Bases de Datos Relacionales y SQL: Una Introducción

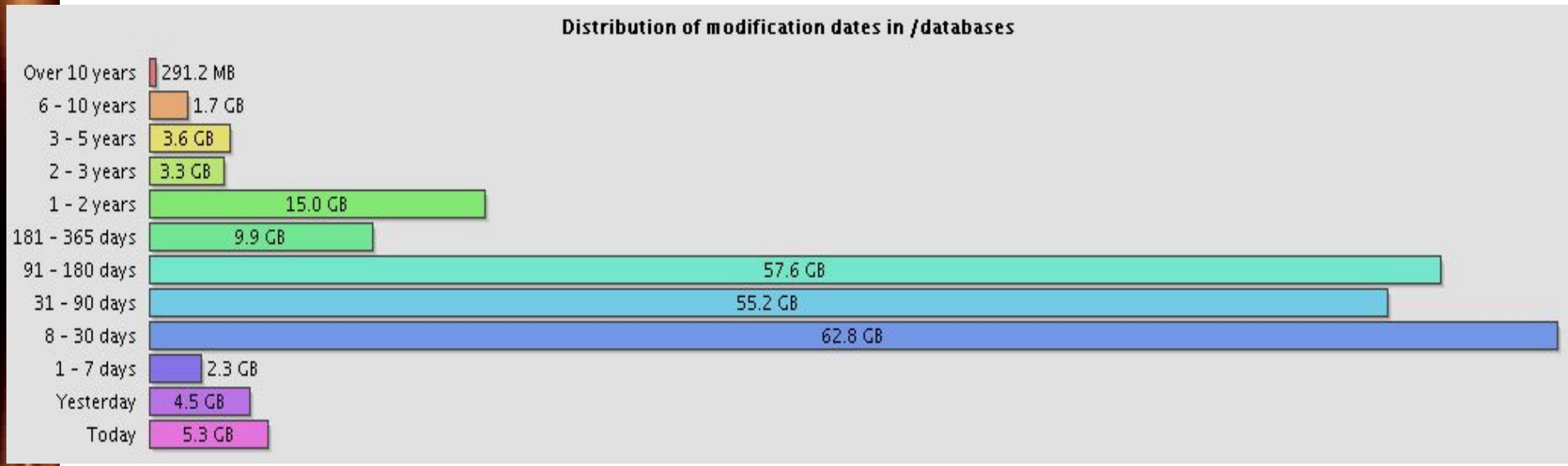
José María Fernández González

Protein Design Group, CNB – CSIC

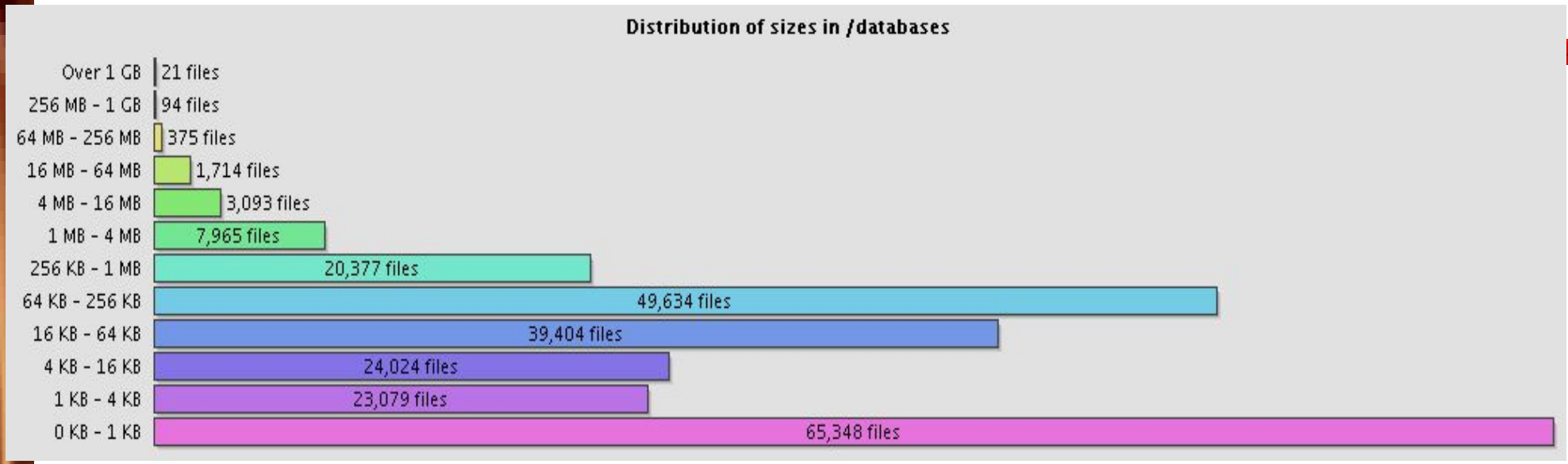
Sumario

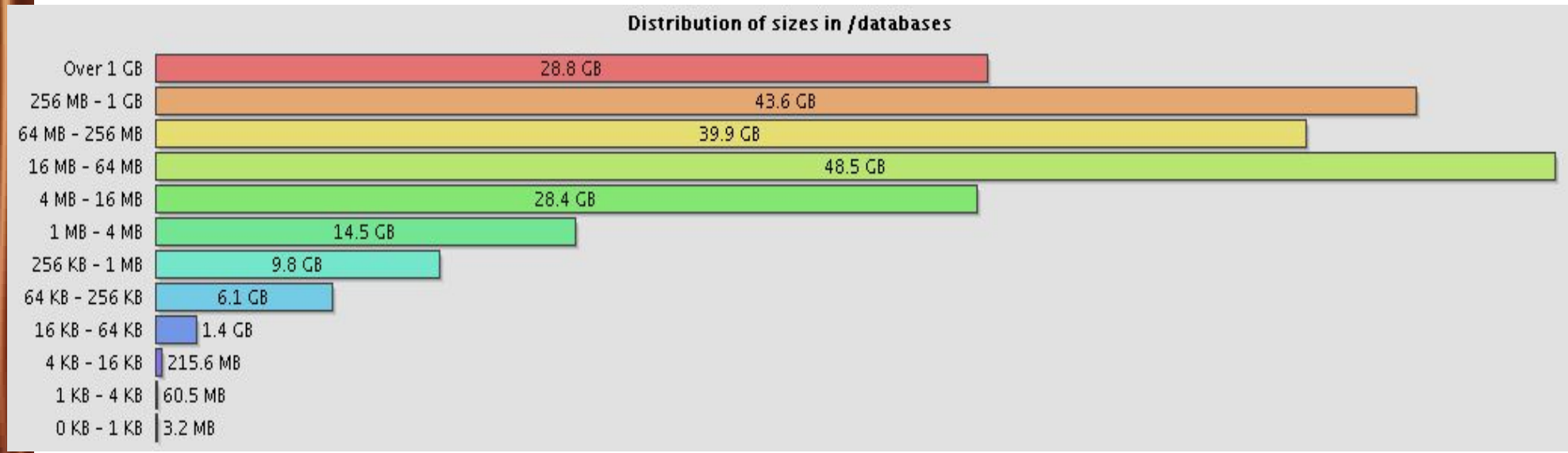
- ¿Por qué una base de datos relacional?
- ¿Qué es un SGBDR?
- Usuarios de base de datos
- Tablas: creación y definición de restricciones
- Manipulación de datos: consulta, inserción, actualización y borrado
- SQL (*Structured Query Language*)
- Pistas de cómo diseñar una BD
- Interfaces de programación

¿Por qué una base de datos relacional?

