



Aspectos tecnológicos en Proyectos de Esports

Clase 4

Clase 4

Introducción a las Bases de Datos

Conceptos Generales
Bases de Datos relacionales
Bases de Datos NoSQL

01

BD relacionales

Etapas de diseño
Modelo ER
Modelo relacional

02

Lenguaje de consultas SQL

Lenguaje de definición de datos (DDL)
Lenguaje de manipulación de datos (DML)

03

Aspectos tecnológicos en
Proyectos de Esports

Clase 4

Introducción a las Bases de Datos

Conceptos Generales
Bases de Datos relacionales
Bases de Datos NoSQL

01

BD relacionales

Etapas de diseño
Modelo ER
Modelo relacional

02

Lenguaje de consultas SQL

Lenguaje de definición de datos (DDL)
Lenguaje de manipulación de datos (DML)

03

Aspectos tecnológicos en
Proyectos de Esports

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk. They are looking at a large monitor that displays a complex, glowing blue and white network or data visualization. The person is wearing a dark shirt and a headset with a microphone.

Conceptos Generales

Una **Base de Datos (BD)** se puede definir como cualquier información dispuesta de manera adecuada para su tratamiento por una computadora.

Más precisamente, una base de datos es un **conjunto de datos interrelacionados** que pertenecen a un mismo contexto y que se utiliza para administrar de forma digital grandes volúmenes de información.

Una base de datos se diseña, construye y completa con datos para un **propósito específico**, y esta destinada a un grupo de aplicaciones o usuarios concretos.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor. The monitor displays a complex, colorful data visualization, possibly a network or a game map. The person's hands are on the keyboard.

Conceptos Generales

Las bases de datos son la forma más común de almacenar y manipular datos. En el proceso de creación de una base de datos existen **dos procesos esenciales**:

- La **selección de la información a representar** del dominio de un problema específico → realización del proceso de abstracción
- El **diseño de un esquema de representación** para esos datos que cumpla con ciertas características deseables

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk. They are looking at a large monitor that displays a complex, colorful data visualization or network diagram. The person is wearing a dark shirt and a headset with a microphone.

Conceptos Generales

Un **Sistema de Gestión de Bases de Datos (SGDB)** es un sistema de software que permite a los usuarios crear, mantener y utilizar la BD.

El SGDB facilita los procesos que se realizan sobre la BD:

- **Definición:** especificar los tipos de datos, las estructuras y restricciones entre los datos.
- **Construcción:** almacenar datos concretos.
- **Manipulación:** consultar la BD para recuperar datos específicos, realizar operaciones de altas, bajas y modificaciones.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor. The monitor displays a game with a blue and white interface. The person is wearing a dark shirt and is sitting in a black office chair.

Conceptos Generales

Los **objetivos de un SGBD** son:

- Control de redundancia.
- Acceso a los datos en todo momento.
- Acceso concurrente a los datos.
- Seguridad: control de acceso a datos, usuarios, recursos, backups, entre otros.
- Integridad: persistencia de datos aún ante fallos, restricciones de datos, entre otros.

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk. They are looking at a large monitor that displays a complex, colorful data visualization or network diagram. The person is wearing a dark shirt and a headset with a microphone.

Bases de Datos Relacionales

Desde hace mucho tiempo, las **bases de datos relacionales** son las más utilizadas. Este tipo de base de datos permite almacenar conjuntos de datos en tablas y establecer relaciones entre ellos.

El **modelo relacional** es simple, potente y formal. Representa a la BD como una colección de archivos (las tablas) con ciertas características. Cada tabla tiene un conjunto de atributos, en donde cada atributo tiene un dominio establecido.

El esquema final de la BD queda definido por la colección de los esquemas de las tablas.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and is looking at a large monitor. The monitor displays a game with a blue and white interface. The person is wearing a dark shirt and is sitting in a black office chair.

Bases de Datos Relacionales

En cada tabla se define una **clave primaria**, que es un atributo o grupo de atributos que identifican de forma unívoca a cada fila (tupla) de la tabla.

El modelo relacional permite además plantear **restricciones entre tablas**. Esta es la propiedad denominada **integridad referencial**, y sirve para mantener la consistencia de los datos almacenados en la base de datos.

Mediante la integridad referencial es posible definir reglas que restringen las operaciones que podrían generar conflicto.



Bases de Datos Relacionales

En los últimos años, **el uso de las bases de datos se ha extendido de los servidores hacia los dispositivos móviles**. Eso ha generado la creación de diversos métodos de almacenamiento de información en dispositivos móviles, embebidos y empotrados.

En el año 2000 se crea **SQLite**, que es una herramienta de **software libre** que permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser un teléfono celular.

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk. They are looking at a large monitor that displays a complex, colorful data visualization or network diagram. The person is wearing a dark shirt and a headset with a microphone.

Bases de Datos Relacionales

SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje **SQL**, y es posible usarlo tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos

Esto se debe a que existe una **total compatibilidad entre las diversas plataformas disponibles**, haciendo que la **portabilidad** entre dispositivos y plataformas sea transparente.

La **base de datos completa se encuentra en un solo archivo**. Además, es totalmente autocontenida (sin dependencias externas) y puede funcionar enteramente en memoria, lo que la hace **muy rápida**.

Bases de Datos Relacionales

Las BD relacionales más utilizadas según el sitio DB-engines son:

Rank May 2024	DBMS	Database Model	Score May 2024
1.	Oracle +	Relational, Multi-model i	1236.29
2.	MySQL +	Relational, Multi-model i	1083.74
3.	Microsoft SQL Server +	Relational, Multi-model i	824.29
4.	PostgreSQL +	Relational, Multi-model i	645.54
5.	IBM Db2	Relational, Multi-model i	128.46
6.	Snowflake +	Relational	121.33
7.	SQLite +	Relational	114.32

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk with multiple computer monitors. The background is a blurred office or server room environment.

Bases de Datos NoSQL

Desde hace ya varios años han aparecido nuevas aplicaciones que **demandan nuevas características** a las tecnologías de BD, tales como un alto grado de concurrencia en las operaciones, baja latencia, mayor escalabilidad y disponibilidad, entre otros. Además, en muchos casos, la información a ser almacenada **no tiene una estructura fija**.

