

Diseño de Bases de Datos
Bases de Datos Documentales
Grao en Información e Documentación
Curso 2013/2014

Miguel Ángel Rodríguez Luaces
Laboratorio de Bases de Datos
Universidade da Coruña

Diseño de Bases de Datos

El último día...

- Los problemas de no utilizar un SGBD:
 - Redundancia: información repetida
 - Inconsistencia: información incoherente
 - Aislamiento: no es posible saber qué tienen otros
 - Dificultad de acceso:
 - Los ficheros están repartidos
 - Consultarlos es muy complicado
 - Integridad de la información: no hay validación
 - Acceso concurrente: ¿y si hay dos bibliotecarios?
 - Atomicidad: ¿y si tengo que cambiar dos ficheros y sólo lo hago en uno?
 - Seguridad: ¿y si Windows se come el fichero?

Diseño de Bases de Datos

El último día...

- Utilizando un sistema gestor de bases de datos
 - Contra la redundancia y el aislamiento:
 - Centralización de los datos
 - Contra la dificultad de acceso:
 - Interfaces de acceso a la información
 - Lenguajes de consulta
 - Contra la integridad:
 - Restricciones de integridad (validaciones)
 - Contra los problemas de concurrencia:
 - Gestión de usuarios y bloqueos de datos
 - Para la seguridad:
 - Políticas de copias de seguridad

Diseño de Bases de Datos

El último día...

- Utilizando un SGBD documental
 - Los documentos no son independientes de su registro bibliográfico, sino que su texto está almacenado en la BD
 - El SGBD permite hacer consultas al texto de los documentos
 - El texto completo del documento se puede localizar a partir del catálogo

Diseño de Bases de Datos

El último día...

- Características de las BDDoc. por su contenido
 - Referenciales: información del documento, pero no su contenido. Ej: la BD del OPAC
 - De texto completo: contiene el texto del documento y permite búsquedas por contenido. Ej: la BD de la BVG
 - Con reproducción del documento: contiene las páginas escaneadas. Ej: la BD de la hemeroteca de la RAG
- Características de las BDDoc. en base al modelo de tratamiento documental:
 - BD de sumarios (sin análisis del contenido)
 - Catálogo de biblioteca (según normas internacionales)
 - Análisis documental completo:
 - Descriptores
 - Palabras clave
 - Texto completo

Diseño de Bases de Datos

El último día...

- Aprenderemos a diseñar bases de datos
- Aprenderemos a extraer información de la BD
 - Lenguaje de consulta *universal*: SQL
- Aprenderemos cómo funciona una BD Documental
- Aprenderemos como marcar documentos para su procesamiento automático: XML
 - Metadatos: Dublin Core
 - Catálogo: MarcXML

Diseño de Bases de Datos

Objetivo

- ¿Por qué es necesario diseñar la BD? ¿No es una pérdida de tiempo?
- ¿Cómo realizo el diseño?
- ¿Cómo describo la realidad en un papel para que cualquiera me entienda?

Diseño de Bases de Datos

Problemas de un mal diseño

- En una BD mal diseñada pueden ocurrir estos problemas
 - Anomalías de inserción
 - Redundancia por tener que almacenar información repetida
 - Complejidad por tener que realizar múltiples operaciones
 - Ineficiencia por necesitar filas *medio llenas*
 - Anomalías de borrado
 - Complejidad para evitar borrar información sin desearlo
 - Ineficiencia por necesitar filas *medio llenas*
 - Anomalías de modificación
 - Complejidad por tener que modificar la información redundante

Diseño de Bases de Datos

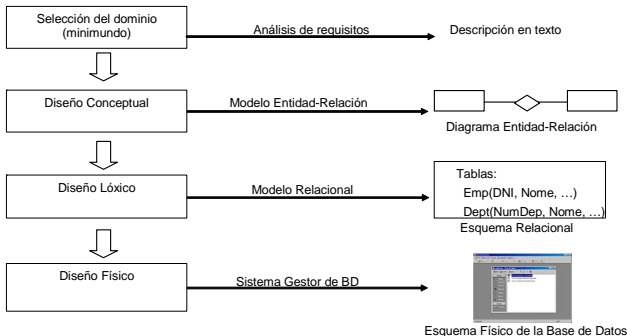
Problemas de un mal diseño

- Estas anomalías las causa un mal diseño
- Un diseño correcto evita todos estos problemas:
 - Una inserción implica una única fila nueva
 - Un borrado no provoca pérdida de información adicional
 - Una modificación es una operación sencilla

