

Temario

- ♦ Introducción y fundamentos
- ♦ Introducción a SQL
- ♦ Modelo Entidad / Relación
- ♦ Modelo relacional
- ♦ Diseño relacional: formas normales
- ♦ Consultas
 - Cálculo relacional
 - Álgebra relacional
- ♦ Implementación de bases de datos
 - Estructura física: campos y registros
 - Indexación
 - Índices simples
 - Árboles B
 - Hashing

Un primer ejemplo informal

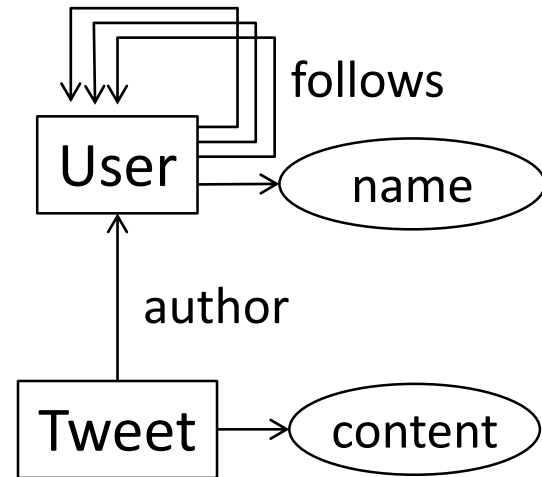
- ◆ Datos Twitter
 - Usuarios, relaciones follow, tweets...
- ◆ Consultas y operaciones
 - Ver timeline de un usuario, seguir a otro usuario, buscar tweets...

En C – Estructuras de datos

```
struct User {  
    char *name;  
    User *follows[5000];  
    int nfollows;  
};
```

```
struct Tweet {  
    char *content;  
    struct User *author;  
};
```

```
// Más alguna estructura para almacenar  
// todos los usuarios y tweets, etc.  
// ...
```



En C – Creación de datos

```
User *create_user (char *name) {
    User *u = (User*) malloc(sizeof (User*));
    u->name = name;
    u->nfollows = 0;
    return u;
}

void add_follows (User *u, User *v) {
    u->follows[u->nfollows++] = v;
}

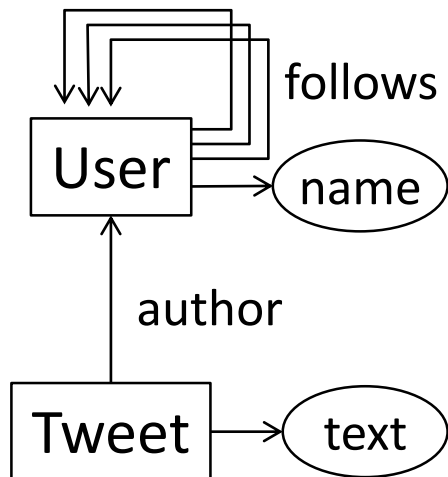
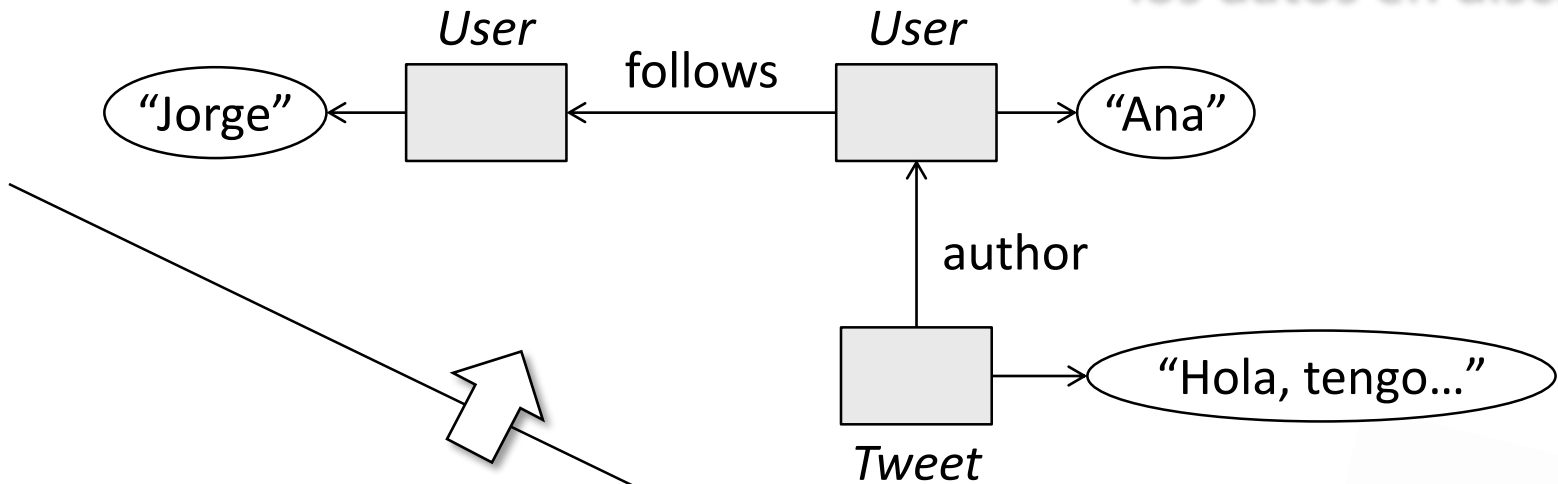
Tweet *create_tweet (User *u, char *text) {
    Tweet *t = (Tweet*) malloc(sizeof(Tweet*));
    t->content = text;
    t->author = u;
    return t;
}
```