Estos nuevos tipos de aplicaciones generan un gran volumen de datos. Por ejemplo, según **Internet Live Stats**, en solo 1 segundo:

- Se generan más de 6000 tweets.
- Se suben más de 800 fotos en Instagram.
- Se realizan más de 70000 búsquedas en Google.

A person is shown from the side, sitting at a desk and working on a computer. The computer monitor displays a complex, colorful visualization of data, possibly a network or a game map. The person is wearing a headset and a dark shirt. The background is slightly blurred, showing other parts of the workspace.

Bases de Datos NoSQL

Toda esta gran cantidad de información tiene que poder ser almacenada y consultada con tiempos de respuesta relativamente cortos. Aquí es en donde comienzan a mostrar dificultades las bases de datos relacionales y aparecen las **bases de datos NoSQL (Not only SQL)**.

Las bases de datos NoSQL son un enfoque para la gestión de datos que es útil para **conjuntos muy grandes de datos distribuidos**. Abarca una amplia gama de tecnologías y arquitecturas y busca resolver la **escalabilidad** y los problemas de **rendimiento de big data** que las bases de datos relacionales no fueron diseñadas para resolver.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor. The monitor displays a complex, colorful data visualization, possibly a network diagram or a game map. The person's hands are on the keyboard.

Bases de Datos NoSQL

Características principales:

- **Esquema no estructurado.** No es necesario diseñar un esquema estructurado para poder almacenar los datos.
- **Escalabilidad.** Facilita la adición o reducción de la capacidad de forma rápida.
- **Rendimiento.** Es posible aumentar el rendimiento agregando más servidores básicos, de forma confiable y rápida.
- **Alta disponibilidad.** Usan arquitecturas para disponer de los datos incluso ante fallos.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor that displays a complex, colorful data visualization or network diagram. The image is partially cut off on the left side.

Bases de Datos NoSQL

Tipos de BD NoSQL:

- **Clave-valor.** Es el tipo más simple, se puede acceder a un determinado valor a partir de una clave única. Es un tipo de almacenamiento no estructurado que tiene un gran rendimiento y se puede escalar fácilmente.
- **Documental.** Son similares a las de clave-valor, la diferencia radica en que el valor contiene datos estructurados o semiestructurados. Este valor se conoce como un documento y puede estar en formato XML, JSON o BSON, entre otros.







A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk with multiple computer monitors. The background of the image is a blurred, high-tech environment with blue and white light effects.

Bases de Datos NoSQL

- **Familia de columnas.** Los datos se almacenan en celdas agrupadas en columnas de datos (en lugar de filas de datos, como en una tabla relacional). Las columnas se agrupan lógicamente en “familias de columnas” representando un grupo de datos similares a los que normalmente se accede de forma conjunta.
- **Basadas en grafos.** Se basa en nodos con propiedades. Es una forma muy flexible de describir cómo se relacionan unos datos con otros. La utilidad de este modelo se da cuando los datos que se almacenan presentan muchas interrelaciones entre sí, y es esto último lo que se pondera más que los propios datos. Se almacena más información sobre las relaciones de los datos que de los datos en sí.

Bases de Datos Relacionales

Las BD NoSQL de tipo clave-valor más utilizadas según el sitio DB-engines son:

Rank May 2024	DBMS	Database Model	Score May 2024
1.	Redis 	Key-value , Multi-model 	157.80
2.	Amazon DynamoDB 	Multi-model 	74.07
3.	Microsoft Azure Cosmos DB 	Multi-model 	29.04
4.	Memcached	Key-value	19.42
5.	etcd	Key-value	7.25

Bases de Datos Relacionales

Las BD NoSQL de tipo documental más utilizadas según el sitio DB-engines son:

Rank May 2024	DBMS	Database Model	Score May 2024
1.	MongoDB +	Document , Multi-model i	421.65
2.	Databricks +	Multi-model i	78.61
3.	Amazon DynamoDB +	Multi-model i	74.07
4.	Microsoft Azure Cosmos DB +	Multi-model i	29.04
5.	Couchbase +	Document , Multi-model i	17.30
6.	Firebase Realtime Database	Document	14.29
7.	CouchDB	Document , Multi-model i	9.30

Clase 4

Introducción a las Bases de Datos

Conceptos Generales
Bases de Datos relacionales
Bases de Datos NoSQL

01

BD relacionales

Etapas de diseño
Modelo ER
Modelo relacional

02

Lenguaje de consultas SQL

Lenguaje de definición de datos (DDL)
Lenguaje de manipulación de datos (DML)

03

Aspectos tecnológicos en
Proyectos de Esports



Etapas de diseño

- **Diseño Conceptual:** representación abstracta de los datos.
 - **Propósito:** describir el contenido de información de la BD, captando y representando las necesidades del usuario.
 - **Entrada:** documento de especificación de requerimientos.
 - **Salida:** esquema conceptual de la BD → descripción de alto nivel de la estructura de la BD.
 - **Lenguaje:** modelo ER.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor. The monitor displays a game with a blue and white interface. The person is wearing a dark shirt and is sitting in a black office chair.

Etapas de diseño

- **Diseño Lógico:** representación de los datos en una computadora.
 - **Propósito:** describir el contenido de información de la BD, para que sea procesada por un SGBD relacional.
 - **Entrada:** esquema conceptual de la BD.
 - **Salida:** esquema lógico de la BD
 - **Lenguaje:** modelo ER.



Etapas de diseño

- **Diseño Físico:** determinación de estructuras de almacenamiento.
 - **Propósito:** describir las estructuras de almacenamiento y métodos de acceso utilizados para tener un acceso efectivo a los datos.
 - **Entrada:** esquema lógico de la BD.
 - **Salida:** esquema físico de la BD → script SQL.
 - **Lenguaje:** sintaxis del SGBD elegido.



