

# 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, B\* y B+

# Un primer ejemplo informal

- ◆ Banco
  - Clientes, cuentas, movimientos...
- ◆ Consultas y operaciones
  - Consultar movimiento, totales mensuales, número de clientes con ingresos superiores a ...

## *Cliente*

uid	Nombre	Email
1	Carlos	carlos@gg.com
2	Pedro	pedro@hh.com
3	Laura	laura@ms.com

## *Cuenta Corriente*

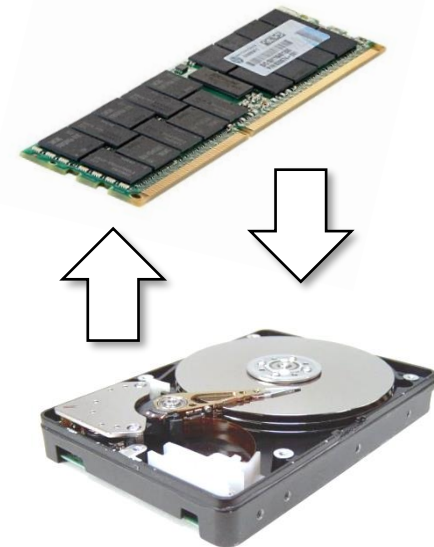
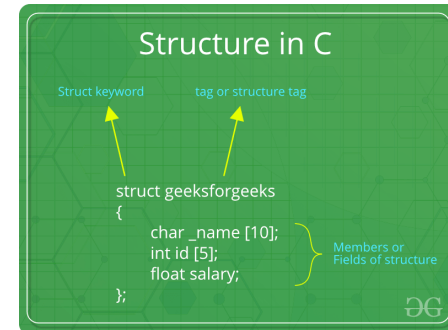
NumeroCC	uid_principal	uid_autorizado	Tipo	Fecha Apertura
0000000000	1		1	2020/03/05
0000000001	1	3	1	2020/05/07
0000000002	2		2	1990/04/26

## *Movimiento*

NumeroCC	uid	Fecha y hora	Importe	Tipo Mov.
0000000000	1	2020/05/01 11:10:00	1000,00	1
0000000000	1	2020/07/02 23:05:20	1.667,00	2
0000000002	2	2019/02/25 15:05:20	500,00	3

# Implementación en C

- ◆ Definición de estructuras de datos
- ◆ Punteros que apuntan a estructuras
- ◆ Funciones de lectura y escritura
- ◆ Funciones de búsqueda de información
- ◆ Programas ad hoc para manejo de la información



# 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

**Entidad/relación/  
esquema de relación**

*Cliente*

uid	Nombre	Email

*Registro/dupla/fila*

*Propiedad/atributo/columna*



# Tablas – Creación de datos

¿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...

*Cliente*

uid	Nombre	Email
1	Carlos	carlos@gg.com
2	Pedro	pedro@hh.com
3	Laura	laura@ms.com



# Tablas – Creación de datos

¿Cómo almacenar “punteros”?

*“Clave primaria”*

*Cliente*

uid	Nombre	Email
1	Carlos	carlos@gg.com
2	Pedro	pedro@hh.com
3	Laura	laura@ms.com

*Un campo juega el papel de puntero*

*“Clave primaria”*

*Cuenta*

NumeroCC	uidTitular	uidAutorizado	Tipo	FechaApertura
000000000	1		1	2020/03/05
000000001	1	3	1	2020/05/07
000000002	2		2	1990/04/26