En C – Creación de datos (cont)

```
void main () {  
    User *ana = create_user("Ana");  
    User *jorge = create_user("Jorge");  
    add_follows(ana, jorge);  
    create_tweet(ana, "Hola, tengo una cuenta en Twitter");  
    // ...  
}
```

Datos en RAM

¿Y si queremos almacenar los datos en disco?



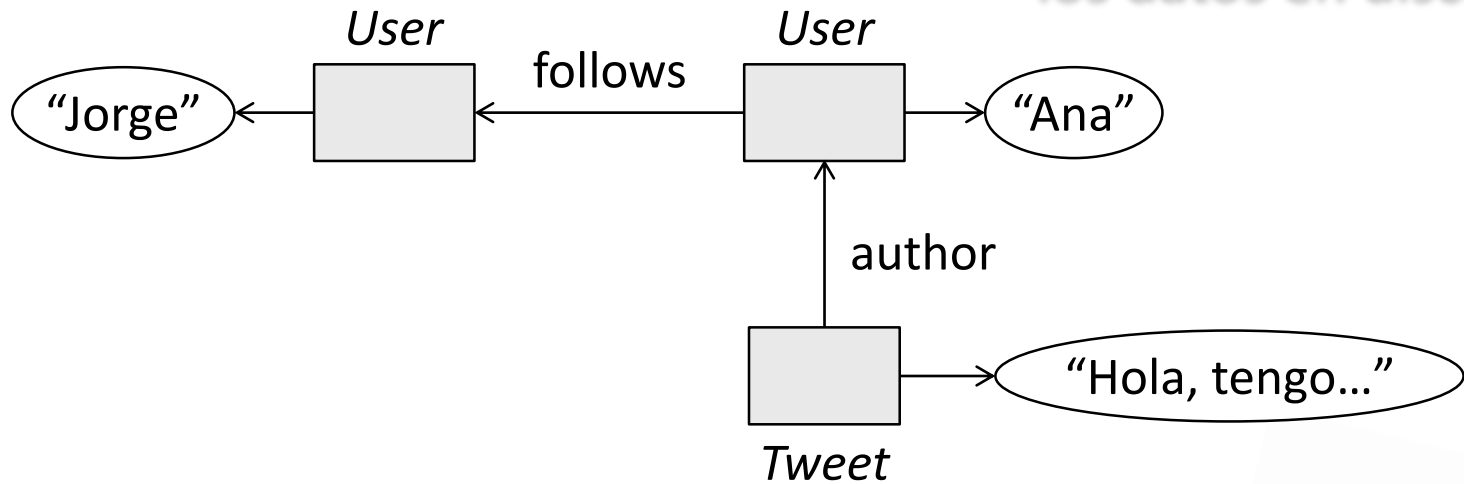
Datos (memoria)

Esquemas de datos
(diseño)

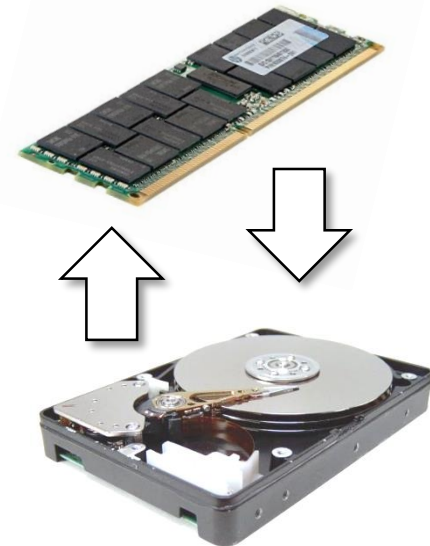


Datos en RAM

¿Y si queremos almacenar los datos en disco?



