



# *Bases de Datos*

## Tema 4: El modelo de datos relacional

*Dpto. de Ingeniería Informática*

# Contenidos

- Objetivos
- Introducción
- El MD relacional
- Doce reglas de Codd
- Lenguajes
- Referencias

# Introducción

- El modelo relacional es un modelo de datos global, llamados convencionales o lógicos
- Es un modelo sencillo y fácil de entender ya que está formado solamente por el concepto matemático de relación
- Una relación se representa como una tabla bidimensional donde los datos se muestran utilizando la estructura registro para formar el cuerpo de dicha tabla
- Este modelo soporta un lenguaje de programación llamado SQL (lenguaje de consulta estructurado)
- Propuesto por Codd en 1970

# Introducción

- Objetivos:
  - Independencia física: modificación del almacenamiento físico sin modificación de aplicaciones
  - Independencia lógica: manipulación lógica de la BD sin modificación de las aplicaciones
  - Flexibilidad: ofrecer los datos a las aplicaciones como se requieran
  - Uniformidad: presentación de la información en tablas
  - Sencillez: manipulación con lenguajes sencillos
- Se divide en dos partes:
  - *Estática*: está formada por el concepto de tabla o relación y los términos asociados a este concepto
  - *Dinámica*: formada por un conjunto de operadores, que se aplican a las relaciones, conformando el álgebra y el cálculo relacional

# Introducción

- Toda BD necesita, además de los datos, unas reglas de integridad para que se contemple las restricciones del mundo a modelar
- Toda BD está sujeta a un gran número de estas reglas y son de dos tipos:
  - *Generales* para toda BD: claves primarias y foráneas
  - *Específicas* para cada BD: todo hijo tiene una sola madre
- Diseño basado en la teoría de la normalización, eliminando dependencias entre atributos
  - Las tres FN de Codd
  - FN de Boyce-Codd
  - Formas normales de Fagin

# Introducción

- Los primeros productos comerciales fueron:
  - QBE: 1978, lenguaje de acceso relacional a ficheros VSAM de IBM. No era un verdadero SGBDR
  - Oracle: 1979, primer SGBDR comercial, con lenguaje SQL para definición y manipulación de datos
  - Ingres: 1980, origen en el prototipo de igual nombre de la Univ. de Berkeley, con lenguaje QUEL basado en cálculo relacional. Más utilizado por las universidades, en especial las de EE.UU.

# Introducción

- Etapas:
  - Prerrelacional: 1ª generación de BD. Los SGBD se soportan con los modelos de datos Codasyl y Jerárquico
  - Relacional: 2ª generación de BD. Se van consolidando los sistemas relacionales y van desplazando a los sistemas de 1ª generación
  - Postrelacional: 3ª generación de BD. Aparición de otros modelos de datos: basados en objetos, en grafos, NoSQL, ...

# Estructura

- Representa a una BD como un conjunto de tablas o relaciones
- Una BD relacional se divide en tres partes y cada una de ellas tiene sus propios términos especiales:
  - Estructura: conjunto de relaciones que forman la BD
  - Integridad: conjunto de restricciones de la BD
  - Manipulación: conjunto de operaciones sobre la información contenida en la BD
    - introducción
    - Extracción (consultas/informes)
    - actualización
    - borrado de información



# Estructura

- Terminología en BD (~equivalencia)

<b>Término</b>	<b>Equivalencia</b>
Relación	Tabla
Tupla	Fila o registro
Cardinalidad	Número de filas
Atributo	Columna o campo
Grado	Número de columnas
Clave primaria	Identificador único
Dominio	Conjunto de valores

# Tablas

- Las tablas son estáticas, y se definen en un esquema que indica atributos y restricciones
  - El diseño de una BD relacional consiste en un conjunto de esquemas de relaciones, llamado esquema de la BD relacional o *esquema de la BD*
  - Los nombres de tablas son únicos en el esquema
  - Debe cambiar poco en el tiempo
- Ese esquema almacena un conjunto de datos (tuplas) en un momento concreto, definiendo una instancia
  - Son los que interesan a los usuarios / aplicaciones
  - Evolucionan en el tiempo

# Estructura

- Ejemplo de esquema e instancias de la tabla *Autores* de la base de datos *Editorial*:

Código	Nombre	Apellido
1	Neal	Stephenson
2	Lawrence	Lessig
3	Noam	Chomsky

# Atributos

- Los atributos de una relación se corresponden con los nombres de las columnas de dicha relación
- Toda tabla/relación está formada por varios atributos que describen los significados de las entradas de datos en las columnas
- Por ejemplo, en la relación *Autores*, la columna con el atributo *Código* contiene los números de códigos de los autores cuyos datos están en dicha relación
- Los nombres de los atributos deben ser únicos en una relación, siendo inequívoca la nomenclatura:

*nombre\_tabla.nombre\_atributo*

# Esquemas

- El esquema de una relación está formado por el nombre y el conjunto de atributos
- Se utiliza la siguiente nomenclatura: nombre de la relación seguida, entre paréntesis, de los atributos que la forman
- Ej: el esquema de la relación *Autores*:  
*Autores (Código, Nombre, Apellido)*
- Pregunta: ¿Hay orden concreto en los atributos de una relación?

# Esquemas

- ¿Hay orden concreto en los atributos de una relación?
  - Algunos autores defienden que no: no es necesario añadir un orden que complique el modelo. Por lo tanto, los atributos serían un conjunto
  - Otros defienden que imponer un orden concreto no complica en exceso el modelo y facilita significativamente la nomenclatura
    - Ej: INSERT (8, Richard, Stallman) INTO Autores → Código es 8, Nombre es Richard, Apellido es Stallman)
    - Usaremos esta última aproximación

# Tuplas e instancias

- Las tuplas son los registros de una relación y suelen tener un valor para cada uno de los atributos de la relación
- Todas las tuplas de una misma relación son diferentes, y además el valor del *identificador único* ha de ser distinto para cada una de ellas
  - Este identificador único puede estar formado por un único atributo o un conjunto de ellos
- El conjunto de todas las tuplas de una relación en cada instante se denomina *instancia*

# Dominios

- Un dominio es un conjunto de valores todos del mismo tipo
  - Cada atributo está definido sobre un único dominio
- Los dominios se especifican en la definición de la BD, indicando el dominio concreto de cada atributo
- Los dominios restringen las comparaciones, las operaciones y las expresiones que pueden realizarse con los atributos
- Cada componente de cada tupla debe ser atómico y de algún tipo “elemental” de datos
  - Representan la menor unidad semántica de información



# Dominios

- En esta asignatura consideraremos los siguientes tipos elementales:
  - Booleano
  - Entero
  - Decimal
  - Cadena de caracteres
  - Hora
  - Fecha (incluye hora)
  - BLOB (Binary Large Objects, objetos binarios grandes)

•

# Dominios

- Los SGBD suelen ofrecer un conjunto de dominios “base: booleano, entero, decimal, VARCHAR, fecha, BLOB
  - <https://dev.mysql.com/doc/refman/5.7/en/data-types.html>
- Algunos permiten definir dominios más finos
  - Ej: VARCHAR(8) para DNI, provincias de España
    - Importante: que soporten las operaciones que necesitan
  - Algunos comprueban que las operaciones no violen las definiciones de los atributos implicados
    - Pero no todos lo hacen → tener en cuenta estas restricciones a la hora de manipular los datos de la BD

# Relaciones

- Desde el punto de vista matemático, una relación es un subconjunto del producto cartesiano de un listado de dominios, no necesariamente todos distintos

$$\times_{i=1}^n D_i$$

- Toda relación tiene un nombre, un esquema y una instancia
- El esquema está formado por una serie fija de atributos que se representan por la pareja atributo-dominio

$$\langle (A_1 : D_1), (A_2 : D_2), \dots, (A_n : D_n) \rangle$$

- donde  $n$  es el grado o número de atributos de la relación

# Relaciones

- Cada tupla está compuesta por una serie de valores

$$\langle V_{i1}, V_{i2}, \dots, V_{in} \rangle$$

$$(i = 1, 2, \dots, m)$$

donde  $m$  es la cardinalidad o número de tuplas de la relación en un instante dado