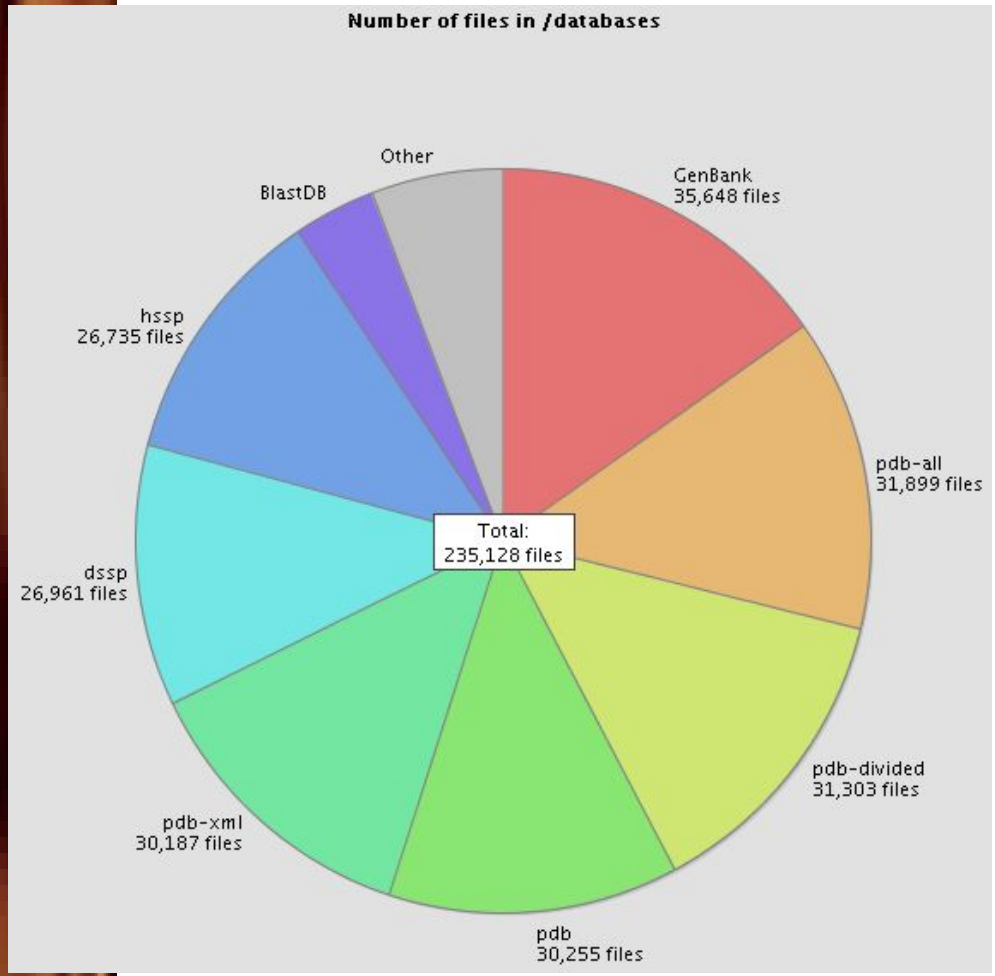


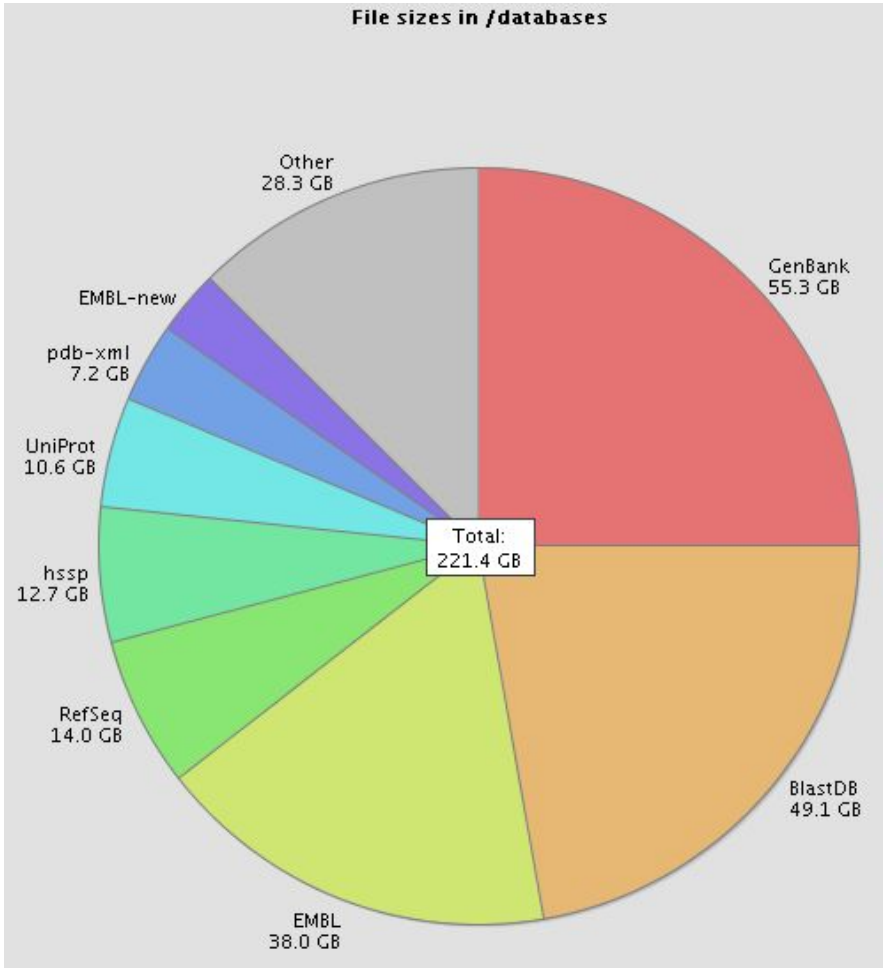
¡Qué grandes!
¿Cómo las consulto?