Diseño de Bases de Datos

El proceso de diseño

- Conseguir un buen diseño directamente no es fácil. Por ello, el problema se descompone en pasos



Diseño de Bases de Datos

El proceso de diseño

- La preocupación en cada paso es distinta:
 - Análisis de requisitos
 - ¿Qué información necesito almacenar?
 - ¿Qué consultas voy a querer realizar?
 - Diseño conceptual
 - ¿Cómo organizo la información?
 - Utilizo un lenguaje claro y no ambiguo
 - Diseño lógico
 - ¿Cómo represento el diseño conceptual en un ordenador?
 - El resultado debe ser utilizable en cualquier SGBD
 - Diseño físico
 - Implementación en un SGBD particular

Modelo conceptual

El modelo E-R

- Organiza de forma sencilla la información
- Cinco conceptos
 - Entidad: elemento del mundo real del que almacenamos información
 - Atributo: elemento de información *atómico* de una entidad
 - Atributo clave: el que permite identificar únicamente a cada entidad
 - Relación: conexión entre dos entidades
 - Cardinalidad: número de entidades que participan en una relación

Modelo conceptual

El modelo E-R: Notación

Entidad:



Nombre

Atributo:



Nombre

Atributo clave:



Nombre

Relación:



Nombre

Modelo conceptual

El modelo E-R: Notación

Cardinalidad:



- Cada entidad A se relaciona con N entidades B
- Cada entidad B se relaciona con 1 entidad A
- Tipos de cardinalidad
 - Uno a uno (1:1)
 - Uno a muchos (1:N)
 - Muchos a muchos (M:N)

Modelo conceptual

El modelo E-R: Reglas comunes de modelado

- Las entidades suelen identificarse por sustantivos, las relaciones suelen corresponder a verbos
- La entidad es algo que existe por sí mismo, las relaciones sólo tienen sentido conectando otras entidades
- La diferencia entre atributo y entidad es difusa. El factor que decide es que el atributo es un valor simple que será tratado como una unidad
- Los atributos clave son muy importantes. Se usarán en todo el proceso de modelado
- Si no se encuentra un atributo clave apropiado, se puede inventar uno (código, número de registro)

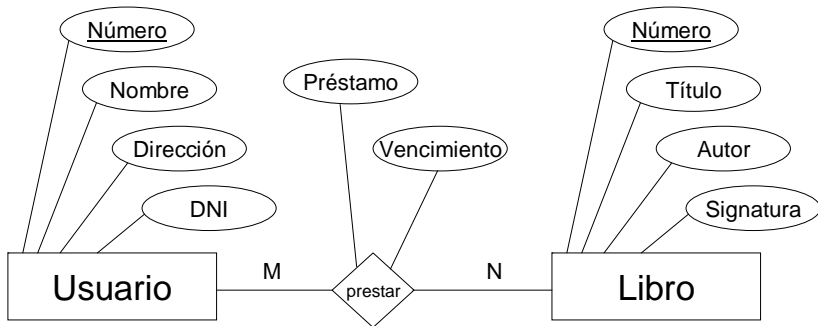
Modelo conceptual

El modelo E-R: Ejemplo

- En la biblioteca se mantiene un registro de usuarios. Para cada uno se almacena el DNI, nombre, domicilio y número de usuario.
- La biblioteca posee una colección de libros. De cada uno almacenamos el número de registro, el título, el autor y la signatura.
- La biblioteca realiza prestamos de los que nos interesan las fecha de retirada y devolución

