

Bases de datos y SQL

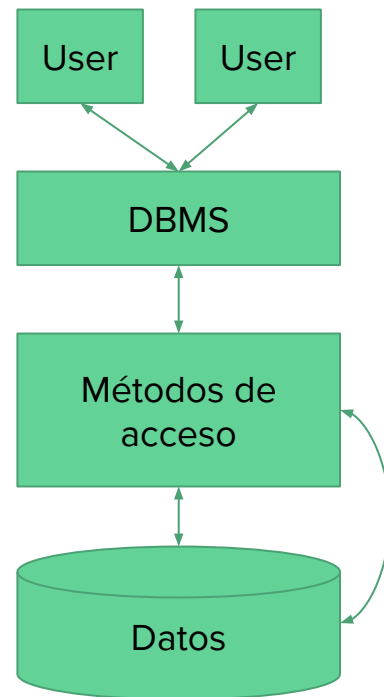
Bases de datos



Una **base de datos** es un sistema organizado para almacenar, administrar y recuperar grandes cantidades de **información** de manera eficiente y segura.

- Las bases de datos suelen funcionar como **servidores** en una arquitectura cliente-servidor.
- El servidor (software) de base de datos, *Database Management System* (DBMS) gestiona las solicitudes de datos de uno o varios clientes.
- Los DBMS utilizan algoritmos avanzados para gestionar el acceso a los datos.
- Estos servidores aseguran la integridad de los datos, controlan el acceso y permiten operaciones simultáneas de varios usuarios.

Ejemplo: Un observatorio virtual VO, ADS.



Bases de datos



Tipos de base de datos

Las bases de datos (BD) se pueden clasificar según la **variabilidad** de su contenido (estáticas o fijas); así como por la **naturaleza** de su contenido (bibliográficas, textos, directorios, datos numéricos/categoricos).

Modelos de base de datos

Según la forma en que se organizan los datos de una base datos, éstas también se pueden clasificar en: relacionales (SQL), no relacionales (NoSQL, elementos tipo JSON), jerárquicas (datos en forma de trees), de grafos, orientada a objetos, etc.

Las más utilizadas para almacenar información científica (datos numéricos y categoricos en forma de tablas) son las **bases de datos relacionales**.

Algunos DBMS usuales: Microsoft SQL server, MySQL, MariaDB, PostgreSQL

Esquemas de las bases de datos

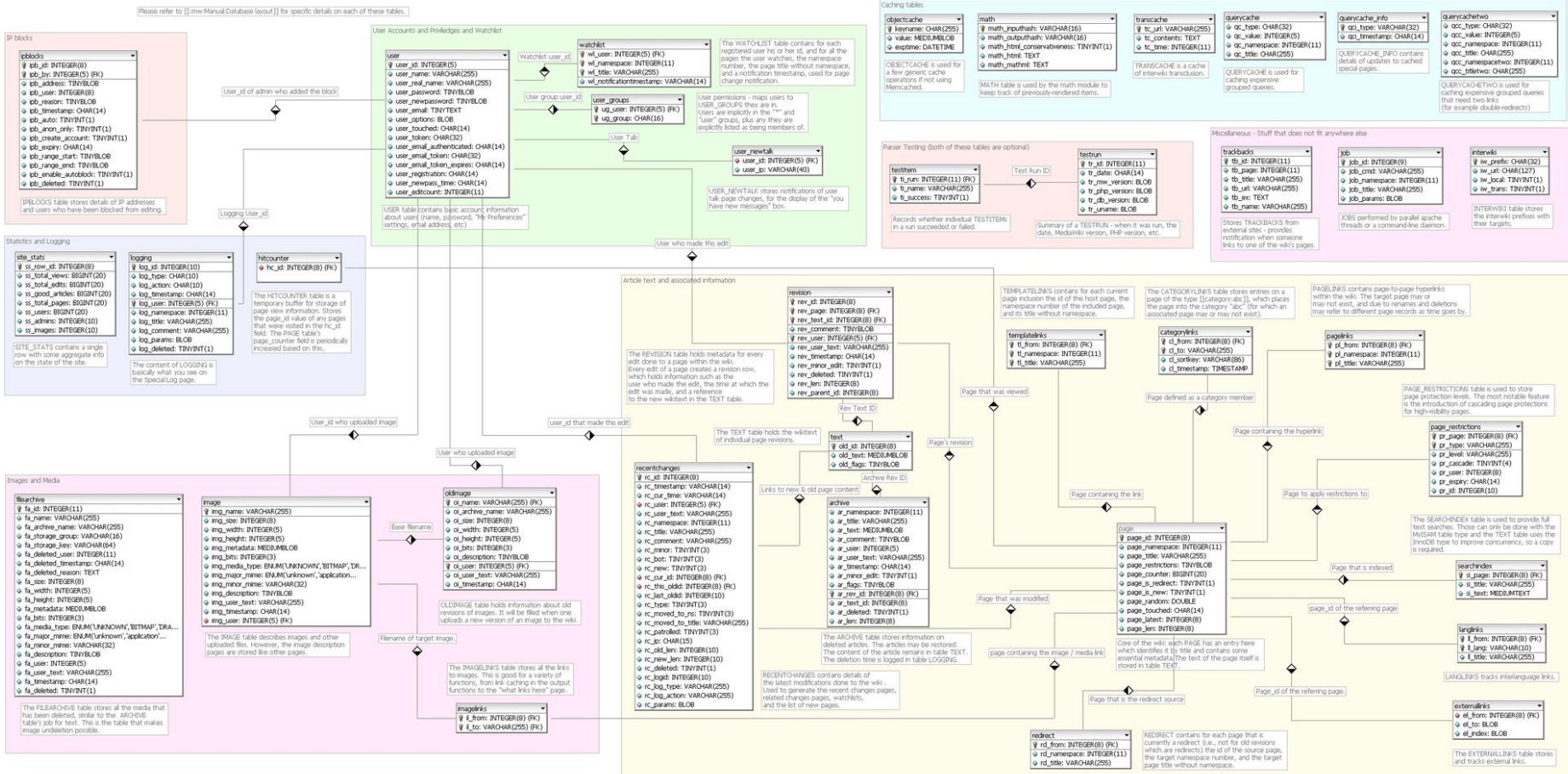
El esquema de una BD (*database schema*) **describe la estructura** de una base de datos, en un lenguaje formal soportado por un DBMS. Este se puede representar visualmente en forma de diagrama que nos permite comprender la organización y contenido de una BD.

