

# Bases de datos

Introducción a las bases de datos  
(SQL)

# Bases de datos

SQL

# Bases de datos

## Enlaces

Enlaces:

- <https://es.wikipedia.org/wiki/SQL>
- <https://www.w3schools.com/sql/default.asp>
- <https://libros.catedu.es/books/bases-de-datos-relacionales-y-lenguaje-sql>
- Diagrama E-R (instalar): <http://dia-installer.de/index.html.es>
- Diagrama E-R (on line): <https://www.rollapp.com/>
- Modelo relacional (on line): <https://erdplus.com/>
- Lenguaje SQL (on line): <https://app.codingrooms.com/>

# Bases de datos

## Índice

- Lenguajes de bases de datos.
- SQL.
  - Estructura básica.
  - La clausula select.
  - La clausula where.
  - La clausula from.
  - La operación renombramiento.
  - Operaciones sobre cadenas.
  - Orden de la presentación de las tuplas.
  - Operaciones sobre conjuntos.
  - Funciones de agregación.
  - Valores nulos.

# Bases de datos

Bases de datos  
(Lenguajes de bases de datos)

# Bases de datos

## Lenguajes de bases de datos

# Bases de datos relacionales

## Lenguajes de bases de datos

- Un **sistema de bases de datos** proporciona un lenguaje de definición de datos para especificar el esquema de la base de datos y un lenguaje de manipulación de datos para expresar las consultas a la base de datos y las modificaciones.
- En la práctica, los lenguajes de definición y manipulación de datos no son dos lenguajes separados, sino que forman parte de un único lenguaje de bases de datos, como puede ser SQL, ampliamente utilizado.
- Suelen ser lenguajes de un nivel superior al de los lenguajes de programación habituales.

# Bases de datos relacionales

## Lenguajes de bases de datos

- SQL (Structured Query Language) es un lenguaje diseñado para administrar y recuperar información de sistemas de gestión de bases de datos relacionales.
- SQL permite ejecutar consultas para recuperar información de bases de datos, así como realizar cambios en ellas.
- SQL consiste en un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de control de datos.
- El alcance de SQL incluye la inserción de datos, consultas, actualizaciones y borrado, la creación y modificación de esquemas y el control de acceso a los datos.

# Bases de datos

Bases de datos  
(SQL)

# SQL

# SQL

# SQL

## Estructura básica

- La estructura básica de una instrucción SQL consiste en tres cláusulas: select, from y where.
- La cláusula **select** corresponde a la operación proyección del álgebra relacional. Se usa para listar los atributos deseados del resultado de una consulta.
- La cláusula **from** corresponde a la operación producto cartesiano del álgebra relacional. Lista las relaciones que deben ser analizadas en la evaluación de la expresión.
- La cláusula **where** corresponde al predicado selección del álgebra relacional. Es un predicado que engloba los atributos de las relaciones que aparecen en la cláusula from.

# SQL

## Estructura básica

Una consulta en SQL tiene la forma:

```
select A1, A2, ... , An  
from r1, r2, ... , rn  
where P
```

donde cada  $A_i$  representa un atributo y cada  $r_i$  representa una relación.  $P$  es un predicado.

SQL forma el producto cartesiano de las relaciones incluidas en la cláusula `from`, lleva a cabo la selección del álgebra relacional usando el predicado de la cláusula `where` y entonces proyecta el resultado sobre los atributos de la cláusula `select`.

# **SQL**

## **La cláusula select**

El resultado de una consulta SQL es una relación.

SQL permite duplicados en las relaciones y en el resultado de las expresiones SQL.

En el caso de que quieran eliminarse duplicados, se insertará la palabra clave `distinct` después de `select`.

# SQL

## La cláusula select

Una consulta en SQL con duplicados:

```
select A1, A2, ..., An
from r1, r2, ..., rn
where P
```

```
select all A1, A2, ..., An
from r1, r2, ..., rn
where P
```

Una consulta en SQL sin duplicados:

```
select distinct A1, A2, ..., An
from r1, r2, ..., rn
where P
```

# SQL

## La cláusula select

El símbolo asterisco “\*” se puede usar para denotar todos los atributos.

La cláusula select puede contener expresiones aritméticas que contengan los operadores +, -, \* y / operando sobre constantes o atributos de las tuplas.

SQL proporciona tipos de datos especiales, tales como varias formas del tipo fecha y permite varias funciones aritméticas para operar sobre esos tipos.