Modelo ER

El **Modelo Entidad Relación (ER)** tiene tres elementos básicos:

- **Entidades:** clases de objetos del mundo real con identidad.
- **Relaciones:** agregaciones de entidades. Tienen cardinalidad.
- **Atributos:** propiedades básicas de las entidades y relaciones. Tienen un dominio y cardinalidad. Pueden ser simples o compuestos.



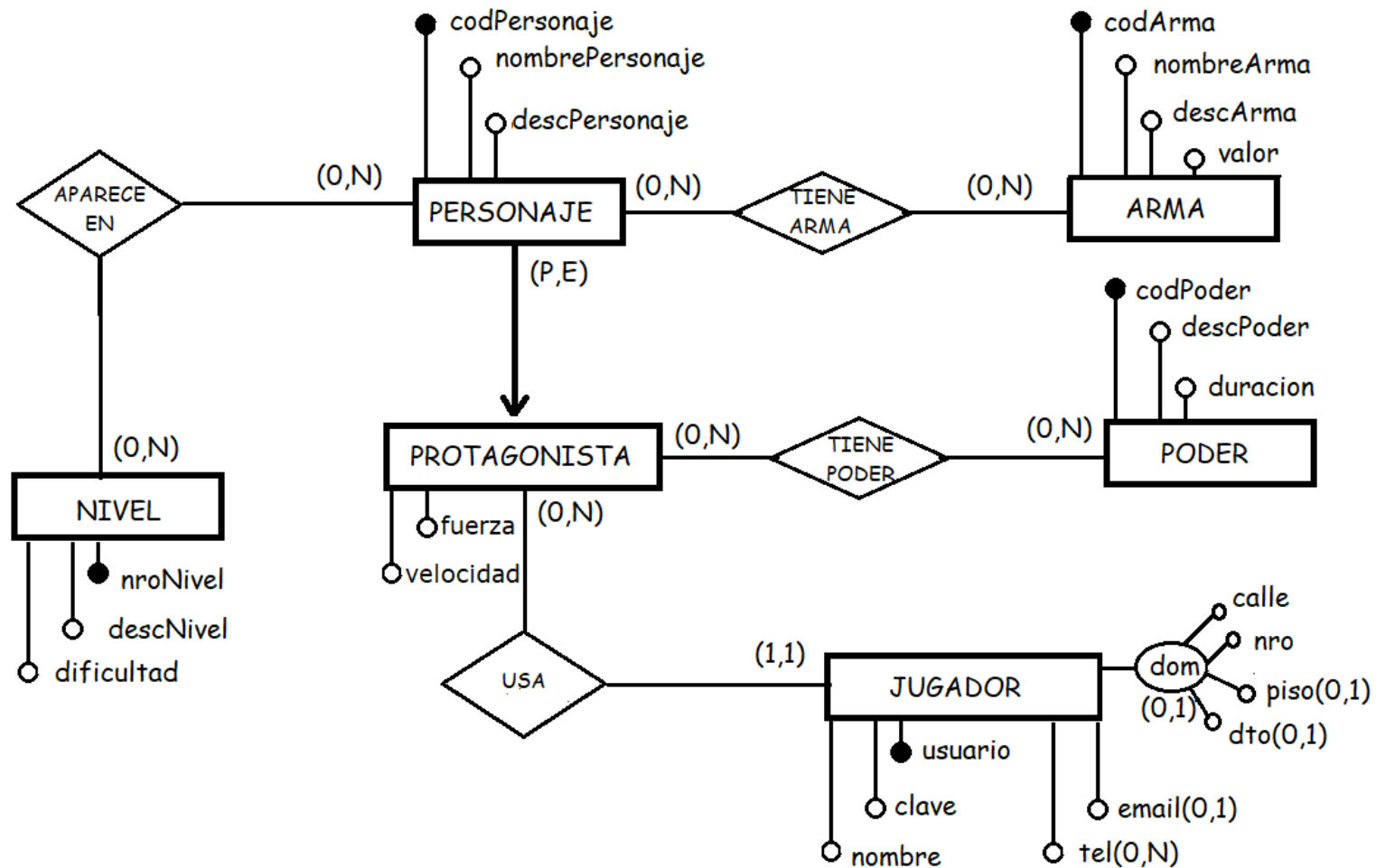
Modelo ER

Otros elementos importantes del modelo ER:

- **Identificadores:** atributo o conjunto de atributos que permite distinguir a una entidad de manera unívoca dentro del conjunto de entidades.
- **Jerarquías de generalización**
 - En la clase genérica se definen las propiedades en común, que heredan todas las clases especiales.
 - Se define la cobertura: total o parcial, exclusiva o superpuesta.

Modelo Relacional

Ej: esquema conceptual sencillo



A person is shown from the side, sitting at a desk and working on a computer. The monitor displays a game with a large, stylized character. The person is wearing a dark shirt and a headset. The background is slightly blurred, showing a desk with various items.

Modelo ER

El **esquema conceptual** debe ser adaptado al **modelo relacional**, generando así un **esquema lógico** relacional.

Al hacer esta conversión, se deben tener en cuenta varios aspectos:

- Restricciones del modelo relacional.
- Criterios de rendimiento.
- Información de carga de la BD.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor. The monitor displays a game with a blue and white interface. The person is wearing a dark shirt and a headset with a microphone.

Modelo ER

Las restricciones del modelo relacional hacen necesarias las siguientes tareas:

- Resolución de datos derivados y ciclos de relaciones.
- Conversión de atributos polivalentes.
- Conversión de atributos compuestos.
- Conversión de jerarquías de generalización.
- Partición y/o fusión de entidades/relaciones (tareas optativas).



Modelo ER

- **Datos derivados**

Son datos que están **repetidos** en el esquema, por lo que pueden ser obtenidos de más de una forma.

Se puede decidir conservarlos si resultan en una **mejora de la performance** de la BD. En este caso, se debe documentar.

Como desventaja, **requiere más espacio de almacenamiento y procesamiento adicional** para mantener la consistencia de los datos guardados.

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk. A large monitor displays a complex, colorful data visualization, possibly a network or sports performance graph. The person is wearing a dark shirt and is focused on the screen.

Modelo ER

- Ciclos de relaciones

En el caso en que un ciclo genere una repetición de datos, se debe decidir si se conserva o no.

