

Manipulación de bases de datos

Tema 1

Bases de Datos - Grado en Ingeniería Informática
Antonio Balderas Alberico

Comentarios

- Transparencias correspondientes al **capítulo 2** del documento “Apuntes de prácticas” del Campus Virtual:
 - <https://av03-19-20.uca.es/moodle/mod/resource/view.php?id=90729>
- Para el lenguaje MySQL, se aportan referencias web oficiales donde se amplía y especifica el uso de las diferentes instrucciones.

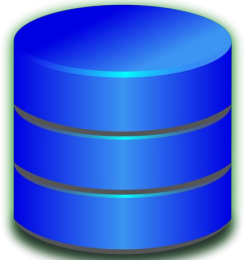
Base de datos

Una base de datos es una colección organizada de datos, generalmente almacenados y a los que se accede electrónicamente desde un sistema informático.



Sistema de gestión de bases de datos

Sistema de software que permite a los usuarios definir, crear, mantener y controlar el acceso a la base de datos [1]



Sistema de gestión de bases de datos

Ventajas:

- Control sobre la redundancia de datos
- Consistencia de datos
- Compartición de datos
- Mantenimiento de estándares
- Integridad de los datos
- Seguridad
- Accesibilidad
- Productividad



SGBDR

- SGBDR: Sistema de Gestión de Base de Datos Relacional
- Basados en el modelo relacional
- Modelo de datos basado en la lógica de predicados y en la teoría de conjuntos
- Lenguaje SQL: SQL (del inglés, Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje de dominio específico para administrar, y recuperar información de SGBDR

SGBDR

Todos los SGBDR se basan en el estándar SQL, aunque cada uno de ellos puede extender el lenguaje con funciones adicionales

Los SGBDR más utilizados son:

- Oracle
- MySQL
- PostgreSQL
- SQL Developer

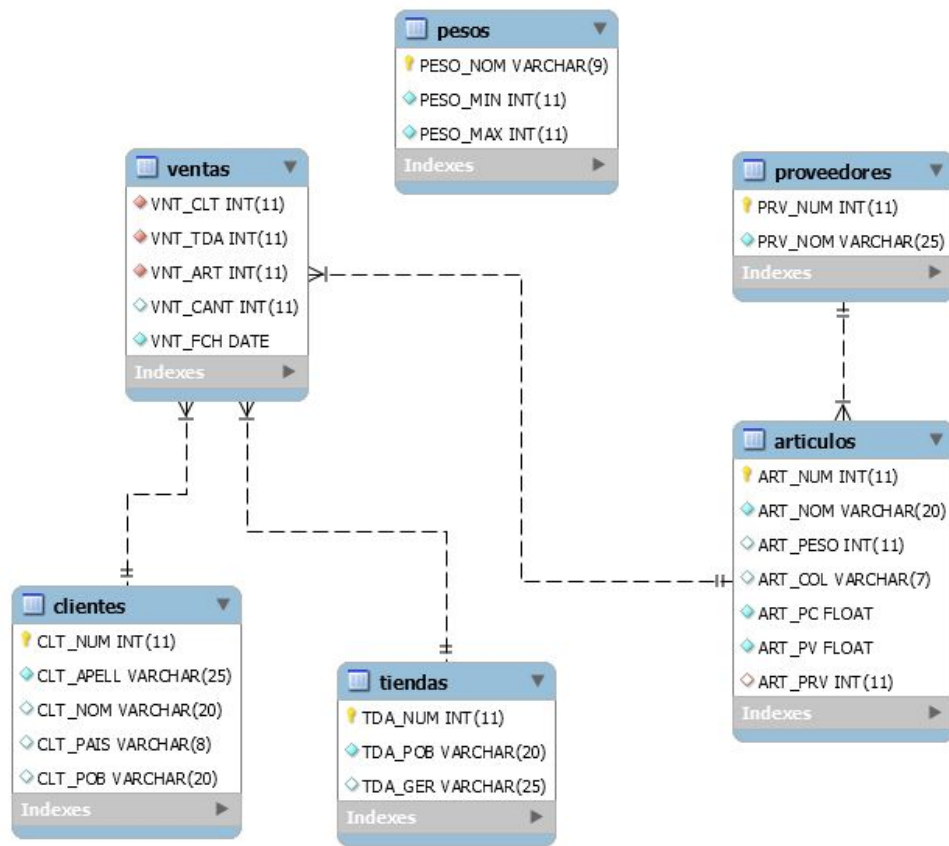
MySQL

- Licencia pública general - Licencia comercial por Oracle Corporation
- Base de datos de código abierto más popular del mundo [2]
- Muy popular en entornos de desarrollo web
- <https://www.mysql.com/>



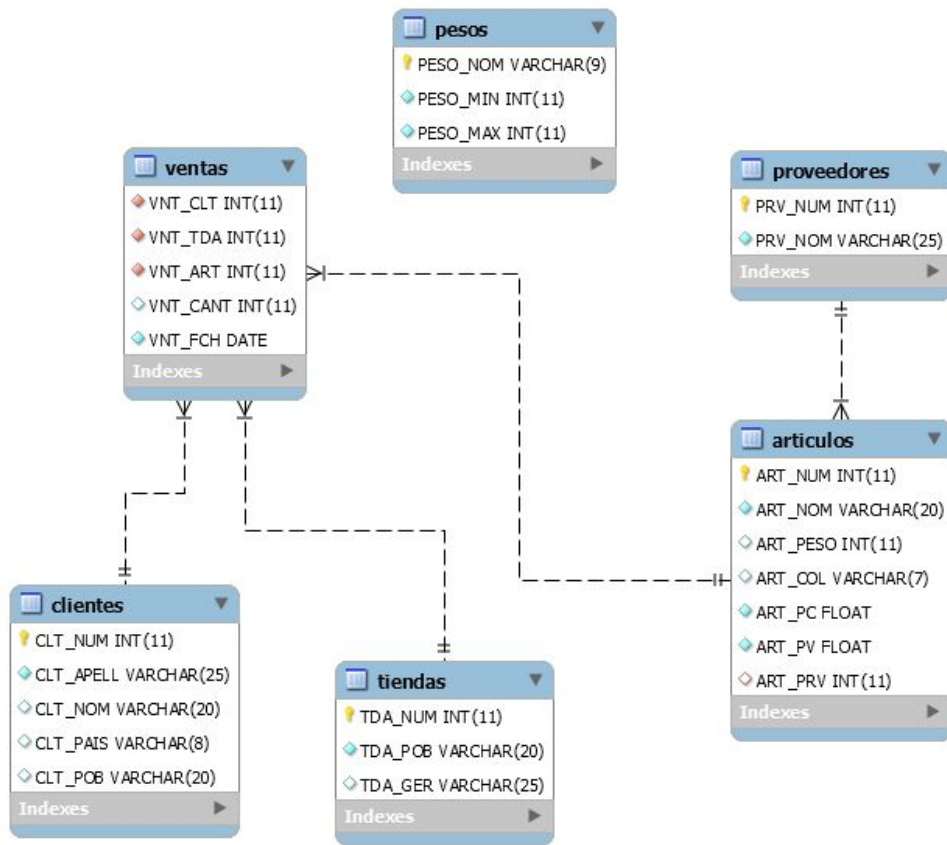
Ejemplo

- ¿Qué nos vamos a encontrar en un SGBD?