Modelo conceptual

El modelo E-R: Ejemplo



Modelo conceptual

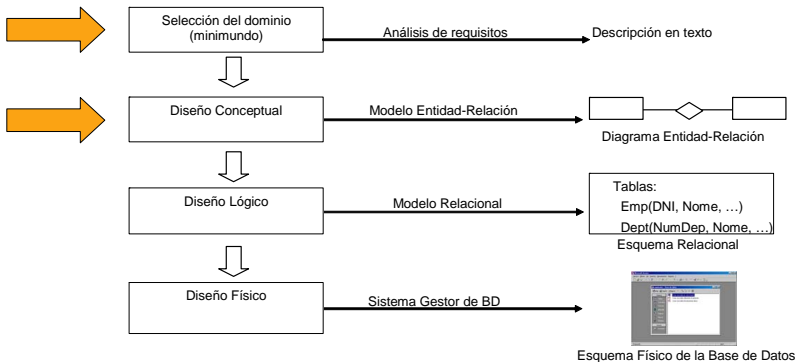
Conclusión

- ¿Por qué es necesario diseñar la BD?
 - Para evitar las anomalías de diseño
- ¿No es una pérdida de tiempo?
 - No. La BD resultante tiene más calidad y el diseño nos sirve de documentación
- ¿Cómo realizo el diseño?
 - Siguiendo el proceso de modelado (conceptual, lógico, físico)
- ¿Cómo describo la realidad en un papel para que cualquiera me entienda?
 - Utilizando un modelo entidad-relación

El Modelo Relacional

Diseño de Bases de Datos

- ¿Qué hemos hecho hasta ahora?



El Modelo Relacional

Diseño de Bases de Datos

- El siguiente paso es el diseño lógico:
 - ¿Cómo represento el diseño conceptual en un ordenador?
 - El resultado debe ser utilizable en cualquier SGBD
- Para representar el diseño lógico utilizaremos el modelo relacional
 - Propuesto en 1970 para aislar el diseño de la base de datos del entorno físico
 - Utilizado en todos los sistemas gestores de bases de datos actuales

El Modelo Relacional

Conceptos básicos

- Una **relación** es un conjunto de **tuplas**
- Una **tupla** es un conjunto de **valores atómicos**
- Cada elemento de la tupla es un **atributo**
- Cada atributo tiene un **nombre** y un **dominio**
 - El nombre no se puede repetir
 - El dominio es el conjunto de valores posibles para el atributo
- Una relación se describe mediante su **esquema**
 - El esquema enumera el nombre de la relación, sus atributos y sus dominios

El Modelo Relacional

Conceptos básicos

- La representación gráfica de una relación es una tabla:

empleados

dni	nombre	apellido1	apellido 2	nacimiento
32677332	Miguel	Rodríguez	Luaces	11/07/1975
36473821	Ana	García	López	07/11/1977
32665212	Pedro	Pérez	Patiño	04/04/1944
43435332	Marcos	López	Golpe	01/12/1987

El Modelo Relacional

Conceptos básicos

- Ejemplo de esquema

empleados(dni:número → **Atributo**
 nombre: texto,
 apellido1: texto, → **Nombre**
 apellido2: texto,
 nacimiento: fecha → **Dominio**
)

- Ejemplo de tuplas:

(32677332, Miguel, Rodríguez, Luaces, 11/07/1975)
(36473821, Ana, García, López, 7/11/1977)
(32665212, Pedro, Pérez, Patiño, 4/4/1944)
(43435332, Marcos, López, Golpe, 1/12/1987)

El Modelo Relacional

Conceptos básicos

- Las tuplas **no** están ordenadas
 - La base de datos devuelve los resultados como le apetece
- Los atributos **no** están ordenados
 - Se pueden representar en cualquier orden
- Los valores son **atómicos**
 - La base de datos los interpreta como un único objeto. En general, no se puede extraer sólo un pedazo del mismo
- Cualquier atributo puede tener como valor un **nulo**
 - Un nulo representa un valor desconocido, un valor que no existe, o que no es aplicable

El Modelo Relacional

Restricciones de integridad

- Aseguran que las relaciones se pueden manipular mediante programas informáticos muy eficientemente
- Restricción de dominio:
 - Los dominios de los atributos deben ser atómicos
 - Nuestras tablas deben ser planas (no hay atributos que son a su vez compuestos)
- Restricción de clave:
 - En una relación no puede haber ninguna tupla repetida
 - Toda relación debe tener una clave primaria: un conjunto de atributos que identifican únicamente a cada tupla de la relación