- ◆ Disco \Rightarrow Datos en ficheros
- ◆ ¿Qué estructura? \rightarrow struct \sim tablas!
- ◆ ¿Y los punteros?
Queremos poder recuperar las estructuras en RAM, incluso portar a otra máquina



En tablas – Estructuras de datos

¿Por qué tablas?

- ◆ Es una estructura muy general, se adapta bien a muchos dominios
- ◆ Se adapta bien a su almacenamiento físico
- ◆ Se adapta bien a nuestra forma de pensar

User

name	follows

←
“Registro”

Tweet

content	author

←
“Campo”



En tablas – Creación de datos

User

name	follows
"Ana"	?
"Jorge"	
...	

Tweet

content	author
"Hola, tengo..."	?
...	

¿Cómo se guardan físicamente los datos?

- ♦ En un fichero “uno detrás de otro” ☺
- ♦ Se necesita definir cómo separar los datos (filas y columnas)
- ♦ Se necesitan técnicas de acceso y actualización optimizados
- ♦ Lo estudiaremos...



En tablas – Creación de datos

¿Cómo almacenar “punteros”?

User

name	follows
“Ana”	?
“Jorge”	
...	

Tweet

content	author
“Hola, tengo...”	?
...	



En tablas – Creación de datos

¿Cómo almacenar “punteros”?

“Clave primaria”

name	follows
“Ana”	?
“Jorge”	
...	

En un caso real sería mejor utilizar un campo como DNI, o un ID interno

Tweet

content	author
“Hola, tengo...”	“Ana”
...	

“Clave externa”

Un campo juega el papel de puntero



En tablas – Creación de datos

¿Cómo almacenar “punteros”?

¿Y arrays de punteros?

User

name	follows
“Ana”	?
“Jorge”	
...	

Tweet

content	author
“Hola, tengo...”	“Ana”
...	



En tablas – Creación de datos

¿Cómo almacenar “punteros”?

¿Y arrays de punteros?

“Clave primaria”

name	follows
“Ana”	
“Jorge”	
...	

“Clave externa”

Follows

“Clave externa”

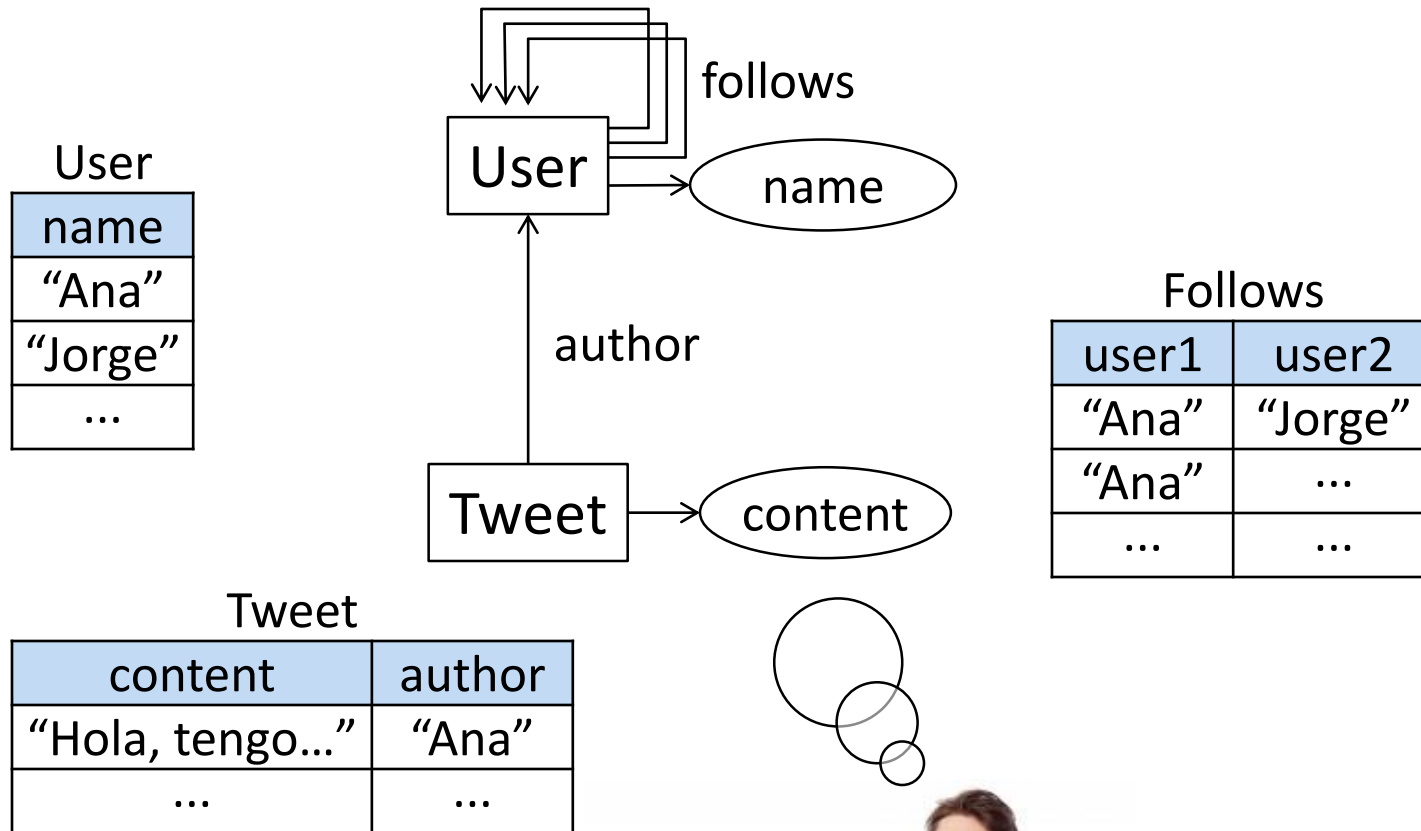
user1	user2
“Ana”	“Jorge”
“Ana”	...
...	...