Demasiado grandes para un cutre-script, ¿no?





¿Qué es un SGDBR?

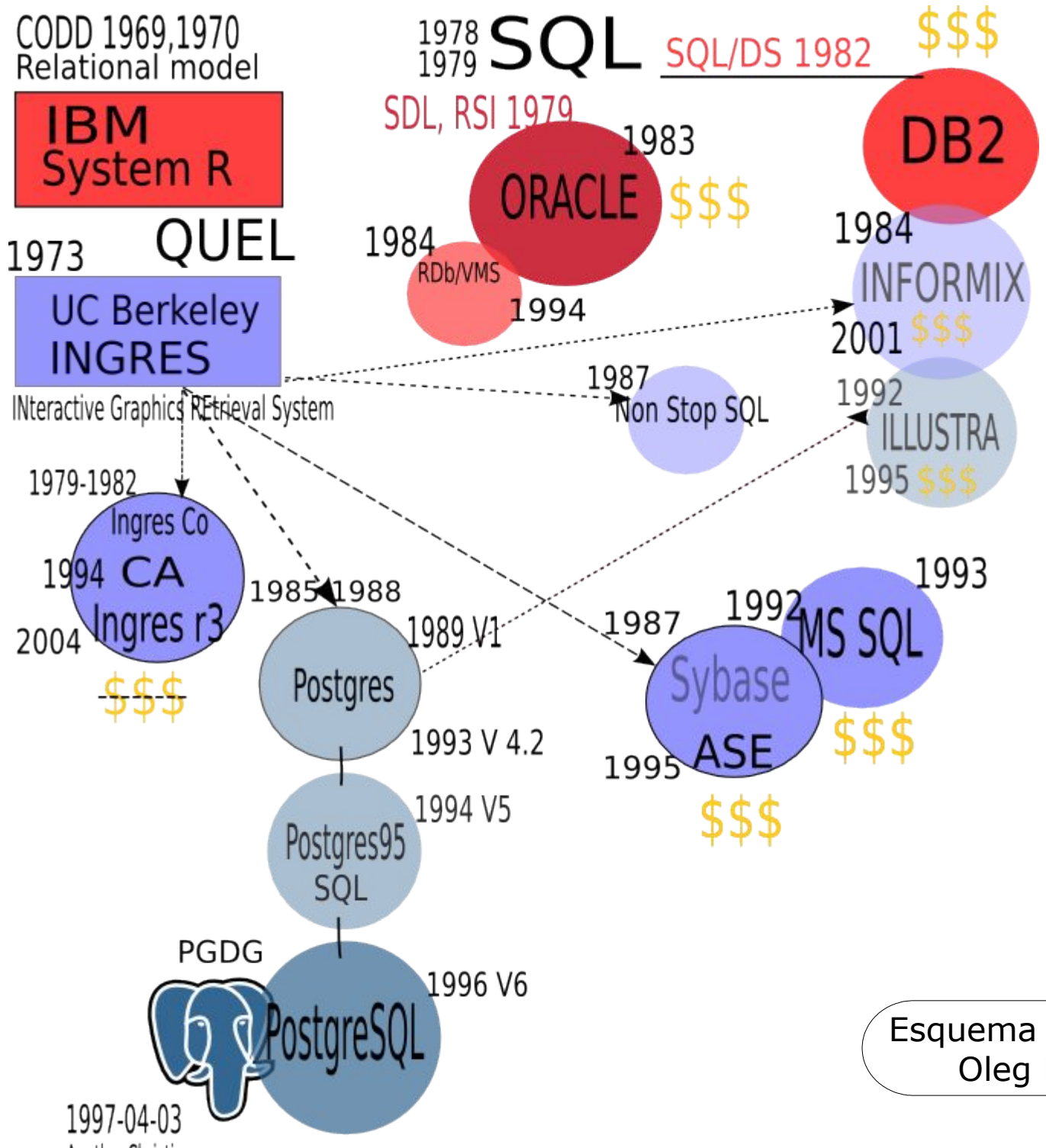
- Sistema Gestor de Base de Datos Relacional (SGDBR). Software que gestiona el uso de las bases de datos relacionales, y optimiza y controla el acceso al contenido de las mismas.
- El almacenamiento físico de los datos lo gestiona única y exclusivamente el gestor de la base de datos. El usuario sólo debe preocuparse de la estructura lógica de los mismos.
- La manipulación de la estructura y contenido de una base de datos relacional se realiza mediante el lenguaje de consultas SQL (*Structured Query Language*)