# SQL

## La cláusula where

SQL usa las conectivas lógicas and, or y not en la cláusula where. Los operandos de las conectivas lógicas pueden ser expresiones que contengan los operadores de comparación <, =<, >, >=, = y <>. SQL permite utilizar los operadores de comparación para comparar cadenas y expresiones aritméticas, así como tipos especiales, tales como el tipo fecha. SQL incluye un operador de comparación between para simplificar las cláusulas where que especifica que un valor sea menor o igual que un valor y mayor o igual que otro. También se puede usar el operador de comparación not between.

# SQL

## La cláusula from

La cláusula from define por si misma un producto cartesiano de las relaciones que aparecen en la cláusula.

# SQL

## La operación renombramiento

- La cláusula `as` permite renombrar tanto relaciones como atributos.  
`Nombre-antiguo as nombre-nuevo`
- La cláusula `as` puede aparecer tanto en `select` como en `from`.
- Casos en los que se suele usar:
  - Relaciones que aparecen en la cláusula `from` y que tienen atributos con el mismo nombre.
  - En la cláusula `select` se incluyen expresiones aritméticas.
  - Cuando se quiere cambiar el nombre de un atributo en el resultado.

# SQL

## Operaciones sobre cadenas

- SQL especifica las cadenas encerrándolas entre comillas simples.
- Un carácter comilla que sea parte de una cadena se puede especificar usando dos caracteres comilla.
- La operación más usada sobre cadenas es el encaje de patrones, para el que se usa el operador like.
- Existen dos caracteres especiales:
  - El carácter % encaja con cualquier subcadena.
  - El carácter \_ encaja con cualquier carácter.
- Los caracteres con mayúsculas no encajan con los caracteres en minúsculas y viceversa.

# SQL

## Operaciones sobre cadenas

- Los patrones se expresan en SQL utilizando el operador de comparación like.
- Para que los patrones puedan contener los caracteres especiales patrón (%) y (\_), SQL permite la especificación de un carácter de escape. El carácter de escape se utiliza inmediatamente antes de un carácter especial patrón para indicar que ese carácter especial va a ser tratado como un carácter normal. El carácter de escape para una comparación like se define utilizando la palabra clave escape.

# SQL

## Operaciones sobre cadenas

- SQL permite buscar discordancias en lugar de concordancias utilizando el operador de comparación not like.
- SQL también proporciona una variedad de funciones que operan sobre cadenas de caracteres, tales como la concatenación (||), la extracción de subcadenas, el cálculo de la longitud de las cadenas, la conversión a mayúsculas y minúsculas, etc.

# SQL

## Orden de la presentación de las tuplas

- SQL ofrece al usuario cierto control sobre el orden en el cual se presentan las tuplas de la relación.
- La cláusula `order by` hace que las tuplas resultantes de una consulta se presenten en un cierto orden.
- De manera predeterminada, la cláusula `order by` lista los elementos en orden ascendente. Para especificar el tipo de ordenación se puede incluir la cláusula `desc` para orden descendente o `asc` para orden ascendente.
- Se puede ordenar con respecto a más de un atributo.
- Como ordenar un gran número de tuplas puede ser costoso, es conveniente ordenar sólo cuando sea estrictamente necesario.

# SQL

## Operaciones sobre conjuntos

- Las operaciones union, intersect y except operan sobre relaciones y corresponden a las operaciones del álgebra relacional unión (U), intersección ( $\cap$ ) y diferencia de conjuntos (-).
- Las relaciones que participan en las operaciones han de ser compatibles (han de tener el mismo conjunto de atributos).
- A diferencia de la cláusula select, la operación union (unión) elimina duplicados automáticamente. Para conservar los duplicados se utiliza union all en lugar de union.
- La operación intersect (intersección) elimina duplicados automáticamente. Para conservar los duplicados se utiliza intersect all en lugar de intersect.
- La operación except (excepto) elimina duplicados automáticamente. Para conservar los duplicados se utiliza except all en lugar de except.

# SQL

## Funciones de agregación

- Las funciones de agregación son funciones que toman una colección (un conjunto o multiconjunto) de valores como entrada y producen un único valor como salida.
- SQL proporciona cinco funciones de agregación primitivas: avg (Media), min (Mínimo), max (Máximo), sum (Total) y count (Cuenta).
- La entrada a sum y avg debe ser una colección de números, pero los otros operadores pueden operar sobre colecciones de datos de tipo no numérico, tales como las cadenas.