El Modelo Relacional

Restricciones de integridad

- Restricción de integridad de entidad:
 - Ninguna tupla puede tener valores nulos en la clave primaria
- Restricción de integridad referencial
 - En una relación puede haber claves foráneas: un atributo cuyo valor hace referencia a un tupla de otra relación
 - Si una tupla de R1 referencia una tupla de R2, la tupla de R2 tiene que existir

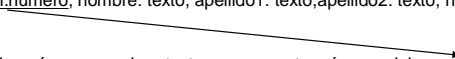
El Modelo Relacional

Representación de una base de datos

- Formada por los esquemas de todas las relaciones
- Se subrayan los atributos que forman la clave primaria
- Se indica a qué relación referencian las claves foráneas
- Ejemplo:

empleados(dni:número, nombre: texto, apellido1: texto, apellido2: texto, nacimiento: fecha)

proyectos(código:número, nombre: texto, presupuesto: número, dniresponsable: número)



El Modelo Relacional

Transformación del modelo E-R al modelo relacional

- Cada **entidad** es una **relación**
 - Cada **atributo** de la entidad es un **atributo** de la relación
 - Hay que determinar los **dominios** de cada atributo
 - La **clave** de la entidad es **clave primaria** de la relación
- Cada **relación** de cardinalidad **1:N**
 - La **clave primaria** del lado 1 es **clave foránea** en la relación del lado N
 - Los **atributos de la relación** también van al lado N
- Cada **relación** de cardinalidad **1:1**
 - Se escoge uno de los dos lados y se trata como una relación de cardinalidad 1:N

El Modelo Relacional

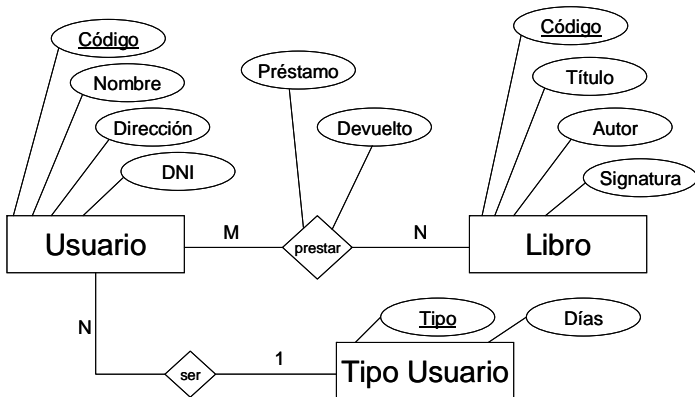
Transformación del modelo E-R al modelo relacional

- Cada **relación** de cardinalidad **M:N**
 - Se crea una **relación**
 - Los **claves primarias** de cada lado de la relación (E-R) **son claves foráneas** de la relación
 - Los **atributos** de la relación (E-R) son **atributos** de la relación
 - La clave primaria de la relación son las claves foráneas. Alguna vez, se necesita algún atributo adicional

El Modelo Relacional

Ejemplo de transformación

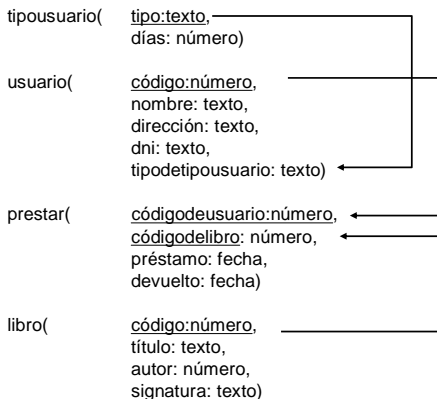
- Modelo entidad-relación:



El Modelo Relacional

Ejemplo de transformación

● Modelo relacional:



El Modelo Relacional

Ejemplo de transformación

- Tablas:

tipousuario

tipo	días
Alumno	7
Profesor	365

El Modelo Relacional

Ejemplo de transformación

● Tablas:

usuario

código	nombre	dirección	dni	tipodeusuario
1	Miguel	Real 3	3-b	Profesor
2	Ana	Sol 11	2-p	Alumno

tipousuario

tipo	días
Alumno	7
Profesor	365



El Modelo Relacional

Ejemplo de transformación

● Tablas:

usuario

código	nombre	dirección	dni	tipodeusuario
1	Miguel	Real 3	3-b	Profesor
2	Ana	Sol 11	2-p	Alumno

tipousuario

tipo	días
Alumno	7
Profesor	365

libro

código	título	autor	signatura
1	HTML	López	TI 20
2	XML	Pérez	TI 15
3	XML	Pérez	TI 15

El Modelo Relacional

Ejemplo de transformación

• Tablas:

usuario

código	nombre	dirección	dni	tipodeusuario
1	Miguel	Real 3	3-b	Profesor
2	Ana	Sol 11	2-p	Alumno

tipousuario

tipo	días
Alumno	7
Profesor	365

libro

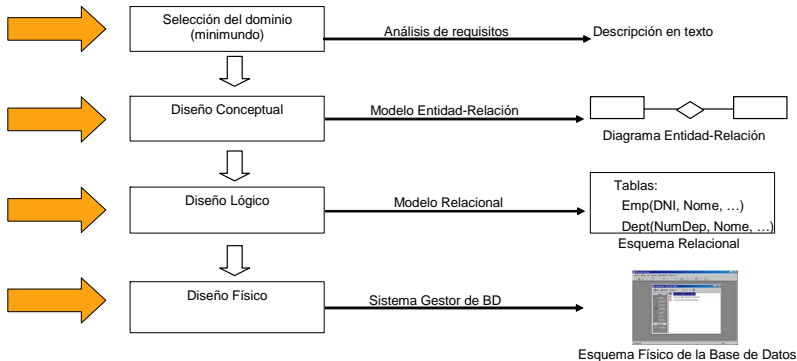
código	título	autor	signatura
1	HTML	López	TI 20
2	XML	Pérez	TI 15
3	XML	Pérez	TI 15

prestar

codigode usuario	codigode libro	préstamo	devuelto
1	1	10/10/06	28/03/07
1	2	28/03/07	
2	1	29/03/07	
2	2	01/03/07	08/03/07
2	2	08/03/07	15/03/07
2	2	15/03/07	27/03/07

El Lenguaje SQL

¿Qué hemos hecho hasta ahora?



El Lenguaje SQL

¿Qué hemos conseguido?

- Representar la información del mundo real de forma eficiente en un Sistema Gestor de Bases de Datos (SGBD)

¿Qué necesitamos ahora?

- Un método para extraer la información del SGBD que sea:
 - Sencillo
 - Eficaz
 - Utilizable en cualquier SGBD

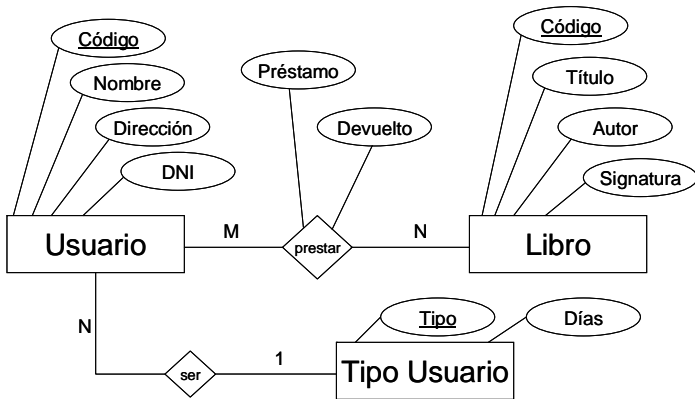
El Lenguaje SQL

Nuestra solución: el lenguaje SQL

- SQL significa Structured Query Language
- Es un lenguaje declarativo: se indica qué se quiere, no cómo obtenerlo
- Es un lenguaje de alto nivel: no es un lenguaje para ordenadores, sino que es casi inglés
- Es un lenguaje completo:
 - Permite recuperar información de la BD
 - Permite crear, actualizar y borrar información de la BD
 - Permite definir la estructura de la BD