Se puede decidir conservar un ciclo que genera repetición de información si resulta en una **mejora de la performance** de la BD. En este caso, se debe documentar.

Como desventaja, requiere **más espacio de almacenamiento** y **procesamiento adicional** para mantener la consistencia de los datos guardados.

A person is shown from the side, sitting at a desk and working on a computer. The monitor displays a game with a blue and white interface. The person is wearing a dark shirt and a headset. The background is slightly blurred, showing a desk with various items.

Modelo ER

- **Conversión de atributos polivalentes**

Los SGBD relacionales **no permiten atributos con múltiples valores** de una dimensión determinada dinámicamente.

Se debe **quitar el atributo de la entidad y generar una nueva entidad** con ese atributo y establecer una nueva relación con la entidad a la cuál pertenecía.



Modelo ER

- **Conversión de atributos compuestos**

Existen 3 alternativas de resolución:

- Considerar el atributo compuesto entero como **un solo atributo**
- Considerar todos sus componentes como **atributos individuales**
- Generar una **nueva entidad**, la que representa el atributo compuesto, conformada por cada uno de los atributos simples que contiene



Modelo ER

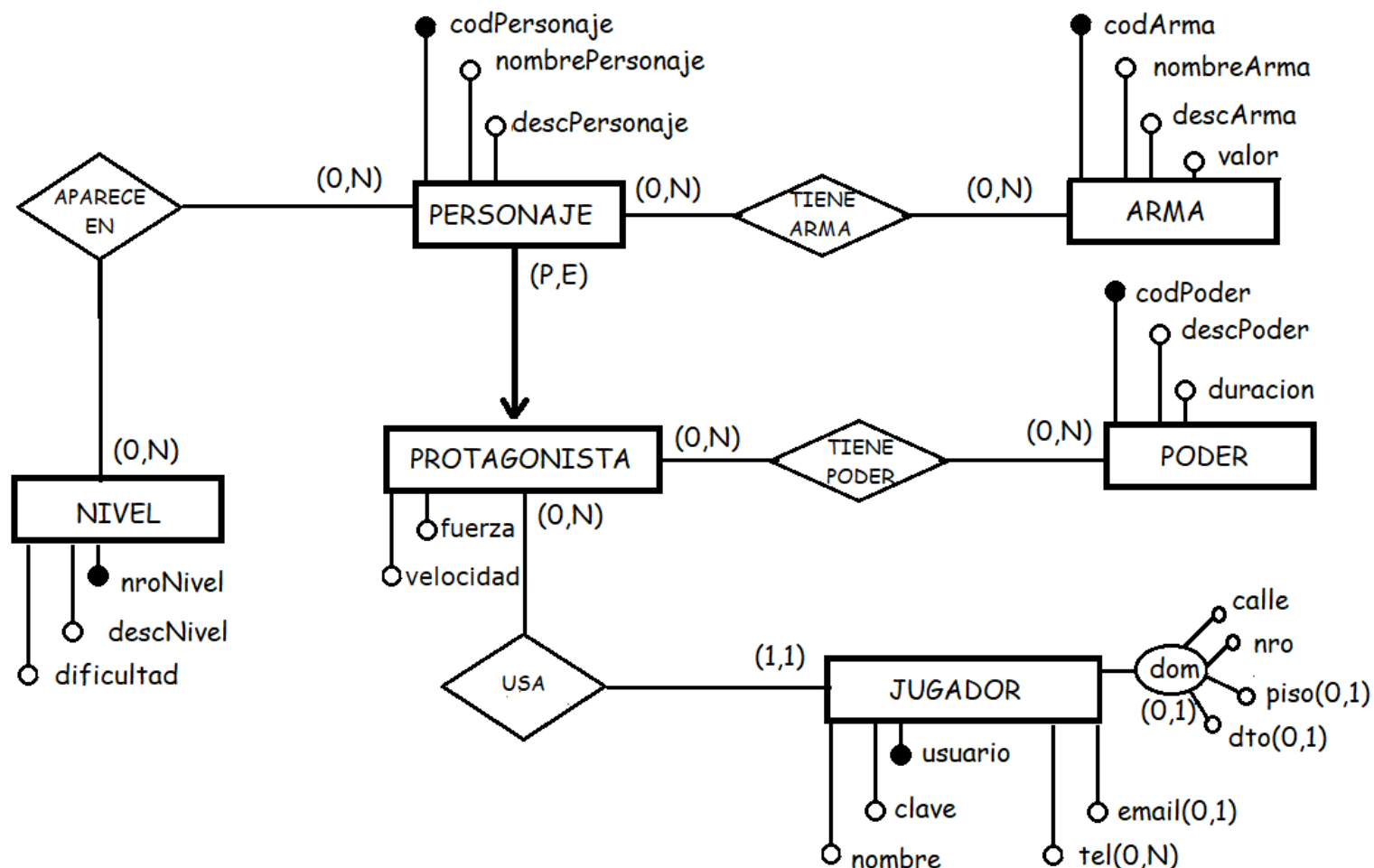
- **Conversión de jerarquías de generalización**

Existen 3 alternativas de resolución:

- **Integrar la jerarquía de generalización.** La superentidad contendrá sus atributos y también los de todas las subentidades. Generan valores nulos de atributos.
- **Eliminar la superentidad, pero retener las subentidades.** Genera repetición de atributos y relaciones de la superentidad. Inadecuado para cobertura superpuesta o parcial.
- **Retener todas las entidades.** Se establecen explícitamente las interrelaciones entre las superentidades y subentidades. Es el caso más general y aplicable.