Ejemplo

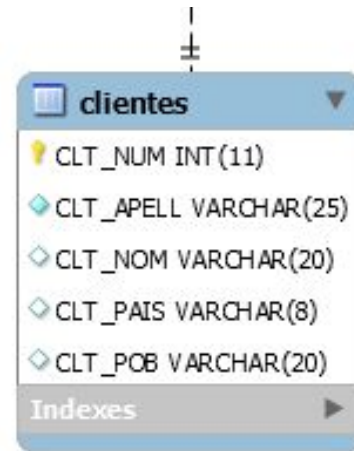
- ¿Qué nos vamos a encontrar en un SGBD?
- Tablas que almacenan registros con información estructurada conforme a las columnas especificadas



Ejemplo

- Según la estructura de la tabla clientes, los registros podrían contener la siguiente información:

| CLT_NUM | CLT_APELL | CLT_NOM | CLT_PAIS | CLT_POB |
|---------|-----------|---------|----------|---------|
| 1 | Gómez | Elsa | e | Cádiz |
| 2 | Sánchez | José | e | Cádiz |
| 3 | Romero | Elías | e | Ubrique |

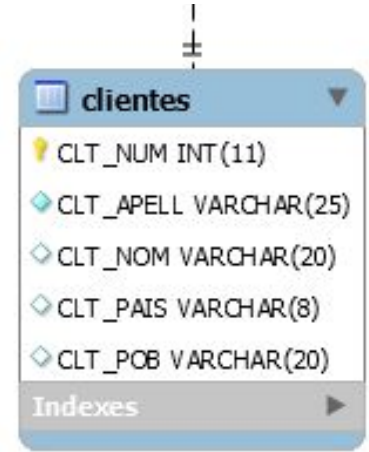


SQL

Proyección: operación del álgebra relacional que consiste en seleccionar una o varias columnas o atributos de una o varias tablas, indicando el orden de aparición de izquierda a derecha de las columnas.

¿Cómo podría obtener una proyección de contenido de la tabla clientes?

Mediante una consulta SQL



SQL

Proyección: operación del álgebra relacional que consiste en seleccionar una o varias columnas o atributos de una o varias tablas (FROM), indicando el orden de aparición de izquierda a derecha de las columnas. Mediante un * se muestran todas las columnas de la tabla que aparece en el FROM.

SELECT nombre de las columnas

FROM nombre de las tablas

WHERE condiciones de selección de tuplas

ORDER BY números o nombres de las columnas

<https://dev.mysql.com/doc/refman/8.0/en/select.html>

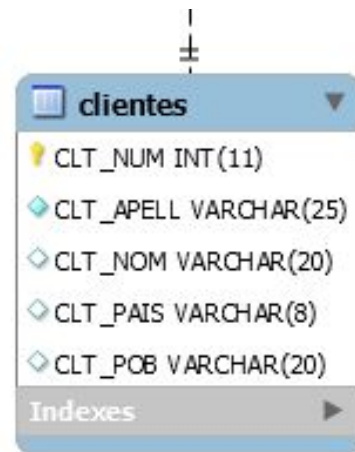
SQL

Ejemplo: Listado de clientes que muestre su nombre y población.

SELECT clt_nom, clt_pob

FROM clientes;

| CLT_NOM | CLT_POB |
|---------|---------|
| Elsa | Cádiz |
| José | Cádiz |
| Elías | Ubrique |



SQL

Mediante una consulta SELECT obtenemos una proyección, es decir, su contenido.

Es importante diferenciar entre estructura y contenido de una tabla:

- **Estructura:** nombre y tipo de cada una de las columnas que componen una tabla.
- **Contenido:** registros que almacenan información en esa tabla conforme a la estructura definida

SQL - describe

Diferenciar entre estructura y contenido de una tabla:

- La **estructura** de una tabla se obtiene mediante la instrucción '*describe nombre_de_la_tabla*'

Por ejemplo, la estructura de la tabla clientes se visualizará mediante la siguiente instrucción:

Describe clientes

| | | | |
|-----------|-------------|-----|-----|
| CLT_NUM | int(11) | NO | PRI |
| CLT_APELL | varchar(25) | NO | |
| CLT_NOM | varchar(20) | YES | |
| CLT_PAIS | varchar(8) | YES | |
| CLT_POB | varchar(20) | YES | |

SQL - tipos

Con la estructura de la tabla podemos ver el tipo de cada una de las columnas. Dependiendo del tipo, se podrá realizar unas operaciones u otras con la información contenido.