# SQL

## Funciones de agregación

- Existen situaciones en las cuales sería deseable aplicar las funciones de agregación no sólo a un único conjunto de tuplas sino también a un grupo de conjuntos de tuplas; esto se especifica en SQL usando la cláusula group by.
- El atributo o atributos especificados en la cláusula group by se usan para formar grupos.
- Las tuplas con el mismo valor en todos los atributos especificados en la cláusula group by se colocan en un grupo.
- Hay operaciones, como por ejemplo calcular la media, en las que es importante la conservación de duplicados.
- Hay casos en los que se deben eliminar los duplicados antes de calcular una función de agregación. Para eliminar duplicados se utiliza la palabra clave distinct en la expresión de agregación.

# SQL

## Funciones de agregación

- A veces es más útil establecer una condición que se aplique a los grupos que una que se aplique a las tuplas. Para expresar este tipo de consultas, se utiliza la cláusula having.
- Los predicados de la cláusula having se aplican después de la formación de grupos, de modo que se pueden usar las funciones de agregación.
- A veces se desea tratar la relación entera como un único grupo. En casos de este tipo no se usa la cláusula group by.
- La función de agregación count se utiliza para contar el número de tuplas de una relación. La notación para esta función en SQL es count(\*) .
- SQL no permite el uso de distinct con count(\*). Si se permite el uso de distinct con max y min, incluso cuando el resultado no cambia.

# SQL

## Funciones de agregación

- Si en una misma consulta aparece una cláusula where y una cláusula having, se aplica primero el predicado de la cláusula where. Las tuplas que satisfagan el predicado de la cláusula where se colocan en grupos según la cláusula group by. La cláusula having, si existe, se aplica entonces a cada grupo; los grupos que no satisfagan el predicado de la cláusula having se eliminan. La cláusula select utiliza los grupos restantes para generar las tuplas resultado de la consulta.

# SQL

## Valores nulos

- SQL permite el uso de valores nulos para indicar la ausencia de información sobre el valor de un atributo.
- En un predicado se puede usar la palabra clave especial null para comprobar si un valor es nulo.
- El predicado is not null pregunta por la ausencia de un valor nulo.
- El uso de un valor nulo en las operaciones aritméticas y de comparación causa varias complicaciones. SQL maneja los valores nulos de la siguiente manera:
  - El resultado de una expresión aritmética es nulo si cualquiera de los valores de entrada es nulo.
  - SQL trata como desconocido el resultado de cualquier comparación que implique un valor nulo (aparte de is null e is not null).

# SQL

## Estructura básica

SELECT [{ALL|DISTINCT}]  
<nombre\_campo>[, <nombre\_campo>...]

FROM {<nombre\_tabla>|<nombre\_vista>}[,  
<nombre\_tabla>|<nombre\_vista>...]

[WHERE <condición> [{AND|OR} <condición>...]]

[GROUP BY <nombre\_campo>[, <nombre\_campo>...]]

[HAVING <condición> [{AND|OR} <condición>...]]

[ORDER BY {<nombre\_campo>|<indice\_campo>} [{ASC|DESC}][,  
<nombre\_campo>|<indice\_campo>} [{ASC|DESC}]]];

<https://es.wikipedia.org/wiki/SQL>

# SQL

## Estructura básica

SELECT [{ALL|DISTINCT}]  
<nombre\_campo>[, <nombre\_campo>...]

FROM {<nombre\_tabla>}|{<nombre\_vista>}[,  
<nombre\_tabla>|<nombre\_vista>...]

[WHERE <condición> [{AND|OR} <condición>...]]

[GROUP BY <nombre\_campo>[, <nombre\_campo>...]]

[HAVING <condición> [{AND|OR} <condición>...]]

[ORDER BY {<nombre\_campo>}|{<indice\_campo>} [{ASC|DESC}][,  
<nombre\_campo>|<indice\_campo>} [{ASC|DESC}]]];

<https://es.wikipedia.org/wiki/SQL>

# Bases de datos

Bases de datos  
(Ejercicios)