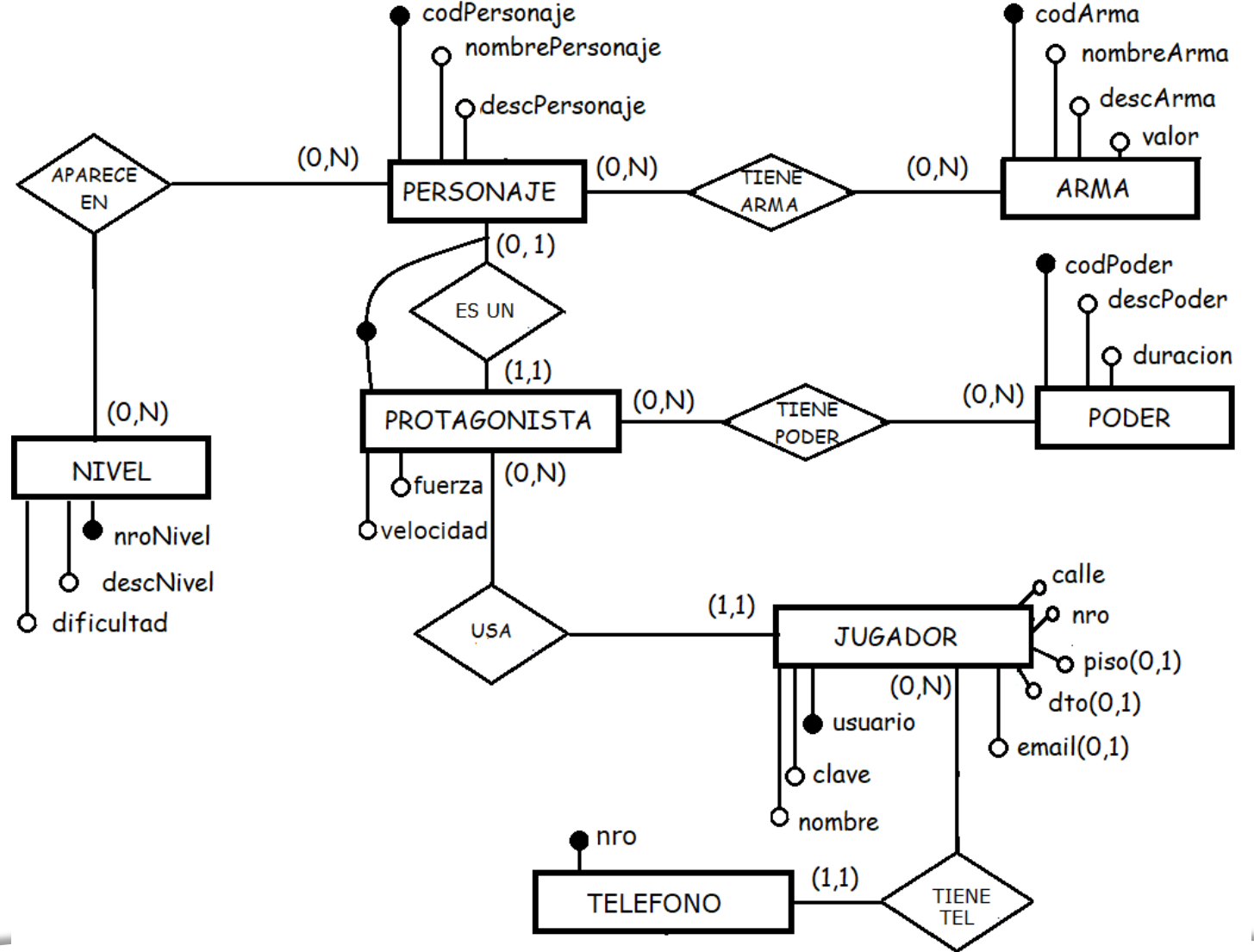
Modelo Relacional

Ej: esquema conceptual



Modelo Relacional

Ej: conversión al
esquema lógico



A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor that displays a complex, colorful data visualization or network diagram. The image is partially cut off on the left edge.

Modelo Relacional

El **Modelo Relacional** es ampliamente difundido y utilizado desde 1970. Es simple, potente y formal.

Representa a la BD como una colección de archivos, que llama tablas (o relaciones). Una tabla contiene los datos en filas y columnas:

- Filas de la tabla → **tuplas**
- Columnas de la tabla → **atributos**

Una tabla **no puede tener tuplas duplicadas**.

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk. The desk has a computer monitor displaying a network diagram with various nodes and connections. The person is wearing a dark shirt and a headset with a microphone.

Modelo Relacional

Se debe llevar a cabo una **conversión del esquema lógico al esquema físico**, para que dicho esquema pueda ser procesado por una computadora.

Esta conversión incluye las siguientes tareas:

- Eliminación de identificadores externos.
- Selección de claves (candidatas, primarias, secundarias, foráneas).
- Conversión de entidades.
- Conversión de relaciones.



Modelo Relacional

- **Eliminación de identificadores externos**

Una entidad debe incorporar aquellos atributos externos que son necesarios para la identificación.

Si una entidad tiene más de un identificador, se debe elegir el más adecuado/representativo → **tipos de claves**



Modelo Relacional

- **Tipos de claves**

- **Clave Candidata.** Una tabla puede tener una o más claves que identifican sus tuplas → cada una se denomina clave candidata.
- **Clave Primaria.** Es la **clave candidata elegida** para identificar las tuplas de la tabla. Se subraya en el esquema de la tabla.
- **Clave Foránea.** Es un atributo o grupo de atributos de una tabla, que es clave primaria en otra tabla. Sirven para plantear restricciones de **integridad referencial**.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor. The monitor displays a game with a blue and white interface. The person is wearing a dark shirt and a headset.

Modelo Relacional

- **Conversión de entidades**

Salvo casos excepcionales, cada una de las entidades definidas se convierte en una tabla.

La clave primaria se indica subrayada.

Es posible agregar campos para **índices autoincrementales**.

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk with multiple computer monitors. One monitor displays a game, and the other shows some data or code. The person is wearing a dark shirt and a headset with a microphone.