- Ejemplo de tupla:

$$\langle 3, \text{Noam}, \text{Chomsky} \rangle$$

# Relaciones

- La instancia es el conjunto de tuplas en un momento dado
  - Ejemplo de instancia:  
 $\{ \langle 1, \text{Neal}, \text{Stephenson} \rangle, \langle 2, \text{Lawrence}, \text{Lessig} \rangle, \langle 3, \text{Noam}, \text{Chomsky} \rangle \}$
  - O igual:  
 $\{ \langle 2, \text{Lawrence}, \text{Lessig} \rangle, \langle 1, \text{Neal}, \text{Stephenson} \rangle, \langle 3, \text{Noam}, \text{Chomsky} \rangle \}$

# Propiedades

- Hay propiedades consecuencia de la propia definición de relación:
  - No existen tuplas repetidas: todos los objetos de un clase son diferentes
  - Las tuplas no están ordenadas: la instancia de una relación es un conjunto y los elementos de un conjunto no tienen orden
  - Los atributos sí están ordenados: el esquema de una relación es una serie, luego tiene orden
  - ...

# Propiedades

- Hay propiedades consecuencia de la propia definición de relación:
  - Todos los valores de los atributos son atómicos: toda relación está *normalizada* en primera forma (FN1), pues sus atributos no pueden ser relaciones
  - ¿Qué hago si tengo un atributo derivado?

# Propiedades

- Hay propiedades consecuencia de la propia definición de relación:
  - Todos los valores de los atributos son atómicos: toda relación está *normalizada* en primera forma (FN1), pues sus atributos no pueden ser relaciones
  - ¿Qué hago si tengo un atributo derivado?
    - Sólo almaceno los componentes
  - ¿Qué hago si tengo un atributo multivaluado?



# Propiedades

- Hay propiedades consecuencia de la propia definición de relación:
  - Todos los valores de los atributos son atómicos: toda relación está *normalizada* en primera forma (FN1), pues sus atributos no pueden ser relaciones
  - ¿Qué hago si tengo un atributo derivado?
    - Sólo almaceno los componentes
  - ¿Qué hago si tengo un atributo multivaluado?
    - Con carácter general dará lugar a una nueva relación
    - En algún caso puede admitir:
      - Poner repeticiones (3 teléfonos máximo por persona)
      - Ponerlo varias tuplas, en cada una de ellas uno de los múltiples valores

# Tipos de relaciones

- Estos tipos de relaciones son los más usuales, aunque no todos los SGBD los manejan:
  - *Relaciones base*: es una relación autónoma. Se crea explícitamente (nombre y conjunto de atributo/dominio). Tanto el esquema como la instancia se almacenan en la BD
  - *Vistas*: son relaciones virtuales con nombre pero sin datos (nombre asociado a una consulta). Corresponde al nivel externo de la arquitectura ANSI
  - *Instantáneas*: es una relación derivada con nombre (*snapshot*). Se ejecuta una consulta y el resultado se almacena en la BD aunque se va refrescando regularmente por el propio sistema

# Tipos de relaciones

- *Resultados de consultas*: es una relación resultante de una consulta y no tiene existencia en la BD
  - Pueden tener o no nombre
- *Resultados intermedios*: es una relación, normalmente sin nombre, resultado de una expresión relacional anidada dentro de otra expresión relacional
- *Relaciones temporales*: es una relación con nombre pero que se destruye de forma automática en cualquier momento

# Resumen

- Cada relación tiene un nombre que la identifica y un esquema formado por una o más columnas
- Cada relación tiene cero o más tuplas conteniendo un único valor en cada columna
  - Una tabla vacía es una tabla “válida” que informa de algo concreto
- Todos los valores de una columna determinada tienen el mismo tipo de datos, extraídos del dominio de la columna
- Las tablas se relacionan entre si por los datos que las contienen. Este modelo utiliza:
  - Clave primaria
  - Clave foránea
  - Combinación de claves primaria/foránea

# Claves primarias

- Toda relación tiene un identificador único formado por uno o varios atributos de su esquema
- Una relación puede tener más de un identificador, llamados *claves candidatas*
- La *clave primaria* (CP) es un miembro del conjunto de claves candidatas
- Las *claves alternativas* son las claves candidatas que no han sido elegidas como clave primaria