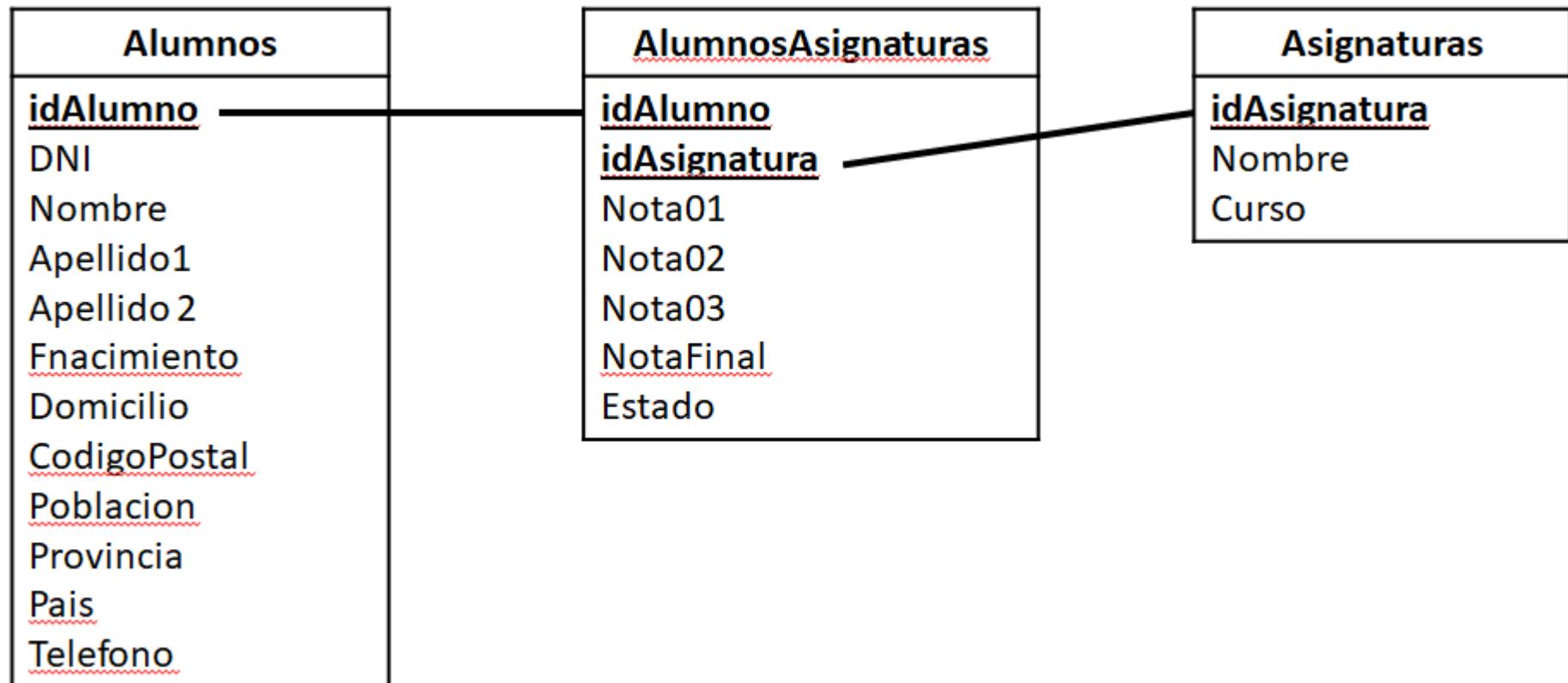
# SQL

## Ejercicios

# SQL

## Ejercicios

- Dada la siguiente base de datos expresada por su modelo relacional:



# SQL

## Ejemplo

- La siguiente tabla representa la relación Articulo:

Articulo

id-articulo	descripcion-articulo	categoria-articulo	precio-articulo
A0001	Placa base	Hardware	42.85
A0002	Microprocesador	Hardware	134.75
A0003	Impresora	Hardware	89.00
A0004	Disco SSD 256 GB	Hardware	65.47
A0005	Memoria RAM 16 GB	Hardware	18.54
A0006	Windows 11 Home 64 bits	Software	145.00
A0007	Autodesk AutoCad 2024	Software	220.00
A0008	Altavoces Pc	Hardware	55.95

Cada fila de la tabla representa una relación entre un conjunto de valores.  
Una tabla es el conjunto de dichas relaciones  
Las cabeceras de columna son los atributos.  
Para cada atributo hay un conjunto de valores, llamado dominio de ese atributo.

# SQL

## Ejemplo

id-cliente	nombre-cliente	Calle-cliente	Ciudad-cliente
19.283.746	González	Arenal	La Granja
01.928.374	Gómez	Carretas	Cerceda
67.789.981	López	Mayor	Peguerinos
18.273.609	Abril	Preciados	Valsaín
32.112.312	Santos	Mayor	Peguerinos
33.666.999	Rupérez	Ramblas	León
01.928.374	Gómez	Carretas	Cerceda

id-cliente	número-cuenta
19.283.746	C-101
19.283.746	C-201
01.938.374	C-215
67.789.901	C-102
18.273.609	C-305
32.112.312	C-217
33.666.999	C-222
01.928.374	C-201

número-cuenta	saldo
C-101	500
C-215	700
C-102	400
C-305	350
C-201	900
C-217	750
C-222	700

# Bases de datos

Fin