Tweet

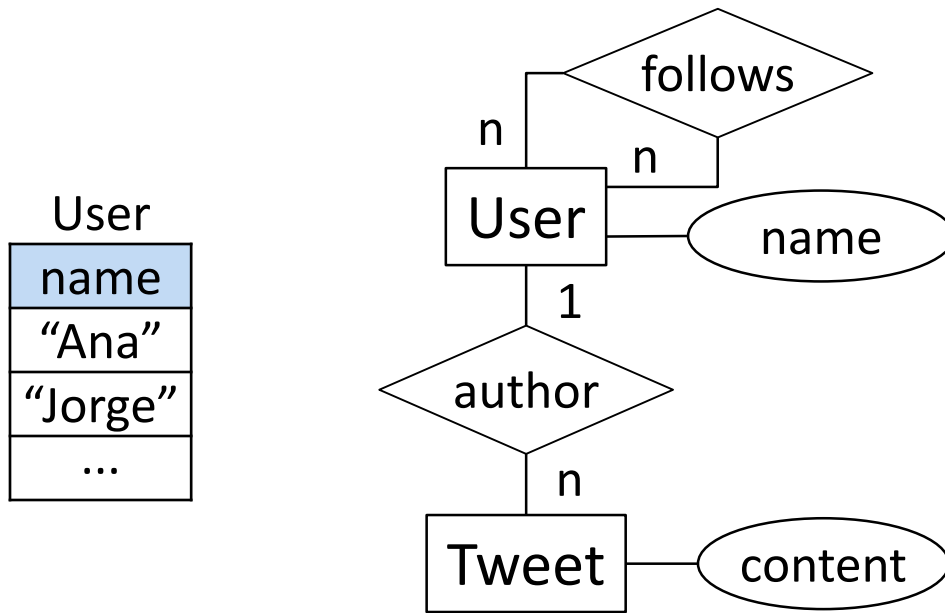
content	author
“Hola, tengo...”	“Ana”
...	



En tablas – Diseño de datos



En tablas – Diseño de datos

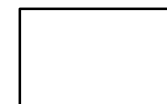


User
name
"Ana"
"Jorge"
...

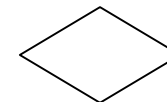
Follows	
user1	user2
"Ana"	"Jorge"
"Ana"	...
...	...

Tweet	
content	author
"Hola, tengo..."	"Ana"
...	...

Modelo ER



Entidad

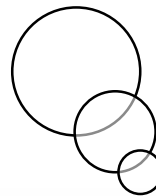


Relación



Propiedad

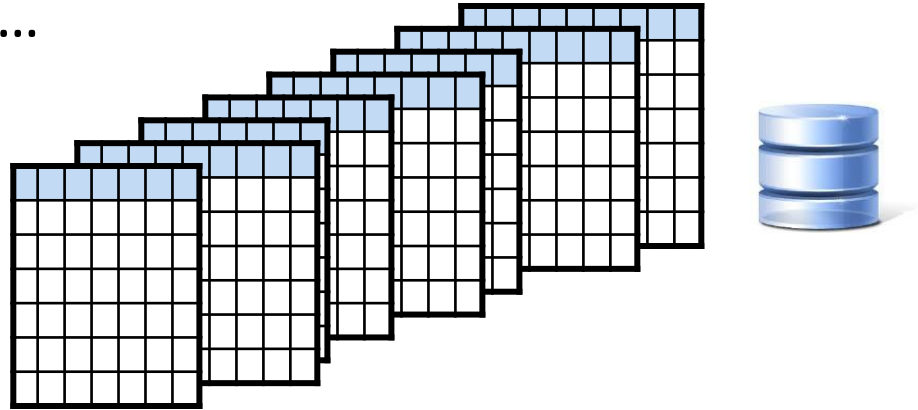
...