Structured Query Language

- Es el lenguaje estándar que se emplea para consultar y modificar bases de datos relacionales.
- SQL86, SQL89, SQL92, SQL99, SQL200X.
- Cada una de las versiones SQL clasifica las distintas operaciones por su complejidad: *entry*, *intermediate*, *advanced*.
- Por ello, podemos encontrar hoy en día gestores de bases de datos que implementan SQL89, algunas características de SQL92 y otras pocas de SQL99.

Algunos SGBDR

- Actualmente existen decenas de sistemas gestores de bases de datos relacionales.
- Sistemas *open source*: PostgreSQL, MySQL, ...
- Sistemas de pago: Oracle, Sybase, DB2, ...
- A pesar de que SQL es un estándar, cada gestor de bases de datos implementa sus propias extensiones, con lo cuál nos encontramos con un dialecto SQL por cada gestor de bases de datos.



Esquema realizado por
Oleg Bartunov

¿Cual es el mejor SGBDR?

- Si buscáis el mejor SGBDR, la respuesta corta es: **depende**.
- En Bioinformática, tanto PostgreSQL como MySQL se han vuelto muy populares. Ello es debido a que pueden ser instalados en prácticamente cualquier plataforma hardware sin ningún coste adicional (la licencia de uso es gratuita).
- El principal éxito de PostgreSQL: la simplicidad y muchas de las características y potencia de los SGBDR de pago.
- El principal éxito de MySQL: la gran simplicidad y su velocidad de acceso a tablas sencillas.

¿Que hay dentro de una BDR?

- Datos.
- Tablas+columnas.
- Tipos de datos. **INTEGER, VARCHAR, DATE, REAL**
- Usuarios.
- Restricciones.
- Índices y otros elementos.

Tablas: Introducción

- Una base de datos relacional está compuesta de varias tablas relacionadas entre sí.
- Cada tabla tiene un nombre, y está estructurada en una o más columnas.
- Una entrada de datos de una tabla es una tupla, y está compuesta por los valores asociados a cada columna de la tabla.
- En cada tupla, una columna puede tener asociado a lo sumo un valor.
- Una tabla puede tener una o más restricciones asociadas a la misma.

Ejemplo de tabla

Proteína		
AccNumber	Nombre	Descripción
P53905	lsm7_yeast	U6 snRNA-associated
P07260	if4e_yeast	Eukaryotic translat
P39517	dhh1_yeast	Putative ATP-depend
Q06819	dib1_yeast	DIB1 protein
P47017	lsm1_yeast	Sm-like protein LSm
P38203	lsm2_yeast	U6 snRNA-associated
P20053	pr04_yeast	U4/U6 small nuclear
P19735	pr06_yeast	Pre-mRNA splicing f
Q06217	smd2_yeast	Small nuclear ribon

Tablas: Columnas

- Cada columna tiene nombre, y un tipo de datos.
- Cada columna puede participar en una o varias restricciones.
- Las restricciones básicas de una columna son: de contenido nulo, de restricciones de contenido.
- Se puede asignar a una columna una expresión por omisión. Se emplea cuando se guarda una tupla en la que no se haya dado explícitamente un valor a esa columna.