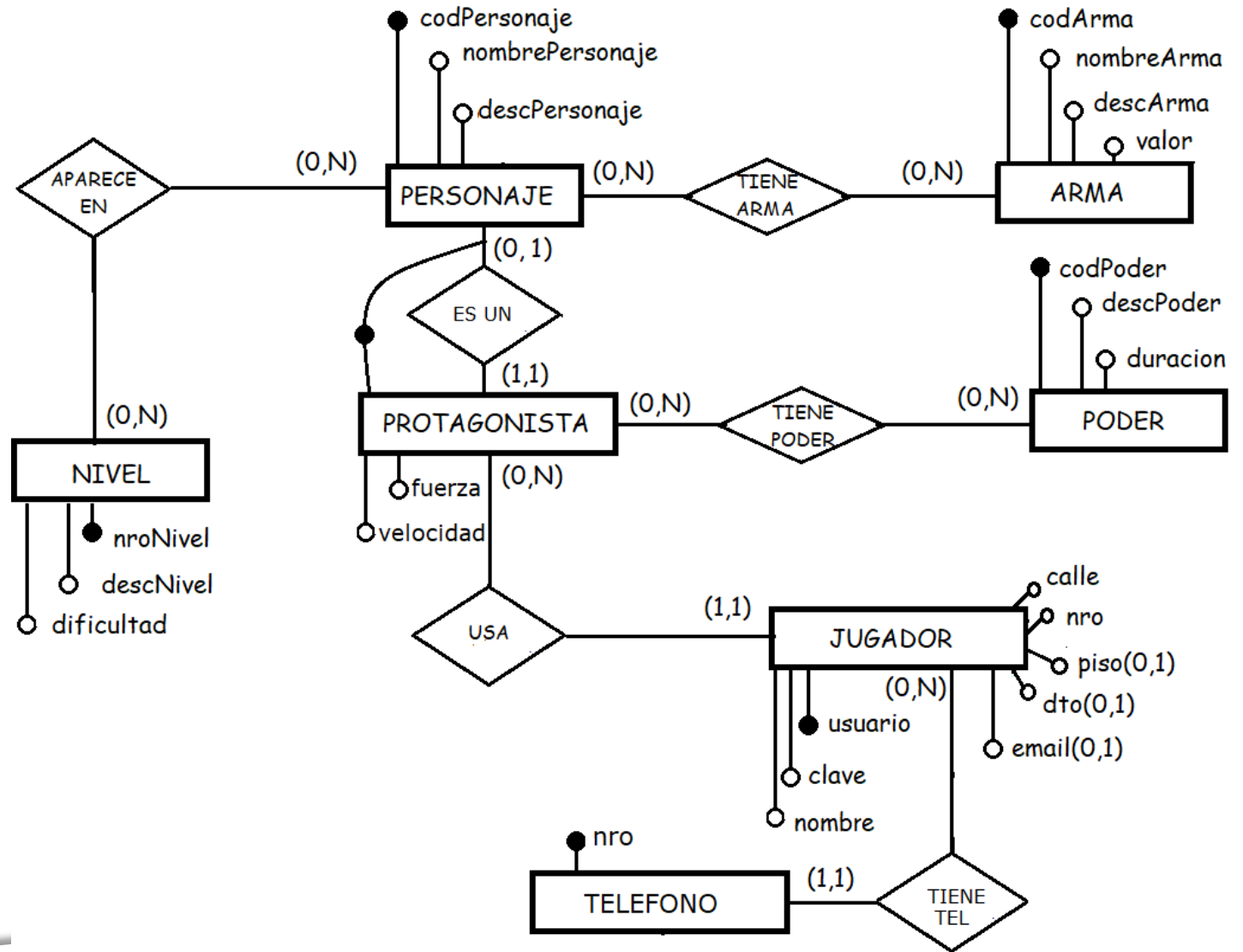
Modelo Relacional

- **Conversión de relaciones**

- **Relaciones Uno-Uno.** Se lleva la clave a la entidad que tenga participación total. En caso de que ambas sean con participación parcial, se puede generar una nueva tabla.
- **Relaciones Uno-Muchos.** Se lleva la clave a la entidad con cardinalidad 1. En caso de que dicha entidad tenga participación parcial, se genera una nueva tabla.
- **Relaciones Muchos-Muchos.** En todos los casos se genera una nueva tabla, en donde la clave primaria será la suma de los atributos clave de ambas tablas.

Modelo Relacional

Ej: esquema
lógico





Modelo Relacional

Ej: conversión al esquema físico

- **PERSONAJE** (codPersonaje, nombrePersonaje, descPersonaje)
- **ARMA** (codArma, nombreArma, descArma, valor)
- **TIENE_ARMA** (codArma, codPersonaje)
- **PROTAGONISTA** (codPersonaje, fuerza, velocidad)
- **PODER** (codPoder, descPoder, duracion)
- **TIENE_PODER** (codPoder, codPersonaje)
- **NIVEL** (nroNivel, descNivel, dificultad)
- **APARECE_EN** (nroNivel, codPersonaje)
- **JUGADOR** (usuario, clave, nombre, email, calle, nro, piso, dto, codPersonaje)
- **TELEFONO** (nro, usuario)

Clase 4

Introducción a las Bases de Datos

Conceptos Generales
Bases de Datos relacionales
Bases de Datos NoSQL

01

BD relacionales

Etapas de diseño
Modelo ER
Modelo relacional

02

Lenguaje de consultas SQL

Lenguaje de definición de datos (DDL)
Lenguaje de manipulación de datos (DML)

03

Aspectos tecnológicos en
Proyectos de Esports

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor that displays a complex, colorful data visualization or network diagram. The image is partially cut off on the left edge.

Lenguaje de definición de datos (DDL)

El **DDL** permite la **definición y modificación** de bases de datos, sus esquemas, relaciones, índices, vistas, entre otros.

También posibilita la definición de **permisos** y autorizaciones de acceso a los datos.

Mediante este lenguaje, además, es posible definir **reglas de integridad**, de forma de mantener la consistencia de los datos, y de controlar la concurrencia para permitir el acceso a los datos en todo momento.



Lenguaje de definición de datos (DDL)

Operaciones básicas: **CREATE**, **DROP**, **ALTER**.