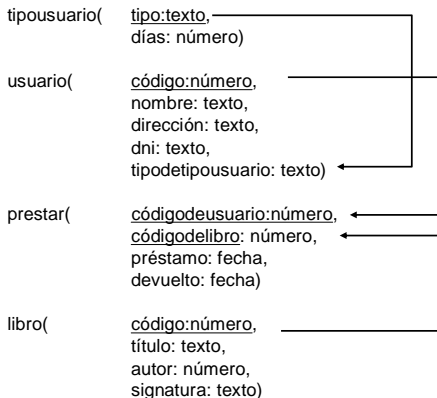
El Lenguaje SQL

Vamos a utilizar el siguiente ejemplo:



El Lenguaje SQL

Que generaba este modelo relacional:



El Lenguaje SQL

Cualquier consulta a una BD siempre sigue este formato

```
SELECT <lista de columnas>  
FROM <lista de tablas>  
WHERE <condición de selección>  
ORDER BY <lista de orden>
```

Que quiere decir:

A partir de todas las tablas en <lista de tablas>, quédate con aquellas filas que cumplan <condición de selección>, muéstrame una nueva tabla cuyas columnas sean <lista de columnas> y ordena las nuevas filas según <lista de orden>

El Lenguaje SQL

¿Qué hace el SGBD?

- Recopilar todas las tuplas de las relaciones de la línea FROM
- Para cada una de las tuplas, evaluar la línea WHERE
- Si la condición de la línea WHERE se cumple:
 - Se añade al resultado una nueva tupla, cuyo contenido es el resultado de evaluar la línea SELECT
- Si la condición no se cumple, la tupla se descarta
- Se ordena el resultado usando las expresiones de la línea ORDER BY

El Lenguaje SQL

Ejemplo

- Nombre y DNI de los usuarios de la biblioteca [Consulta 01]

```
SELECT nombre, DNI  
FROM usuario
```

- Resultado:

nombre	dni
Miguel	3-b
Ana	2-p

El Lenguaje SQL

La línea SELECT puede contener una o más expresiones separadas por comas. Una expresión puede ser:

- Un nombre de columna. Por ejemplo:

nombre
dni
prestamo

- Un valor constante. Por ejemplo:

3
'El nombre es:'

- Una operación compleja. Por ejemplo:

devuelto - prestamo
'El nombre es:' + nombre
*((salario + prima) * 1,25) / 12*

El Lenguaje SQL

Ejemplos

- Nombre y dirección de los usuarios de la biblioteca como una frase [Consulta 02]

```
SELECT nombre + ' vive en ' + direccion  
FROM usuario
```

- Tiempo que ha estado prestado cada libro a cada usuario [Consulta 03]

```
SELECT codigodelibro, codigodeusuario, devuelto - prestamo  
FROM prestar
```

El Lenguaje SQL

El modelo relacional admite el valor nulo

- Un valor nulo representa un valor desconocido
- El resultado de cualquier operación en la que esté involucrado un valor nulo es otro valor nulo. Por ejemplo:
 - $5 - \text{nulo} = \text{nulo}$
 - $3 > 2 = \text{verdad}$, pero $3 > \text{nulo} = \text{nulo}$
- En el resultado de la consulta anterior se aprecia este efecto. Si un libro no ha sido devuelto (atributo devuelto = nulo), el número de días prestado es otro valor nulo

El Lenguaje SQL

El modificador DISTINCT permite eliminar filas repetidas del resultado

- Libros existentes en la biblioteca con sus autores [Consulta 04]:

```
SELECT titulo, autor  
FROM libro
```

- Si hay varios ejemplares del libro, aparece repetido. De esta manera no aparecen [Consulta 05]:

```
SELECT DISTINCT titulo, autor  
FROM libro
```

El Lenguaje SQL

La versión abreviada del SELECT permite no tener que escribir todos los nombres de las columnas

- Obtener todos los datos de los usuarios [Consulta 06]

```
SELECT codigo, nombre, direccion, dni, tipodetipousuario  
FROM usuario
```