Bases de datos

Simplificando un poco...

Una **base de datos**
es un conjunto
de tablas en disco



¿Cómo implementamos y manejamos tablas en disco?

- ♦ ¿Lo implementamos desde cero? Normalmente no...
- ♦ Utilizamos un **Sistema de Gestión de Bases de Datos** (SGBD)
- ♦ Y el lenguaje **SQL**

SGBD y SQL

- ♦ SQL proporciona sintaxis para:
 - Definir estructuras de tablas
 - Introducir, eliminar y modificar datos en las tablas
 - Realizar consultas simples y complejas
- ♦ El SGBD proporciona:
 - Una interfaz de usuario para seleccionar, abrir, etc., bases de datos
 - Una interfaz para realizar operaciones interactivamente en lugar de utilizar SQL
 - Un motor de SQL

Ejemplo en PostgreSQL...

Ejemplos SQL

Creación de esquemas de tabla

```
create table TwitterUser ( -- "User" es reservada SQL
    dni varchar primary key, -- Pequeña variación...
    name varchar
);

create table Follows (
    user1 varchar references TwitterUser(dni),
    user2 varchar references TwitterUser(dni)
);

create table Tweet (
    author varchar references TwitterUser(dni),
    content text
);
```

Ejemplos SQL – Inserción de datos

```
-- Inserción de datos en las tres tablas
```

```
insert into TwitterUser values
```

```
    ('123', 'Ana'),
```

```
    ('456', 'Jorge');
```

```
insert into Follows values
```

```
    ('123', '456');
```

```
insert into Tweet values
```

```
    ('123', 'Hola, tengo una cuenta en Facebook');
```

Ejemplos SQL – Consultas

```
-- Usuarios que se llaman "Ana"
```

```
select dni, name from TwitterUser  
where name = 'Ana'
```

```
-- Tweets escritos por usuarios que se llaman "Ana"
```

```
select content from Tweet, TwitterUser  
where author = dni and name = 'Ana'
```

```
-- Usuario con más seguidores
```

```
select name, count(dni) as nfollowers  
from TwitterUser, Follows  
where dni = user2  
group by dni, name  
order by nfollowers desc limit 1
```

En C sería algo como...

```
User *getUser (User *users[], int nusers, char *name) {
    for (int i = 0; i < nusers; i++)
        if (strcpy(users[i]->name, name) == 0)
            return users[i];
    return NULL;
}

User *mostfollowed (User *users[], int nusers) {
    int nfollowers[nusers], max = 0;
    for (int i = 0; i < nusers; i++)
        for (int j = 0; j < users[i]->nfollows; j++)
            if (++nfollowers[users[i]->follows[j]] > nfollowers[max])
                max = users[i]->follows[j];
    return users[max];
}
```

Sistema de gestión de bases de datos (SGBD)

- ◆ Software de gestión y acceso a bases de datos
 - El desarrollo de un SGBD implica miles de personas/año
- ◆ Procesador de SQL
 - Motor de operaciones (consultas)
 - Optimizador de consultas
- ◆ Motor de almacenamiento físico
- ◆ Herramientas de administración
 - Creación y diseño de tablas, usuarios...
- ◆ API con C, Java, PHP...

Usuario final

Interfaz de usuario



Programador aplicación

Lógica de la aplicación

Sentencias SQL

ODBC, JDBC, PHP...

API BD

Almacenamiento
Consultas
Actualización

SGBD

Bases de datos

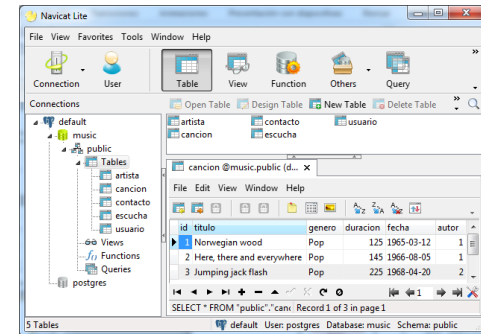


Arquitectura BD

Software aplicación

Administrador

Entorno / herramientas de administración



pgAdmin
Navicat
SQLYog
Workbench
etc.