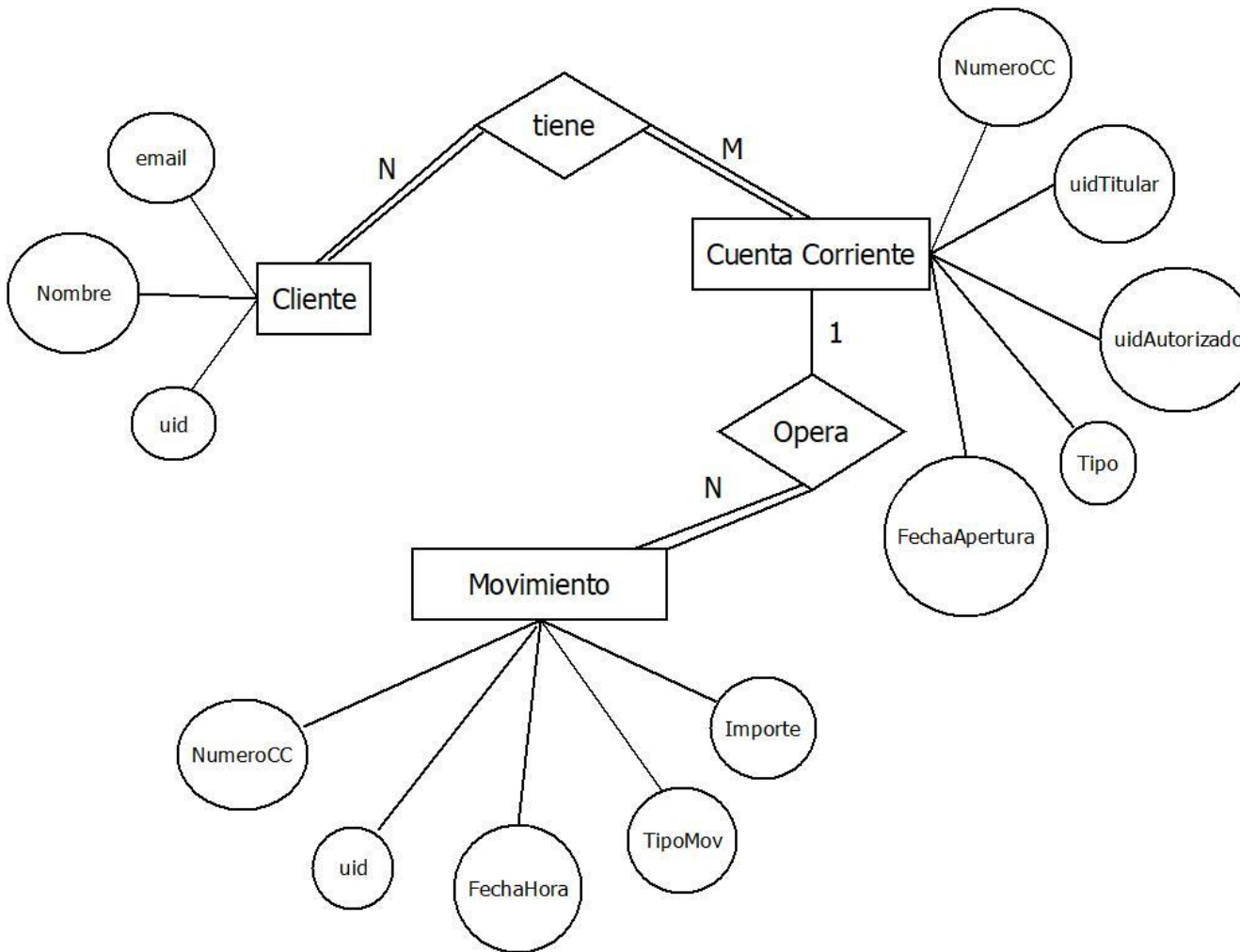
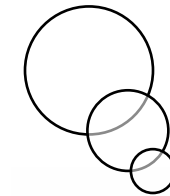
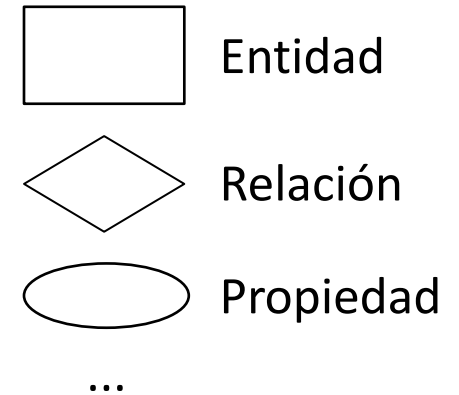
*“Claves externa”*

*Movimiento*

NumeroCC	uid	FechaHora	Importe	TipoMov
000000000	1	2020/05/01 11:10:00	1000,00	1
000000000	1	2020/07/02 23:05:20	1.667,50	2
000000002	2	2019/02/25 15:05:20	500,00	3