Tablas: Tipos SQL

- INTEGER 3, -2
- CHAR 'S','KP'
- VARCHAR 'QLF'
- BOOLEAN TRUE
- TIMESTAMP Tue Apr 19 19:37:31 CEST 2005
- DATE 2005-04-22
- TIME 17:25:38
- NUMERIC 0.03156
- REAL 3,1415
- CLOBs

En un lugar de la Mancha de cuyo nombre no quier acordarme vivia un hidalgo de los de espada y rocin, conocidocomo

- BLOBs
- Etc...

Usuarios de una Base de Datos

- Los usuarios de una base de datos no están relacionados con los usuarios del sistema.

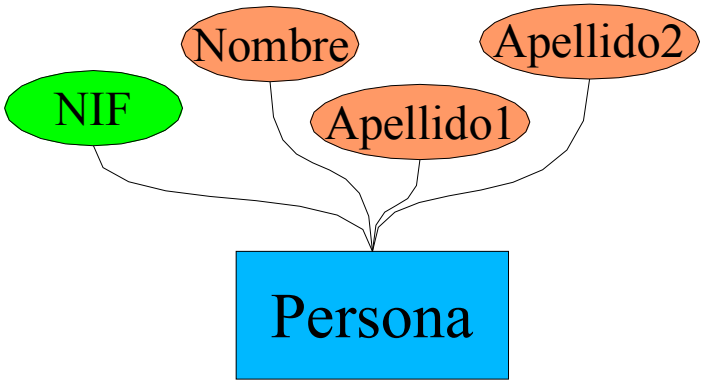

- Al igual que en un sistema informático, existe la figura del administrador. En casi todos los SGBDRs el administrador de una base de datos no tiene por qué ser el administrador del sistema.

- Un administrador crea los usuarios, y les otorga o deniega privilegios (operaciones que pueden realizar).
- Un privilegio es: crear, modificar o borrar una tabla; consultar, insertar, borrar o modificar los datos de una tabla; consultar o crear una vista; crear usuarios o grupos; otorgar privilegios; etc...

Tabla: Restricciones

- Una restricción es una premisa que siempre se debe cumplir. Por ello, los datos almacenados en una tabla siempre deben cumplir todas las restricciones de dicha tabla.

- Existen varios tipos de restricciones
  - De columna (explicado anteriormente)
  - De clave única
  - De clave primaria
  - De clave externa
  - Otras...



Restricción de Clave Única

- Esta restricción se construye sobre una o más columnas, y obliga a que los valores asociados a esas columnas sean únicos. Por ejemplo:

```
(nombre, apellido1, apellido2)
```

  podría definir una clave única, de forma que no pudiese haber dos personas con el mismo nombre y apellidos.

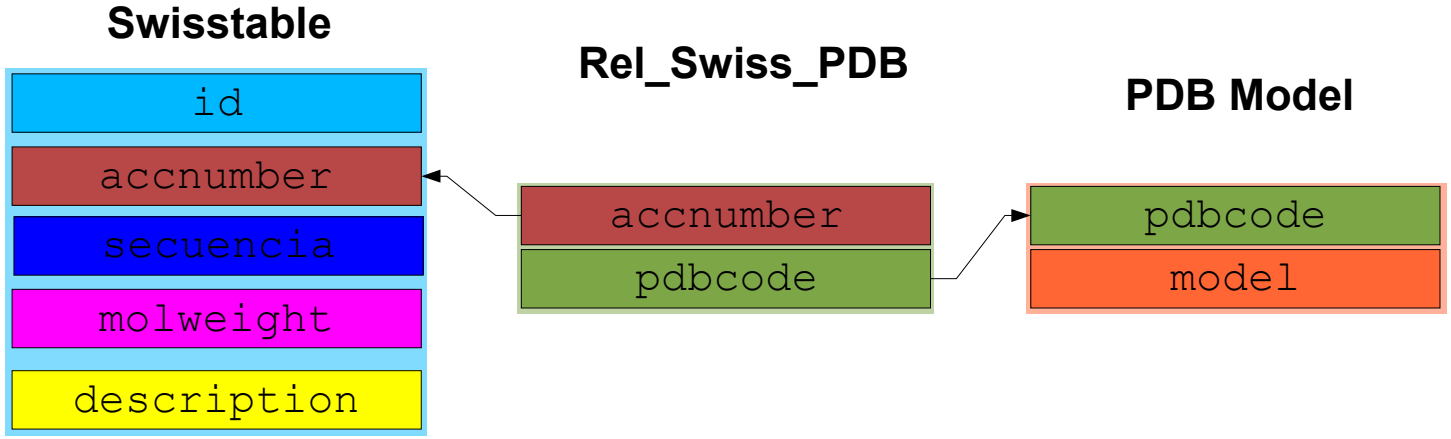
- Una tabla puede tener más de una restricción de clave única. Por ejemplo, una clave única sobre el NIF.