Arquitectura ad hoc

Interfaz de usuario



Usuario final

Software aplicación

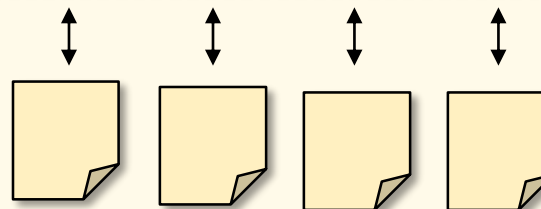


Programador aplicación

Lógica de la aplicación

Acceso y gestión de datos

Archivos de datos
Memoria externa



Roles en el uso de una base de datos

- ♦ Usuarios finales
 - Interactúan con aplicaciones que acceden a la BD
- ♦ Usuarios avanzados
 - Interactúan con la BD en SQL
- ♦ Programadores de aplicación
 - Interactúan con la BD escribiendo programas
- ♦ Diseñadores
 - Definen el diseño de la BD
- ♦ Administradores
 - Mantienen el diseño de la BD
 - Gestionan usuarios y permisos de acceso
 - Gestionan necesidades de actualización
- ♦ Desarrolladores de herramientas SGBD
 - Implementan la capa inferior de acceso físico a los datos
 - Desarrollan el software y herramientas que dan servicio a todo lo anterior

Otros niveles de abstracción

Además de SQL estudiaremos...

- ♦ Los formalismos en los que se basan las BDs
 - Modelo relacional (piezas básicas de diseño)
 - Formas normales (calidad de un diseño)
 - Álgebra y cálculo relacional (para construir consultas)
- ♦ Técnicas de implementación internas a un SGBD

Breve perspectiva temporal

1960's	Primeras nociones de bases de datos
1970	Propuesta del modelo relacional (E. F. Codd, CACM)
1974	Primer SGBD en el MIT (RDMS) SQL en IBM (D. D. Chamberlin & R. F. Boyce)
1976	Modelo Entidad / Relación
1979	Oracle
1980	dBase II
1983	IBM DB2
1984	FoxPro
Mediados 80's	Despliegue de la tecnología BD
1987	SAP Sybase
1989	MS SQL Server
1992	MS Access
1994	MySQL
1995	PostgreSQL
Mediados 90's	BDs orientadas a objetos
2000's	BDs XML, BD distribuidas, Big Data

Un segundo ejemplo

Aplicación de música online con red social

- ◆ Tipos de datos: usuarios, canciones, discos, autores...
- ◆ Estructuras:
 - Los usuarios tienen nick, nombre, email...
 - Las canciones tienen título, género, duración, fecha...
 - Los artistas tienen nombre, nacionalidad...
- ◆ Relaciones:
 - Las canciones tienen autores, los discos tienen canciones, los usuarios tienen amigos, discos favoritos, escuchan canciones...
- ◆ Funcionalidades:
 - Buscar una canción, escucharla, ver sus datos...
 - Ver / añadir amigos...

Diseño de las estructuras de datos

- ♦ Archivos?
- ♦ Sentencias SQL?
- ♦ Tablas?
- ♦ Diagramas?