# Claves primarias

- Clave candidata: el atributo  $K$ , simple o compuesto, de la relación  $R$  es una clave candidata de  $R$  si satisface las dos propiedades siguientes independientes en el tiempo:
  - *Unicidad*: en cualquier momento dado, no existen dos tuplas en  $R$  con el mismo valor de  $K$
  - *Minimalidad*: si  $K$  es compuesto, no será posible eliminar ningún componente de  $K$  sin destruir la propiedad de unicidad

# Claves primarias (resumen)

- Toda relación tiene al menos una clave candidata que será la clave primaria
  - En el “peor” de los casos serian todos sus atributos
- El *DBA* elige la clave primaria entre las candidatas y el razonamiento para la elección queda fuera del modelo relacional
  - Se suele elegir aquella más simple/corta/manejable
- La *clave primaria* suele ser el principal mecanismo de búsqueda de tuplas
- La clave primaria puede estar formada por más de un atributo, luego debemos decir el *conjunto de clave primaria*
- No debemos emplear el término *clave* (a secas) para la clave primaria, ya que puede llevar a error al existir diferentes tipos de claves en BD

# Claves primarias (resumen)

- No debemos emplear el término *clave* (a secas) para la clave primaria, ya que es ambiguo (existen varios tipos de claves en BD):
  - clave primaria
  - claves candidatas
  - claves alternativas
  - Super-clave: serie de atributos que identifican inequívocamente cada tupla, pero no es mínimo (eliminándole atributos mantienen la unicidad)



# Regla de integridad de las relaciones

- El modelo relacional tiene dos reglas de integridad:
  - Integridad de las relaciones
  - Integridad referencial
- *Regla de integridad de las entidades*: ningún componente de la clave primaria de una relación base puede aceptar nulos
  - En una BD relacional, no se registra información de algo que no se puede identificar
    - Cada valor individual de la CP debe ser no nulo en su totalidad
    - La regla se aplica a relaciones base
    - Esta regla sólo debe cumplirla la CP, cualquier otro tipo de clave no tiene porqué

# Claves foráneas

- Las CF nos permiten interrelacionar relaciones
- Clave foránea (CF): es un atributo, o una serie de ellos, de una relación  $R_2$  cuyos valores deben concordar con los de la CP de alguna relación  $R_1$
- $R_1$  y  $R_2$  no tienen que ser distintas
  - ¿En qué caso podrían no ser distintas?

# Claves foráneas

- Las CF nos permiten interrelacionar relaciones
- Clave foránea (CF): es un atributo, o una serie de ellos, de una relación  $R_2$  cuyos valores deben concordar con los de la CP de alguna relación  $R_1$
- $R_1$  y  $R_2$  no tienen que ser distintas
  - ¿En qué caso podrían no ser distintas?
    - Relaciones reflexivas. Por ejemplo, alumnos que mentorizan a otros alumnos

# Claves foráneas

- La CP y la CF deben estar definidas sobre los mismos dominios
- La CF no necesita ser un componente de la CP de la relación que la contiene
- Toda clave foránea debe satisfacer estas dos propiedades independientes en el tiempo:
  - Cada valor de  $CF$  es nulo del todo (todos sus componentes son nulos) o bien no nulo del todo
  - Para cada valor no nulo de  $CF$  existe en la relación base  $R_1$  una tupla que tiene dicho valor de  $CP$

# Regla de integridad referencial

- *Regla de integridad referencial:* La BD no puede contener valores de clave foránea sin concordancia
- Si existe un valor de clave foránea debe existir un valor concordante de clave primaria
- La definición de clave foránea está ligada al concepto de integridad referencial

# Restricciones

- El modelo relacional tiene estructuras y ocurrencias no permitidas
- Los datos almacenados han de adaptarse a las estructuras impuestas y cumplir las restricciones de usuario para tener ocurrencias válidas del esquema
- Se dividen en:
  - *Restricciones inherentes*: son propias de cada modelo y quita flexibilidad para representar el mundo real
  - *Restricciones semánticas*: o de usuario, son facilidades que el modelo ofrece para poder representar la semántica del mundo real