- La forma abreviada es la siguiente [Consulta 07]

```
SELECT *  
FROM usuario
```


El Lenguaje SQL

Para ordenar los resultados se usa la línea ORDER BY:

- Título de los libros ordenados alfabéticamente [Consulta 08]

```
SELECT DISTINCT titulo  
FROM libro  
ORDER BY titulo
```

- Se ordena en el orden natural:
 - Los textos, alfabéticamente
 - Los números, de menor a mayor
 - Las fechas, de más antigua a más reciente

El Lenguaje SQL

Para ordenar de forma inversa, hay que añadir el modificador DESC

- Prestamos realizados con el más reciente primero [Consulta 09]

```
SELECT codigodelibro, codigodeusuario, prestamo, devuelto  
FROM prestar  
ORDER BY prestamo DESC
```

- Se puede ordenar por varias columnas a la vez. Se ordena primero por la que aparece primero, y para valores iguales, por la segunda columna, y luego por la tercera

El Lenguaje SQL

Se puede ordenar también por el resultado de evaluar expresiones

- Tiempo que ha estado prestado cada libro a cada usuario, ordenado por usuario y por tiempo (de mayor a menor) [Consulta 10]

```
SELECT codigodeusuario, codigodelibro, devuelto-prestamo  
FROM prestar  
ORDER BY codigodeusuario, devuelto-prestamo DESC
```

El Lenguaje SQL

La línea WHERE permite especificar una condición que deben cumplir las filas del resultado

- Se pueden utilizar operadores de comparación
- Se pueden utilizar operadores lógicos (booleanos)

condición1	condición2	condición1 AND condición2	condición1 OR condición2	NOT condición1
verdadero	verdadero	verdadero	verdadero	falso
verdadero	falso	falso	verdadero	falso
falso	verdadero	falso	verdadero	verdadero
falso	falso	falso	falso	verdadero
desconocido	verdadero	desconocido	desconocido	desconocido
desconocido	falso	desconocido	desconocido	desconocido
desconocido	desconocido	desconocido	desconocido	desconocido

El Lenguaje SQL

Ejemplos de filtrado de información

- Libros escritos por Pérez [Consulta 11]

```
SELECT DISTINCT titulo  
FROM libro  
WHERE autor = 'Pérez'
```

- Libros prestados al usuario 1 este año [Consulta 12]

```
SELECT codigodelibro  
FROM prestar  
WHERE codigodeusuario = 1 AND prestamo >= #1/1/2007#
```

El Lenguaje SQL

¿Cómo puedo saber si un valor es nulo?

- Libros no devueltos
- La siguiente consulta no funciona, devuelto = null siempre es nulo

```
SELECT codigodelibro  
FROM prestar  
WHERE devuelto = null
```

- Esta sí que funciona [Consulta 13]

```
SELECT codigodelibro, codigodeusuario, prestamo, devuelto  
FROM prestar  
WHERE devuelto IS NULL
```

El Lenguaje SQL

Otro ejemplo de filtrado de información

- Libros devueltos [Consulta 14]

```
SELECT codigodelibro, codigodeusuario, prestamo, devuelto  
FROM prestar  
WHERE devuelto IS NOT NULL
```

```
SELECT codigodelibro, codigodeusuario, prestamo, devuelto  
FROM prestar  
WHERE NOT devuelto IS NULL
```

- Cualquiera de las dos vale, el resultado es el mismo

El Lenguaje SQL

Consultas sobre varias tablas

- Queremos combinar tuplas de dos o más tablas
- Se hace indicando los nombres de todas las tablas en el FROM
- Sin embargo, ¡SQL no sabe como combinarlas!
 - No aplica el concepto de clave foránea
 - Realiza el producto cartesiano de las dos tablas
 - El resultado es una nueva tabla con todas las combinaciones
 - Ejemplo en Consulta 15

```
SELECT *  
FROM usuario, tipousuario
```