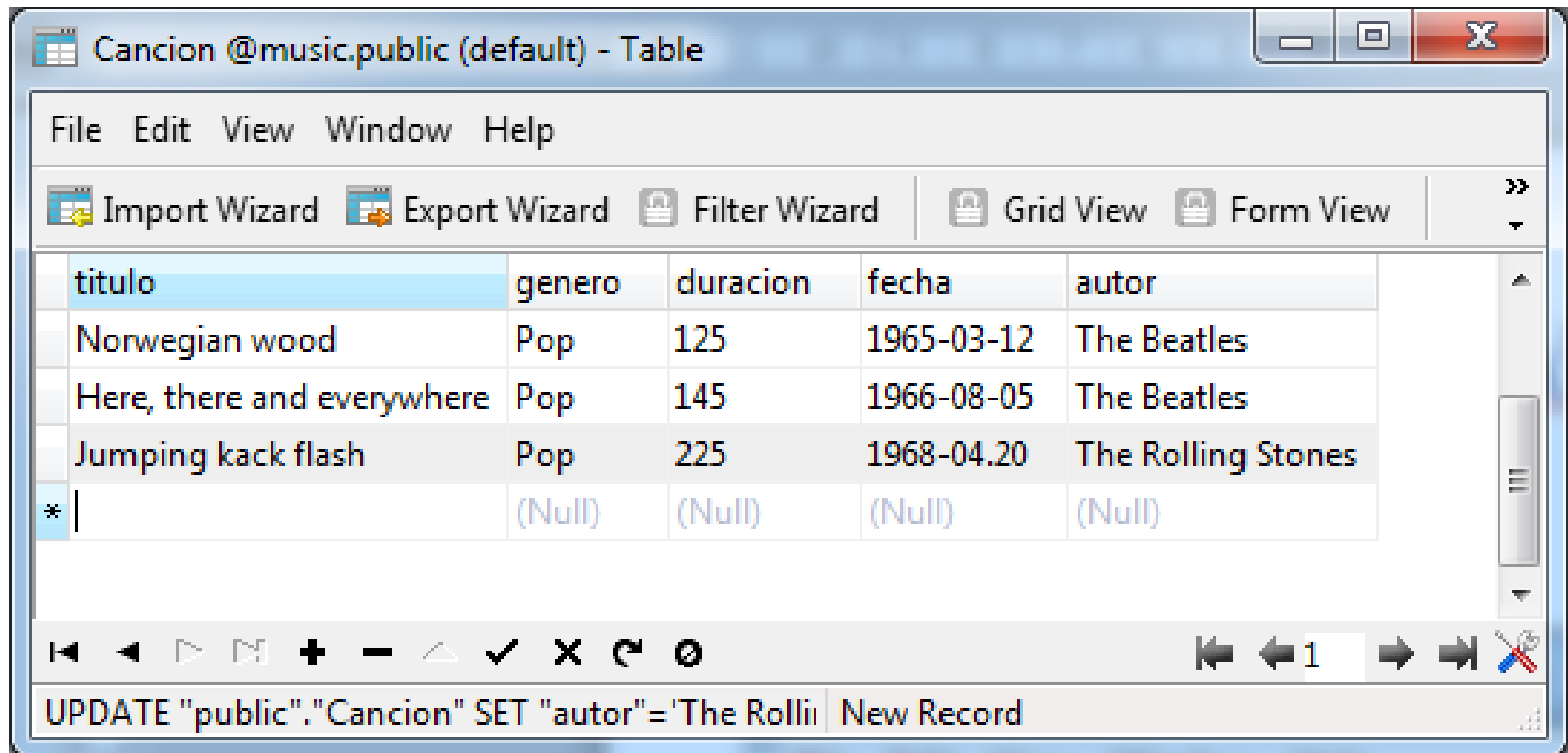
Datos en archivos

```
----- /* Usuarios */
lola      Dolores      lola@gmail.com
pepe      José         jose@gmail.com
chema     José María   chema@gmail.com
charo     Rosario      rosario@gmail.com
----- /* Relaciones sociales */
pepe      lola
charo     pepe
chema     charo
----- /* Artistas */
The Beatles      UK
The Rolling Stones  UK
----- /* Canciones */
Norwegian wood      Pop      125      1965-03-12      The Beatles
Here, there and everywhere  Pop      145      1966-08-05      The Beatles
Jumping jack flash   Pop      225      1968-04-20      The Rolling Stones
----- /* Escuchas */
charo      2011-09-09 16:57:54 /* canción...? */
pepe       2011-09-12 21:15:30 /* canción...? */
```

...

???

Tablas en un SGBD



Cancion @music.public (default) - Table

File Edit View Window Help

Import Wizard Export Wizard Filter Wizard Grid View Form View

titulo	genero	duracion	fecha	autor
Norwegian wood	Pop	125	1965-03-12	The Beatles
Here, there and everywhere	Pop	145	1966-08-05	The Beatles
Jumping kack flash	Pop	225	1968-04.20	The Rolling Stones
*	(Null)	(Null)	(Null)	(Null)

UPDATE "public"."Cancion" SET "autor"='The Rollin' New Record

Tablas...

PROYECTO

NombreProyecto	NumProyecto	UbicacionProyecto	NumDptoProyecto
ProductoX	1	Valencia	5
ProductoY	2	Sevilla	5
ProductoZ	3	Madrid	5
Computación	10	Gijón	4
Reorganización	20	Madrid	1
Comunicaciones	30	Gijón	4

SUBORDINADO

DniEmpleado	NombSubordinado	Sexo	FechaNac	Relación
333445555	Alicia	M	05-04-1986	Hija
333445555	Teodoro	H	25-10-1983	Hijo
333445555	Luisa	M	03-05-1958	Esposa
987654321	Alfonso	H	28-02-1942	Esposo
123456789	Miguel	H	04-01-1988	Hijo
123456789	Alicia	M	30-12-1988	Hija
123456789	Elisa	M	05-05-1967	Esposa

EMPLEADO

Nombre	Apellido1	Apellido2	Dni	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
José	Pérez	Pérez	123456789	01-09-1965	Eloy I, 98	H	30000	333445555	5
Alberto	Campos	Sastre	333445555	08-12-1955	Avda. Ríos, 9	H	40000	888665555	5
Alicia	Jiménez	Celaya	999887777	12-05-1968	Gran Vía, 38	M	25000	987654321	4
Juana	Sainz	Oreja	987654321	20-06-1941	Cerquillas, 67	M	43000	888665555	4
Fernando	Ojeda	Ordóñez	666884444	15-09-1962	Portillo, s/n	H	38000	333445555	5
Aurora	Oliva	Avezuela	453453453	31-07-1972	Antón, 6	M	25000	333445555	5
Luis	Pajares	Morera	987987987	29-03-1969	Enebras, 90	H	25000	987654321	4
Eduardo	Ochoa	Paredes	888665555	10-11-1937	Las Peñas, 1	H	55000	NULL	1