# Restricciones

- Restricciones inherentes:
  - No son definidas por el usuario. Son propias del modelo de datos
  - Quita flexibilidad para representar el mundo a modelar
  - Como modelo matemático, se debe cumplir:
    - No hay dos las tuplas iguales. Obligatorio CP
    - No hay orden de tuplas
    - Los atributos sólo pueden tomar un único valor sobre el dominio (son atómicos)
    - Regla de integridad de las relaciones

# Restricciones

- Restricciones semánticas:
  - Reglas añadidas por el diseñador para reflejar el mundo a modelar
  - Restricciones principales:
    - Clave primaria (*primary key*): permite declarar una CP (restricción inherente). La decisión de cuál es semántica
    - Unicidad (*unique*): los valores de los atributos no se pueden repetir en una relación
    - Obligatoriedad (*not null*): los atributos no pueden ser nulos
    - Integridad referencial (*foreign key*): las CF deben concordar con la CP a la que hace referencia o ser nulas



# Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
  - En órdenes de inserción de datos: ¿qué hay que comprobar?

# Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
  - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
    - ¿Qué hago si se viola?

# Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
  - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
    - Solución: rechazar la solicitud (error)
  - En órdenes de borrado de datos: ¿qué hay que comprobar?

# Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
  - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
    - Solución: rechazar la solicitud (error)
  - En órdenes de borrado de datos: que la CP que se pierda no deje una CF de otra tabla sin emparejar
    - Soluciones?

# Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
  - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
    - Solución: rechazar la solicitud (error)
  - En órdenes de borrado de datos: que la CP que se pierda no deje una CF de otra tabla sin emparejar
    - Soluciones: rechazar la solicitud o propagar el efecto

# Restricciones

- El SGBD vigila que al realizar operaciones que modifiquen la instancia su resultado siga respetando las restricciones del sistema
  - En órdenes de inserción de datos: comprobar dominios, CP, UNIQUE, NOT NULL y CF
    - Solución: rechazar la solicitud (error)
  - En órdenes de borrado de datos: que la CP que se pierda no deje una CF de otra tabla sin emparejar
    - Soluciones: rechazar la solicitud o propagar el efecto
  - En órdenes de modificación de datos (UPDATE): se monta a partir de las anteriores

# Valores nulos en el MR

- Definición:
  - Señal utilizada para representar:
    - Información desconocida,
    - Inaplicable,
    - Inexistente,
    - No válida,
    - No proporcionada,
    - Indefinidad, ...
  - Necesidad de valores nulos:
    - Inserción de tuplas con ciertos atributos desconocidos
    - Añadir atributos nuevos al esquema de una relación
    - Atributos inaplicables a ciertas tuplas

# Valores nulos en el MR

- El tratamiento de valores nulos exige:
  - Operaciones de comparación: problemas para saber si dos valores nulos son o no son iguales
  - Operaciones aritméticas: son nulas si alguno de los operandos es nulo
  - Operaciones algebraicas: hay problemas en algunas operaciones como la unión externa
  - Funciones de agregación: normalmente los valores nulos no entran a formar parte de las tuplas evaluadas. Ej: media aritmética



# Doce reglas de Codd

- *Las 12 reglas de Codd son un sistema de reglas (numeradas del 0 al 12) propuestas por Edgar F. Codd, del modelo relacional para las bases de datos, diseñado para definir qué requiere un sistema de administración de base de datos*
  - *Codd se percató de que existían bases de datos en el mercado las cuales decían ser relacionales, pero lo único que hacían era guardar la información en las tablas, sin estar estas tablas literalmente normalizadas; entonces éste publicó 13 reglas que un verdadero sistema relacional debería tener aunque en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse "más relacional" cuanto más siga estas reglas*

# Doce reglas de Codd

- 0: Cualquier sistema que se proclame como relacional, debe ser capaz de gestionar sus bases de datos enteramente mediante sus capacidades relacionales *[también de Wikipedia]*
- 1 Representación de la información: toda información debe representarse en forma de tabla, principio básico del modelo relacional
- 2 Acceso garantizado: todo dato debe ser accesible mediante una combinación de un nombre de tabla, un valor de su clave y el nombre de una columna
- 3 Tratamiento sistemático de valores nulos: estos valores han de ser tratados sistemáticamente por el sistema, el cual ha de ofrecer las posibilidades necesarias para su tratamiento
- 4 Catálogo en línea basado en el modelo relacional: la representación de la descripción de la BD debe ser igual a la de los otros datos, y su acceso debe poder realizarse por medio del mismo lenguaje relacional. El MD para la metabase debe ser también relacional