Restricción de Clave Primaria

- Este tipo de restricciones es similar en concepto a las de clave única. Adicionalmente, los valores que toman las columnas de la clave primaria en cada tupla se emplean para identificar dicha tupla de forma lógica.
- Sólo se puede definir una clave primaria por tabla. En caso de existir varios candidatos a clave primaria, lo más conveniente es elegir el más representativo para el contexto de uso.
- Por ejemplo, para un coche, tanto la matrícula como el nº de bastidor se podrían emplear como clave primaria.

Restricción de Clave Externa

- Las restricciones de clave externa sirven para mantener la coherencia entre los datos almacenados en distintas tablas. Se establecen desde los campos de una tabla a los campos de clave primaria de otra.



- Por ejemplo, una base de datos con la tabla Swisstable y la tabla PDB Model, que relacionan sus contenidos a través de la tabla Rel_Swiss_PDB. Para mantener la coherencia, los cambios en el `accnumber` de alguna entrada de Protein o bien estarán prohibidos, o bien provocarán un cambio automático en las entradas de Modeled by con el mismo `accnumber`.

Manipulación de datos

- Una vez definida la estructura de la base de datos, podremos insertar, actualizar, borrar y consultar datos.

- De todas ellas, las consultas serán las operaciones más realizadas, tanto para recuperar información previamente almacenada, como para calcular estadísticas o extraer conclusiones de los datos almacenados.

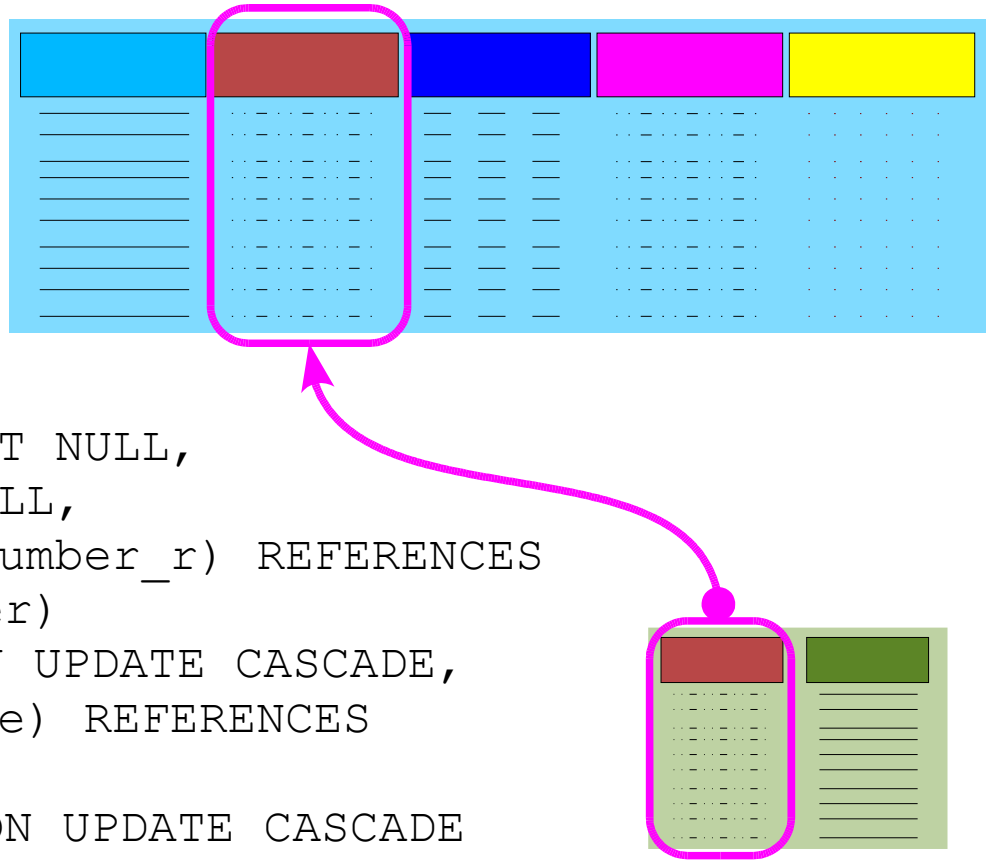
- Un conjunto de operaciones de manipulación de datos se puede realizar en transacción, para garantizar la coherencia de las mismas.