DEPARTAMENTO

NombreDpto	NumeroDpto	DniDirector	FechaIngresoDirector
Investigación	5	333445555	22-05-1988
Administración	4	987654321	01-01-1995
Sede Central	1	888665555	19-06-1981

LOCALIZACIONES_DPTO

NumeroDpto	UbicacionDpto
1	Madrid
4	Gijón
5	Valencia
5	Sevilla
5	Madrid

TRABAJA_EN

DniEmpleado	NumProy	Horas
123456789	1	32,5
123456789	2	7,5
666884444	3	40,0
453453453	1	20,0
453453453	2	20,0
333445555	2	10,0
333445555	3	10,0
333445555	10	10,0
333445555	20	10,0
999887777	30	30,0
999887777	10	10,0
987987987	10	35,0
987987987	30	5,0
987654321	30	20,0
987654321	20	15,0
888665555	20	NULL

Esquemas de tablas

EMPLEADO

Nombre	Apellido1	Apellido2	<u>Dni</u>	FechaNac	Dirección	Sexo	Sueldo	SuperDni	Dno
--------	-----------	-----------	------------	----------	-----------	------	--------	----------	-----

DEPARTAMENTO

NombreDpto	<u>NumeroDpto</u>	DniDirector	FechaIngresoDirector
------------	-------------------	-------------	----------------------

LOCALIZACIONES_DPTO

<u>NumeroDpto</u>	<u>UbicacionDpto</u>
-------------------	----------------------

PROYECTO

NombreProyecto	<u>NumProyecto</u>	UbicacionProyecto	NumDptoProyecto
----------------	--------------------	-------------------	-----------------

TRABAJA_EN

<u>DniEmpleado</u>	<u>NumProy</u>	Horas
--------------------	----------------	-------

SUBORDINADO

<u>DniEmpleado</u>	<u>NombSubordinado</u>	Sexo	FechaNac	Relación
--------------------	------------------------	------	----------	----------

Sentencias SQL

```
CREATE TABLE Usuario (  
    nick varchar(30) PRIMARY KEY,  
    nombre text NOT NULL,  
    email text NOT NULL UNIQUE  
);
```

```
CREATE TABLE Contacto (  
    usuario1 varchar(30) NOT NULL REFERENCES Usuario (nick),  
    usuario2 varchar(30) NOT NULL REFERENCES Usuario (nick),  
    PRIMARY KEY (usuario1,usuario2)  
);
```

```
CREATE TABLE Artista (  
    id int PRIMARY KEY,  
    nombre text NOT NULL,  
    nacionalidad text  
);
```

```
CREATE TABLE Cancion (  
    id int PRIMARY KEY,  
    titulo text NOT NULL,  
    genero text,  
    duracion int,  
    fecha date,  
    autor int NOT NULL REFERENCES Artista (id)  
);
```

...

```
CREATE TABLE Escucha (  
    usuario varchar(30) REFERENCES Usuario (nick),  
    cancion int REFERENCES Cancion (id),  
    instante timestamp,  
    PRIMARY KEY (usuario,cancion,instante)  
);
```

Sentencias SQL

...

```
INSERT INTO Usuario VALUES ('lola', 'Dolores', 'lola@gmail.com');
INSERT INTO Usuario VALUES ('pepe', 'José', 'jose@gmail.com');
INSERT INTO Usuario VALUES ('chema', 'José María', 'chema@gmail.com');
INSERT INTO Usuario VALUES ('charo', 'Rosario', 'rosario@gmail.com');

INSERT INTO Contacto VALUES ('pepe', 'lola');
INSERT INTO Contacto VALUES ('charo', 'pepe');
INSERT INTO Contacto VALUES ('chema', 'charo');

INSERT INTO Artista VALUES (1, 'The Beatles', 'UK');
INSERT INTO Artista VALUES (2, 'The Rolling Stones', 'UK');

INSERT INTO Cancion VALUES (1, 'Norwegian wood', 'Pop', '125', '1965-03-12', 1);
INSERT INTO Cancion VALUES (2, 'Here, there and everywhere', 'Pop', '145', '1966-08-05', 1);
INSERT INTO Cancion VALUES (3, 'Jumping jack flash', 'Pop', '225', '1968-04-20', 2);

INSERT INTO Escucha VALUES ('charo', 2, '2011-09-09 16:57:54');
INSERT INTO Escucha VALUES ('pepe', 3, '2011-09-12 21:15:30');
```

Diagramas Entidad / Relación