El Lenguaje SQL

Consultas sobre varias tablas

- La solución, incluir las condiciones de la combinación en la línea WHERE
 - Para cada clave foránea, una condición del estilo
claveprincipal = claveforanea
 - Se puede indicar la tabla a la que pertenece un atributo con la notación
nombredetabla.nombredatributo
 - Ejemplo: Días de préstamo para cada usuario [Consulta 16]

```
SELECT nombre, dias
FROM usuario, tipousuario
WHERE usuario.tipodetipodeusuario = tipodeusuario.tipo
```

El Lenguaje SQL

Consultas sobre varias tablas

- Ejemplo: Nombre de los libros no devueltos y persona que los tiene
[Consulta 17]

```
SELECT nombre, titulo, prestar, devuelto  
FROM usuario, prestar, libro  
WHERE usuario.codigo = prestar.codigodeusuario  
AND libro.codigo = prestar.codigodelibro  
AND prestar.devuelto IS NULL
```

El Lenguaje SQL

Búsquedas en texto completo

- Para buscar en el contenido de un texto se usa la operación LIKE en la línea WHERE
- Sintáxis: atributo LIKE “valor buscado”
- El valor buscado es un texto con las siguientes reglas:
 - El texto se busca literalmente
 - Si se usa el carácter ? se sustituye por cualquier carácter
 - Si se usa el carácter * se sustituye por cualquier conjunto de caracteres
- Ejemplos:
 - *autor* LIKE “Pérez” encuentra exactamente Pérez
 - *autor* LIKE “P?erez” encuentra Perez y Pérez y Porez y ...
 - *autor* LIKE “P*” encuentra cualquiera que empiece por P

El Lenguaje SQL

Búsquedas en texto completo

- Ejemplo: Libros en cuyo título aparezca XML o HTML [Consulta 18]

```
SELECT titulo  
FROM libro  
WHERE titulo LIKE "*XML*" OR titulo LIKE "*HTML*"
```

El Lenguaje SQL

Agrupación de resultados

- Las consultas realizadas hasta ahora no permiten calcular datos de resumen
- Por ejemplo:
 - Número de libros que hay en la biblioteca
 - Número de copias que hay sin prestar del libro XML
 - Tiempo total que el libro XML ha estado fuera de la biblioteca
- Para realizar esta consulta se utiliza las funciones de agrupación
- Son funciones que agrupan filas del resultado y calculan un único número
- Las más importantes son `count()`, `sum()`, `avg()`, ...

El Lenguaje SQL

Agrupación de resultados

- Ejemplo: Número de libros que hay en la biblioteca [Consulta 19]

```
SELECT count(codigo)  
FROM libro
```

- Número de copias prestadas del libro XML [Consulta 20]

```
SELECT count(libro.codigo)  
FROM libro, prestar  
WHERE libro.codigo = prestar.codigodelibro  
AND libro.titulo = "XML"  
AND prestar.devuelto IS NULL
```

El Lenguaje SQL

Agrupación de resultados

- Ejemplo: Tiempo total que el libro XML ha estado fuera de la biblioteca [Consulta 21]

```
SELECT sum(devuelto-prestamo)  
FROM libro, prestar  
WHERE libro.codigo = prestar.codigodelibro  
AND libro.titulo = "XML"
```

El Lenguaje SQL

Tratamiento de valores nulos

- La función *nz* convierte un nulo en otro valor
- La sintaxis es la siguiente:
nz(nombre_columna, nuevo_valor)
- Ejemplo: Tiempo que ha estado prestado cada libro a cada usuario
[Consulta 22]
- Utilizamos la función *now()* que devuelve la fecha actual

```
SELECT codigodelibro, codigodeusuario, nz(devuelto, now()) -  
prestamo  
FROM prestar
```


El Lenguaje SQL

Tratamiento de valores nulos

- Ejemplo: Tiempo total que el libro XML ha estado fuera de la biblioteca [Consulta 23]

```
SELECT sum(nz(devuelto, now())-prestamo)
FROM libro, prestar
WHERE libro.codigo = prestar.codigodelibro
AND libro.titulo = "XML"
```

Diseño de Bases de Datos

Bases de Datos Documentales

Grao en Información e Documentación

Curso 2013/2014

Miguel Ángel Rodríguez Luaces
Laboratorio de Bases de Datos
Universidade da Coruña