# Tablas – Diseño de datos

## Modelo ER

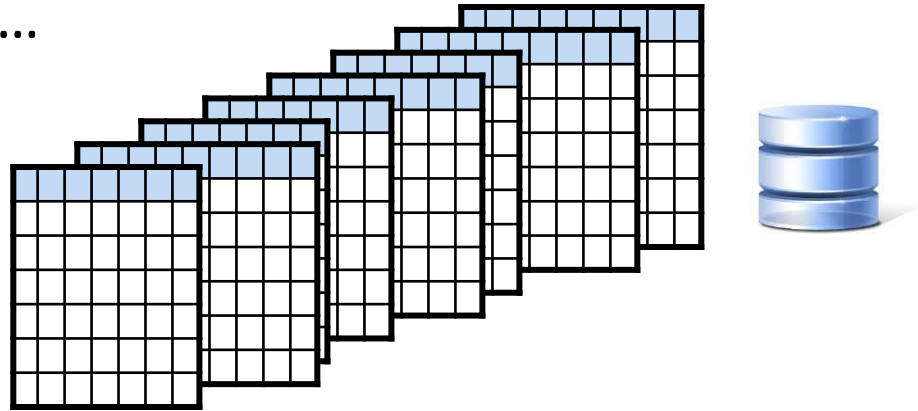




# Bases de datos

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**

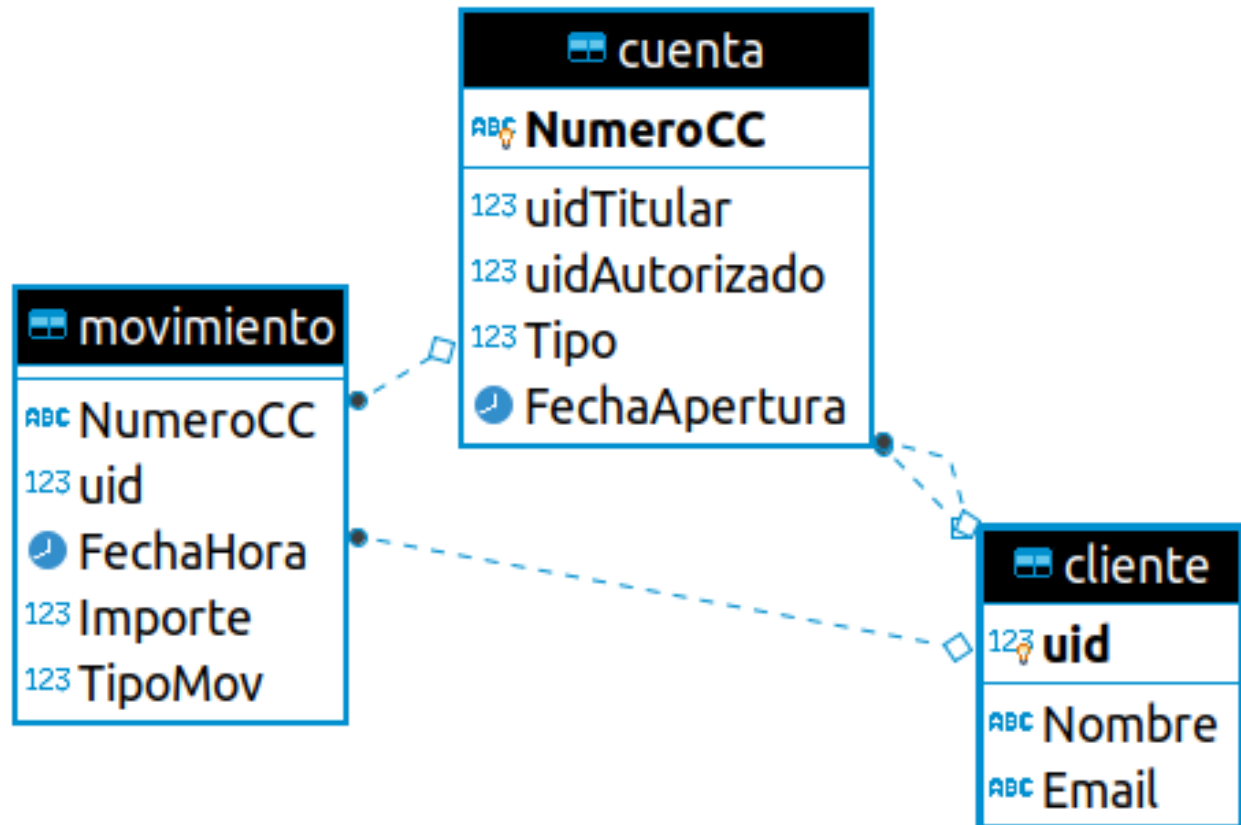
# SGBD y SQL

- ♦ 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 e inserción de datos

script: 1-banco.sql



# Ejemplos SQL – Consultas

- ♦ Movimientos de mas de 1.000€

```
SELECT
    *
FROM
    movimiento as m
WHERE m."Importe" >1000
```

- ♦ Nombre e email de los clientes con movimientos de mas de 1.000€

```
SELECT
    c."Nombre", c."Email"
FROM
    movimiento m JOIN cliente c ON m.uid=c.uid
WHERE m."Importe" >1000
```

# En C sería algo como...



# 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...

# Arquitectura con BD

Usuario final

Interfaz de usuario



Programador aplicación

Lógica de la aplicación

Sentencias SQL

API BD

ODBC, JDBC, PHP...

Almacenamiento  
Consultas  
Actualización

Bases de datos

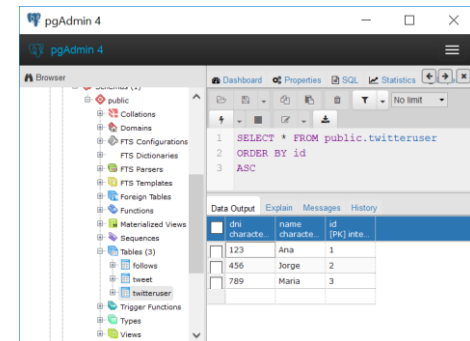
SGBD



Software aplicación

Administrador

Entorno / herramientas de administración



*pgAdmin  
dbeaver  
vs code  
etc.*

# Arquitectura ad hoc

Interfaz de usuario



Usuario final

Software aplicación

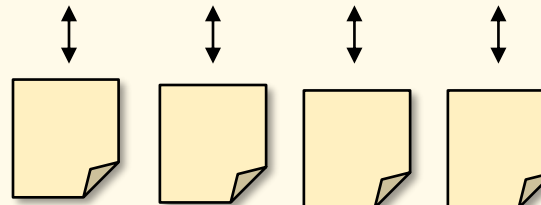


Programador aplicación

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
  - Índices
  - Árboles B

# 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
2010	BD no relacionales (mongoDB, elasticsearch)