→ **CREATE DATABASE** nombreBD;

→ **DROP DATABASE** nombreBD;

Lenguaje de definición de datos (DDL)

→ **CREATE TABLE** empresa (
 idEmpresa INTEGER NOT NULL AUTO_INCREMENT,
 nombreEmpresa VARCHAR(100) NOT NULL,
 cuit VARCHAR(13) NOT NULL,
 domicilio VARCHAR(50) NULL,
 ...,
 PRIMARY KEY(idEmpresa)
);

Lenguaje de definición de datos (DDL)

→ **CREATE TABLE** jugador (
 idJugador INTEGER NOT NULL AUTO_INCREMENT,
 idEmpresa INTEGER UNSIGNED NOT NULL,
 fechaDesde DATE NOT NULL,
 fechaHasta DATE NULL,
 nombreJugador VARCHAR(100) NOT NULL,
 ...,
 PRIMARY KEY(idJugador)
 FOREIGN KEY(idEmpresa) REFERENCES empresa (idEmpresa)
 ON DELETE RESTRICT
);



Lenguaje de definición de datos (DDL)

→ **DROP TABLE** nombreTabla;

→ **ALTER TABLE** empresa (
 ADD COLUMN razonSocial VARCHAR(100) NOT NULL,
 DROP COLUMN nombreEmpresa,
 ALTER COLUMN domicilio VARCHAR(80) NOT NULL,
 ...,
);

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk with multiple computer monitors. The background of the image is a blurred, high-tech environment with blue and white light effects.

Lenguaje de manipulación de datos (DML)

El **DML** permite la **consulta y modificación del contenido** de la BD, es decir, de los datos almacenados.

La estructura básica de una consulta de datos utiliza 3 cláusulas:

- **SELECT.** Permite **seleccionar los atributos** que se desea mostrar en el resultado (la cláusula **DISTINCT** elimina tuplas duplicadas)
- **FROM.** Permite realizar un **producto cartesiano** entre las tablas que se incluyan.
- **WHERE.** En esta cláusula se debe **especificar las condiciones** de cruce entre las tablas incluidas en el FROM, y cualquier otra condición de filtrado.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and looking at a large monitor. The monitor displays a game with a blue and white interface. The person is wearing a dark shirt and is focused on their work.

Lenguaje de manipulación de datos (DML)

La cláusula **AS** permite **renombrar tablas** o **atributos**.

Así, cuando se usan muchas tablas en el FROM, es posible darles un nombre sencillo para luego referenciarlas.

Además, permite realizar cruces (productos) de una tabla consigo misma.

Otra aplicación del AS es para dar un nombre diferente a los resultados proyectados en el SELECT.

A vertical image on the left side of the slide shows a person sitting at a desk, working on a computer. The person is wearing a headset and is looking at a large monitor that displays a complex, colorful data visualization or network diagram. The person's hands are on the keyboard.

Lenguaje de manipulación de datos (DML)

Los resultados proyectados no tienen un orden establecido. Es decir, el resultado se muestra respetando el orden en el que los datos están almacenados en las tablas.

Si se desea mostrarlos en algún orden determinado, se debe utilizar una cláusula especificándolo explícitamente:

- **ORDER BY.** Permite especificar los atributos por los cuáles serán ordenadas las tuplas del resultado. Es posible indicar si el orden debe ser descendente (**DESC**) o ascendente (**ASC**, valor por defecto).

Esta cláusula permite especificar tantos criterios de orden como sean necesarios.

A vertical image on the left side of the slide shows a person in a dark shirt sitting at a desk, working on a computer. The computer monitor displays a complex, colorful data visualization, possibly a network diagram or a game interface. The person's hands are on the keyboard, and they are looking at the screen.

Lenguaje de manipulación de datos (DML)

Funciones de agregación:

- **AVG** (promedio): aplicable a atributos numéricos, retorna el promedio de la cuenta
- **MIN** (mínimo): retorna el menor elemento no nulo dentro de las tuplas para ese atributo
- **MAX** (máximo): retorna el mayor elemento no nulo dentro de las tuplas para ese atributo
- **SUM** (total): realiza la suma; aplicable a atributos numéricos.
- **COUNT** (contador): cuenta las tuplas resultantes.

A person is seated at a desk, working on a computer. The monitor displays a complex data visualization, possibly a network or a map. The person is wearing a headset and is focused on the screen. The background is slightly blurred, showing a typical office or lab environment.

Lenguaje de manipulación de datos (DML)

También es posible realizar agrupamientos de datos mediante las cláusulas GROUP BY y HAVING.

- **GROUP BY.** Permite agrupar un conjunto de tuplas por algún criterio.

Los atributos proyectados en el resultado (que no usan funciones de agregación) deben ser usados como criterio de agrupación.

Se aplica una función de agregación sobre cada grupo

- **HAVING.** Sirve para aplicar condiciones de grupo, es decir que actúa como filtro para los grupos (diferente del WHERE, que actúa sobre cada tupla)

A vertical image on the left side of the slide shows a person in a dark shirt sitting at a desk, working on a computer. The computer monitor displays a complex, colorful visualization, possibly a network diagram or a data map, with many nodes and connections. The person is looking at the screen, and their hands are near the keyboard.

Lenguaje de manipulación de datos (DML)

Otras operaciones:

- **UNION [ALL]**. Agrupa las tuplas de dos subconsultas.
- **INTERSECT**. Intersecta las tuplas de dos subconsultas.
- **EXCEPT**. Realiza la diferencia entre dos subconsultas.
- **IN**. Permite consultar si un elemento es parte o no de un conjunto de tuplas.
- **EXIST**. Devuelve verdadero si la subconsulta argumento no es vacía, y falso en caso contrario.