Los tipos de datos más utilizados son:

- INT
- FLOAT
- VARCHAR
- DATE

Más info en <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

SQL - where

Selección de filas

El criterio en una cláusula where se establece mediante una expresión lógica compuesta por una serie de condiciones, cada una de las cuales toma el valor verdadero o falso, combinado con los operadores lógicos not (negación), and (Y lógico) y or (O lógico). Si una la cumple la condición, hará que el valor de la expresión lógica sea verdadero, y formará parte del resultado

<https://www.mysqltutorial.org/mysql-where/>

SQL - where

Selección de filas

Las condiciones pueden utilizar los siguientes operadores:

- Comparación con un valor: =, <>, !=, <, >, <=, >=
- Comparación con un intervalo de valores: BETWEEN
- Comparación con una lista de valores: IN
- Comparación con un patrón: LIKE
- Búsqueda de valores no definidos: IS NULL

SQL - where

Selección de filas

Mediante el uso de la cláusula WHERE se puede determinar el subconjunto de tuplas que serán seleccionadas.

SELECT nombre de las columnas

FROM nombre de las tablas

WHERE condiciones de selección de tuplas

ORDER BY números o nombres de las columnas

SQL - where

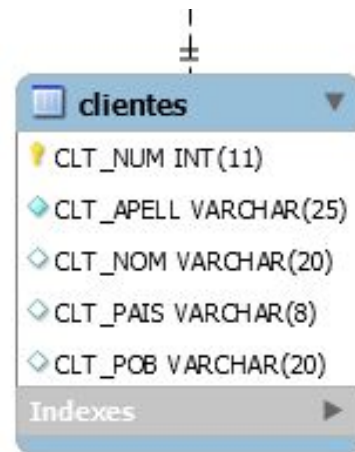
Ejemplo: Visualizar el nombre y la población de los clientes que son de Cádiz

SELECT clt_nom, clt_pob

FROM clientes

WHERE clt_pob = 'Cádiz'

| CLT_NOM | CLT_POB |
|---------|---------|
| Elsa | Cádiz |
| José | Cádiz |



SQL - where

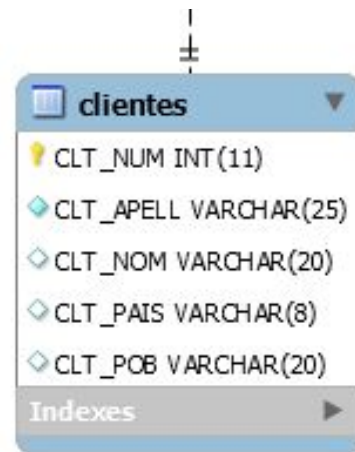
Ejemplo: Visualizar el nombre y la población de los clientes que son de Cádiz y se llaman José

SELECT clt_nom, clt_pob

FROM clientes

WHERE clt_pob = 'Cádiz' **AND** clt_nom = 'Elsa';

| CLT_NOM | CLT_POB |
|---------|---------|
| Elsa | Cádiz |



SQL - order by

Ordenación de filas

Mediante el uso de la cláusula ORDER BY se puede determinar el orden en que se mostrarán las tuplas seleccionadas.