# Doce reglas de Codd

- 5 Sublenguaje de datos completo: debe existir un lenguaje que permita un completo manejo de la BD (definición y manipulación de datos, definición de vistas, restricciones de integridad, autorizaciones y gestión de transacciones)
- 6 Actualización de vista: toda vista teóricamente autorizable debe poder ser actualizada por el sistema
- 7 Inserción, actualización y eliminación de alto nivel: todas las operaciones de manipulación de datos deben operar sobre conjuntos de filas
- 8 Independencia física de los datos: el acceso lógico a los datos debe mantenerse incluso cuando cambien los métodos de acceso o la forma de almacenamiento

# Doce reglas de Codd

- 9 Independencia lógica de los datos: los programas de aplicación no deben verse afectados por cambios realizados en el esquema de la BD
- 10 Independencia de integridad: las reglas de integridad de una BD deben ser definibles por medio del sublenguaje de datos relacional y habrán de almacenarse en el DD de la BD, no en los programas de aplicación
- 11 Independencia de distribución: debe existir un sublenguaje de datos que pueda soportar BD distribuidas sin alterar los programas de aplicación cuando se distribuyan los datos
- 12 Regla de la no subversión: si un SGBD soporta un lenguaje de bajo nivel que permita el acceso fila a fila, éste no puede utilizarse para saltarse las reglas de integridad expresadas por medio del lenguaje de más alto nivel

# Lenguajes de consulta

- Sirven para que el usuario manipule la información de la BD
- Se clasifican en:
  - *Procedimentales*: el usuario indica las operaciones y la secuencia en que se deben realizar. Álgebra relacional
  - *No procedimentales*: el usuario indica lo que quiere obtener sin especificar cómo obtenerlo. Cálculo relacional
- Los lenguajes de consultas pasaron a denominarse de manipulación (*DML*), ya que también permiten insertar, eliminar y modificar la información

# Lenguajes de consulta

- Ejemplo de Álgebra relacional
  - Fechas en las que sacaron todos los libros con código mayor que 10

$$A \leftarrow (\Pi_{\text{cód\_Lib}, \text{Fecha}}(\text{Préstamos}))$$

$$B \leftarrow (\sigma_{\text{Cód\_Lib} > 10}((\Pi_{\text{Cód\_Lib}}(\text{Libros})))$$

$$A/B$$

- Hay que indicar cada paso que se da

# Lenguajes de consulta

- Ejemplo de Cálculo relacional
  - Listar los datos de los socios que tienen al menos un libro en préstamo así como el código del libro y la fecha del préstamo
  - *Resultado*(Cód\_Soc, Nombre, Apellido, Cód\_Lib, Fecha)  
 $\{t \mid t \in \text{Resultado}\}$   
 $\{t \mid (\exists r) ((r \in \text{Prestamos}) \wedge (r.\text{Cód\_Soc} = t.\text{Cód\_Soc})) \wedge (\exists s) ((s \in \text{Socios}) \wedge (s.\text{Cód\_Soc} = t.\text{Cód\_Soc}))\}$
  - Hay que indicar las condiciones del resultado

# Lenguajes de consulta

- Los SGBD disponen de:
  - *DML*: Lenguaje de manipulación de datos: manipulación de la información (instancias) de la BD
  - *DDL*: Lenguaje de definición de datos: creación de los esquemas de las relaciones
  - *DCL*: Lenguaje de control de datos: control de la seguridad de los datos, seguridad de las estructuras de esos datos, control de acceso por parte de los usuarios, información y auditoría del sistema, etc
  - *Lenguajes huéspedes*: lenguaje de alto nivel que soporta instrucciones de un lenguaje de manipulación de datos, como por ejemplo SQL, y que interacciona con la BD



# Referencias

- Apuntes Esther Gadeschi
- Libro Elmasri, 3ª ed.

Gracias por la atención  
*¿Preguntas?*