario emplear la cláusula GROUP BY si va a utilizarse la cláusula HAVING.

Las consultas resumen se verán en unidades posteriores.

2.2.6 Cláusulas ORDER BY

El resultado de las consultas normalmente aparece en el mismo orden en el que ha sido insertada la información en las tablas.

Sin embargo, existe una forma de ordenar el resultado de las consultas por una o varias columnas de forma ascendente o descendente.

Por ejemplo,

```
// Muestra toda la información de los fortines ordenada de menor a mayor por su diámetro

SELECT *
FROM fortin
ORDER BY diametro DESC;
```

2.2.7 Cláusulas LIMIT

Una forma de limitar el número de filas que devuelve una consulta es empleando la cláusula LIMIT.

En esta cláusula puede tener uno o dos argumentos numéricos.

Si tiene un argumento, es el número de registros que va a mostrar a partir del primero. Si posee dos, devolvería a partir de un registro.

```
// Muestra las filas de la 6 a la 15

SELECT                                  *
FROM                                    fortin
LIMIT 5,10;
```

2.2.8 Orden de ejecución de las cláusulas

Es relevante conocer el orden en el que se ejecuta cada una de las cláusulas que forman la sentencia SELECT. El orden de ejecución es:

- Cláusula **FROM**.
- Cláusula **WHERE** (Opcional).
- Cláusula **GROUP BY** (Opcional).
- Cláusula **HAVING** (Opcional).
- Cláusula **SELECT**.
- Cláusula **ORDER BY** (Opcional).
- Cláusula **LIMIT** (Opcional).

Esta ejecución ofrecerá como resultado una tabla de datos.

2.3 Ejemplos de consultas simples

- **Obtener todos los datos de una tabla.**
  SELECT * FROM vigilante; -- Todos los datos de los vigilantes.
- **Obtener los datos de una columna de una tabla.**
  SELECT nombreVig FROM vigilante; -- Muestra el nombre de los v.
- **Obtener los datos de varias columnas de una tabla**
  SELECT nombreVig, ap1 FROM vigilante;
- **Obtener columnas calculadas**
  SELECT nombreFortin, altura/diametro FROM fortin;
- **Emplear una función**
  SELECT CONCAT_WS(“ “, nombreVig, ape1) AS “Nombre Completo” FROM vigilante;