En una **BD relacional**, el esquema define sus **tablas**, los **campos** en cada tabla y las **relaciones** entre cada campo y cada tabla (relaciones base y derivadas).

Ejemplo de esquema

- Tabla **Usuarios**: ID, Nombre, CorreoElectronico, FechaRegistro, etc
- Tabla **Pedidos**: ID, IDUsuario, FechaPedido, Detalles, etc

Ejemplo: esquema base de datos relacional compleja



SQL (Structured Query Language)



SQL es un lenguaje informático específico, diseñado para **administrar**, y **recuperar** información de sistemas DBMS. Una de sus principales características es el manejo del álgebra y el cálculo relacional para efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como realizar cambios en ellas.

A considerar:

- Permite especificar **qué** datos se desean obtener, **no cómo obtenerlos**.
- Operaciones Estándar: **CRUD** (Create, Read, Update, Delete)
- Independiente de la Plataforma: Funciona en múltiples DBMS.
- Utiliza instrucciones/comandos específicas para definir estructuras de datos, consultar datos, modificar datos, y controlar el acceso.

Actividad: Buscar y acceder a un servicio de BD relacional (SQL) que contenga datos relevantes para su trabajo de investigación (VO, SDSS, MillenniumDB, etc.).



SQL: instrucciones de solicitud (Read)

- **Selección de Tablas y Datos (SELECT, FROM)**

Utilizado para seleccionar datos de una o más tablas.

Sintaxis:

```
SELECT columnas FROM tabla;
```

Ejemplo:

```
SELECT Nombre, Correo FROM Usuarios;
```

- **Aplicación de Filtros (WHERE)**

Permite filtrar registros que cumplan ciertas condiciones.

Sintaxis:

```
SELECT columnas FROM tabla  
WHERE condición;
```

La condición puede contener AND, OR, operadores de comparación, IN, BETWEEN, LIKE.

Ejemplo:

```
SELECT * FROM Usuarios WHERE ID = 5;
```

- **Selección de Campos**

SELECT también nos permite definir qué campos se desea leer.

Selección de todos los campos:

```
SELECT * FROM tabla;
```

Selección de campos específicos:

```
SELECT campo1, campo2 FROM tabla;
```

Ejemplo:

```
SELECT Nombre, Fecha FROM Usuarios;
```

- **Limitar Resultados (LIMIT)**

Limita el número de registros solicitados.

Sintaxis:

```
SELECT columnas FROM tabla  
LIMIT número;
```

Ejemplo:

```
SELECT * FROM Usuarios LIMIT 10;
```



SQL: instrucciones de solicitud (Read)

- **Ordenación de Registros (ORDER BY)**

Permite ordenar los resultados (registros) según una o más columnas.

Sintaxis: **SELECT** columnas **FROM** tabla **ORDER BY** columna [**ASC**|**DESC**];

Ejemplo:

```
SELECT * FROM Usuarios ORDER BY Nombre ASC;
```

- **Funciones**

Permiten obtener información adicional de los datos:

COUNT() - retorna el número de registros

SUM() - Retorna la suma de un campo (columna) numérica.

AVG() - Retorna la media de un conjunto de valores

MIN() / **MAX()** - Obtiene el mínimo/máximo de una columna.

Ejemplos:

- **SELECT MAX(released) FROM albums;**

- **SELECT artist,album,released FROM albums
WHERE released = (SELECT MIN(released) FROM albums);**

- Más instrucciones de organización del resultado:

Agrupación de Datos: **GROUP BY**

Combinación de filas: **JOIN**