SELECT nombre de las columnas

FROM nombre de las tablas

WHERE condiciones de selección de tuplas

ORDER BY números o nombres de las columnas. **ASC** (orden ascendente) o **DESC** (orden descendente)

SQL order by

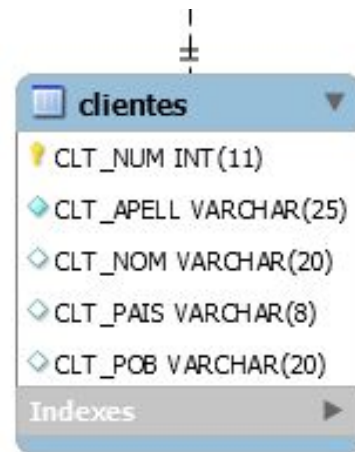
Ejemplo: Listado de clientes que muestre su nombre y población, ordenados por nombre

SELECT clt_nom, clt_pob

FROM clientes

ORDER BY clt_nom

| CLT_NOM | CLT_POB |
|---------|---------|
| Elías | Ubrique |
| Elsa | Cádiz |
| José | Cádiz |



Eliminación de registros repetidos

SQL no elimina las repeticiones

Eliminación de registros repetidos

SQL no elimina las repeticiones

```
SELECT art_col
```

```
FROM colores;
```

Eliminación de registros repetidos

SQL no elimina las repeticiones

SELECT art_col

FROM colores;

| ART_COL |
|----------|
| verde |
| rojo |
| rojo |
| verde |
| azul |
| amarillo |
| azul |
| morado |

Eliminación de registros repetidos

SQL no elimina las repeticiones

SELECT DISTINCT art_col

FROM colores;

DISTINCT elimina del resultado las repeticiones

https://www.w3schools.com/sql/sql_distinct.asp

| ART_COL |
|----------|
| verde |
| rojo |
| azul |
| amarillo |
| morado |

Renombrar columnas

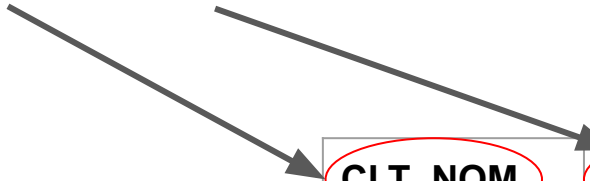
Los nombres de las columnas se pueden cambiar en la visualización del resultado

Renombrar columnas

Los nombres de las columnas se pueden cambiar en la visualización del resultado

SELECT clt_nom, clt_pob

FROM clientes;



The diagram shows two arrows originating from the SQL query. One arrow points from 'clt_nom' to the 'CLT_NOM' header of the table. The other arrow points from 'clt_pob' to the 'CLT_POB' header of the table. Both headers are circled in red.

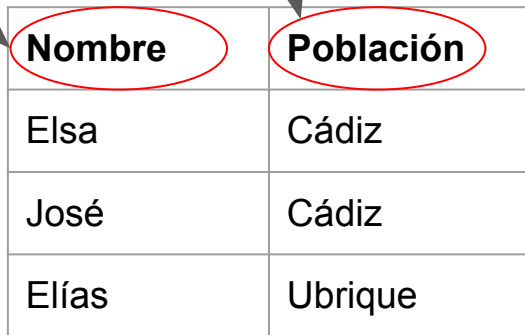
| CLT_NOM | CLT_POB |
|---------|---------|
| Elsa | Cádiz |
| José | Cádiz |
| Elías | Ubrique |

Renombrar columnas

Los nombres de las columnas se pueden cambiar en la visualización del resultado

SELECT clt_nom 'Nombre', clt_pob 'Población'

FROM clientes;



The diagram illustrates the mapping between SQL column aliases and the resulting table headers. Two arrows originate from the SQL query: one from the alias 'Nombre' in the SELECT clause pointing to the 'Nombre' header of the first column, and another from the alias 'Población' pointing to the 'Población' header of the second column. Both headers are circled in red in the original image.

| Nombre | Población |
|--------|-----------|
| Elsa | Cádiz |
| José | Cádiz |
| Elías | Ubrique |

Referencias

[1] Connolly, Thomas M.; Begg, Carolyn E. (2014). Database Systems – A Practical Approach to Design Implementation and Management (6th ed.). Pearson. ISBN 978-1292061184.

[2] Arias, Á. (2014). Bases de Datos con MySQL: 2ª Edición. IT Campus Academy.