Lenguaje de manipulación de datos (DML)

Vamos a realizar algunos ejemplos para ver el uso de todas estas operaciones. Nos basaremos en el siguiente esquema de tablas:

Personaje (codPersonaje, nombre, descripcion)

Arma (codArma, nombre, descripcion, valor)

TieneArma (codArma, codPersonaje)

Nivel (nroNivel, descripcion, dificultad)

ApareceEn (nroNivel, codPersonaje)



Lenguaje de manipulación de datos (DML)

Personaje (codPersonaje, nombre, descripcion)

Arma (codArma, nombre, descripcion, valor)

TieneArma (codArma, codPersonaje)

Nivel (nroNivel, descripcion, dificultad)

ApareceEn (nroNivel, codPersonaje)

Ej1: armas que tienen un valor superior a 1000.

```
SELECT *
```

```
FROM Arma
```

```
WHERE valor > 1000
```



Lenguaje de manipulación de datos (DML)

Personaje (codPersonaje, nombre, descripcion)

Arma (codArma, nombre, descripcion, valor)

TieneArma (codArma, codPersonaje)

Nivel (nroNivel, descripcion, dificultad)

ApareceEn (nroNivel, codPersonaje)

Ej2: nombre de los personajes que utilizan el arma con nombre “Hacha”.

```
SELECT P.nombre  
FROM Personaje P, Arma A, TieneArma TA  
WHERE P.codPersonaje = TA.codPersonaje  
AND TA.codArma = A.codArma  
AND A.nombre = “Hacha”
```



Lenguaje de manipulación de datos (DML)

Personaje (codPersonaje, nombre, descripcion)

Arma (codArma, nombre, descripcion, valor)

TieneArma (codArma, codPersonaje)

Nivel (nroNivel, descripcion, dificultad)

ApareceEn (nroNivel, codPersonaje)

Ej3: listar cada personaje con las armas que utilizan, ordenado por nombre del personaje y nombre del arma.

```
SELECT P.nombre, A.nombre  
FROM Personaje P, Arma A, TieneArma TA  
WHERE P.codPersonaje = TA.codPersonaje  
AND TA.codArma = A.codArma  
ORDER BY P.nombre ASC, A.nombre ASC
```

Lenguaje de manipulación de datos (DML)

Personaje (codPersonaje, nombre, descripcion)

Arma (codArma, nombre, descripcion, valor)

TieneArma (codArma, codPersonaje)

Nivel (nroNivel, descripcion, dificultad)

ApareceEn (nroNivel, codPersonaje)

Ej4: valor promedio de las armas del personaje “Minotauro”.

```
SELECT AVG(valor) as valorPromedio  
FROM Personaje P, Arma A, TieneArma TA  
WHERE P.codPersonaje = TA.codPersonaje  
AND TA.codArma = A.codArma  
AND P.nombre = “Minotauro”
```



Lenguaje de manipulación de datos (DML)

Personaje (codPersonaje, nombre, descripcion)

Arma (codArma, nombre, descripcion, valor)

TieneArma (codArma, codPersonaje)

Nivel (nroNivel, descripcion, dificultad)

ApareceEn (nroNivel, codPersonaje)

Ej5: cantidad de niveles en los que aparecen los personajes, ordenado por cantidad de niveles de forma descendente.

```
SELECT P.nombre, COUNT(AE.nroNivel) as cantNiveles
FROM Personaje P, ApareceEn AE
WHERE P.codPersonaje = AE.codPersonaje
GROUP BY P.nombre
ORDER BY COUNT(AE.nroNivel) DESC
```




Lenguaje de manipulación de datos (DML)

Personaje (codPersonaje, nombre, descripcion)

Arma (codArma, nombre, descripcion, valor)

TieneArma (codArma, codPersonaje)

Nivel (nroNivel, descripcion, dificultad)

ApareceEn (nroNivel, codPersonaje)

Ej6: nombre de aquellos personajes que no aparecen en el nivel 1.

```
SELECT P.nombre
```

```
FROM Personaje P, ApareceEn AE
```

```
WHERE P.codPersonaje = AE.codPersonaje AND codPersonaje NOT IN
```

```
  (SELECT codPersonaje
```

```
    FROM ApareceEn AE2
```

```
    WHERE nroNivel = 1 AND AE.codPersonaje = AE2.codPersonaje)
```



Lenguaje de manipulación de datos (DML)

Para realizar modificaciones en el contenido de la BD se utilizan principalmente 3 cláusulas:

- **INSERT.** Permite insertar nuevas tuplas en tablas de la BD.
- **UPDATE.** Permite actualizar una o más tuplas de una tabla de la BD.
- **DELETE.** Permite eliminar una o más tuplas de una tabla de la BD.



Lenguaje de manipulación de datos (DML)

INSERT INTO nombreTabla (<nombreAtributo>,...) **VALUES** (<valorAtributo>,...)

Arma (codArma, nombre, descripcion, valor)

Ej: agregar un arma, cuando se tienen todos sus datos.

INSERT INTO Arma

VALUES (52, "Escopeta", "Escopeta corta a perdigones", 550)

También es posible dar valor sólo a algunos atributos (si el esquema de la tabla lo permite). Hay que indicar los nombres de las columnas que se van a completar.

Ej: agregar un arma, sin disponer de todos sus datos.

INSERT INTO Arma (codArma, nombre, valor)

VALUES (52, "Escopeta", 550)



Lenguaje de manipulación de datos (DML)

UPDATE nombreTabla **SET** nombreAtributo = valorAtributo

[**WHERE** condición]

Arma (codArma, nombre, descripcion, valor)

Ej: modificar el nombre y valor del arma con código 734.

UPDATE arma

SET nombre = 'Ametralladora', valor = 700

WHERE codArma = 734

Si no se especifica una condición, se actualizan todas las tuplas de la tabla.



Lenguaje de manipulación de datos (DML)

DELETE FROM nombreTabla [**WHERE** condición]

Arma (codArma, nombre, descripcion, valor)

Ej: borrar las armas con valor menor a 150.

DELETE FROM Arma

WHERE valor < 150

Si no se especifica una condición, se eliminan todas las tuplas de la tabla.

A vertical image on the left side of the slide shows a person from the side, wearing a headset and working at a desk. They are looking at a large monitor that displays a complex, colorful data visualization or network diagram. The person is wearing a dark shirt and a headset with a microphone.

Lenguaje de manipulación de datos (DML)

En la clase de hoy vimos algunas alternativas de **tipos de BD**.

Dependiendo del contexto, el tipo de información y el volumen de datos a manejar, se deberá **elegir el tipo de BD adecuada para el proyecto**.

Además, vimos las diferentes etapas del **proceso de diseño** de una BD relacional, para obtener una **estructura de base de datos eficiente**.

Por último, se analizaron las **operaciones básicas de SQL** para crear una base de datos relacional y manipular la información almacenada en ella.