Partes del lenguaje SQL

- DDL (*Data Definition Language*): Es la parte del lenguaje que se ocupa de la gestión de la base de datos: creación y borrado de los usuarios, tablas, vistas, etc...; gestión del control de acceso; manipulación de la estructura de las tablas; optimización del acceso a los datos; tipos de datos...
- DML (*Data Manipulation Language*): Es la parte del lenguaje SQL que se ocupa de las operaciones de inserción, borrado, actualización y consulta de datos.

SQL: Creación de tablas

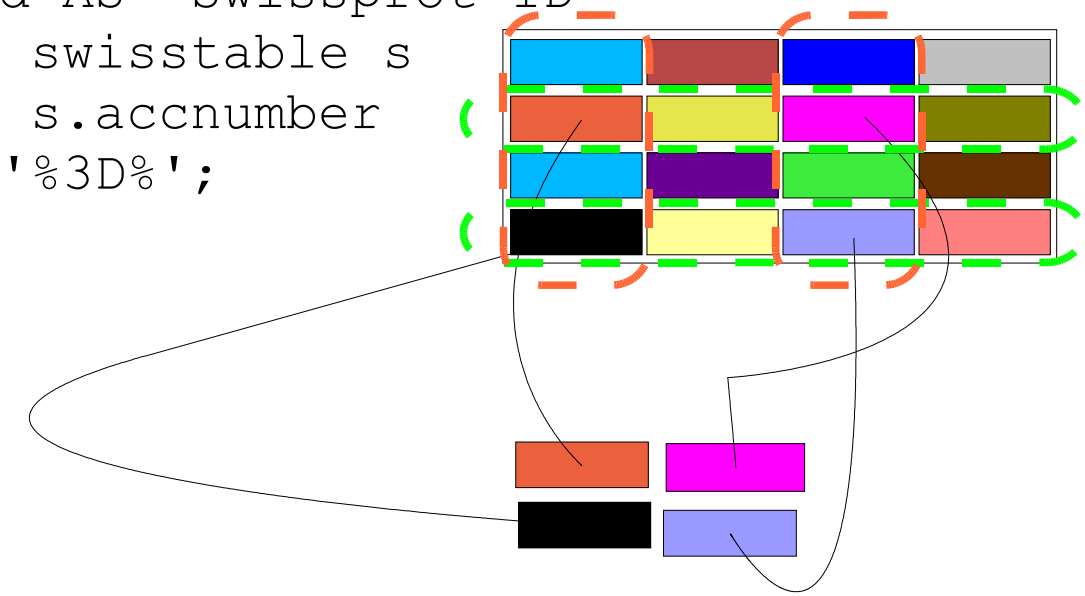
```
CREATE TABLE SWISSTABLE (
      id VARCHAR(10) NOT NULL,
      accnumber VARCHAR(7) NOT NULL,
      secuencia TEXT NOT NULL,
      molweight NUMERIC(8,2),
      description VARCHAR(255),
      PRIMARY KEY (accnumber),
      UNIQUE(id)
);

CREATE TABLE REL_SWISS_PDB (
      accnumber_r VARCHAR(7) NOT NULL,
      pdbcode VARCHAR(8) NOT NULL,
      FOREIGN KEY TOSWISS (accnumber_r) REFERENCES
            SWISSTABLE (accnumber)
            ON DELETE CASCADE ON UPDATE CASCADE,
      FOREIGN KEY TOPDB (pdbcode) REFERENCES
            PDBTABLE (pdb_id)
            ON DELETE RESTRICT ON UPDATE CASCADE
);
```



SQL: Recuperando datos (I)

Consulta normal

```
SELECT p.pdbcode, s.id AS "Swissprot ID"
FROM rel_swiss_pdb p, swisstable s
WHERE p.accnumber_r = s.accnumber
AND description LIKE '%3D%';
```



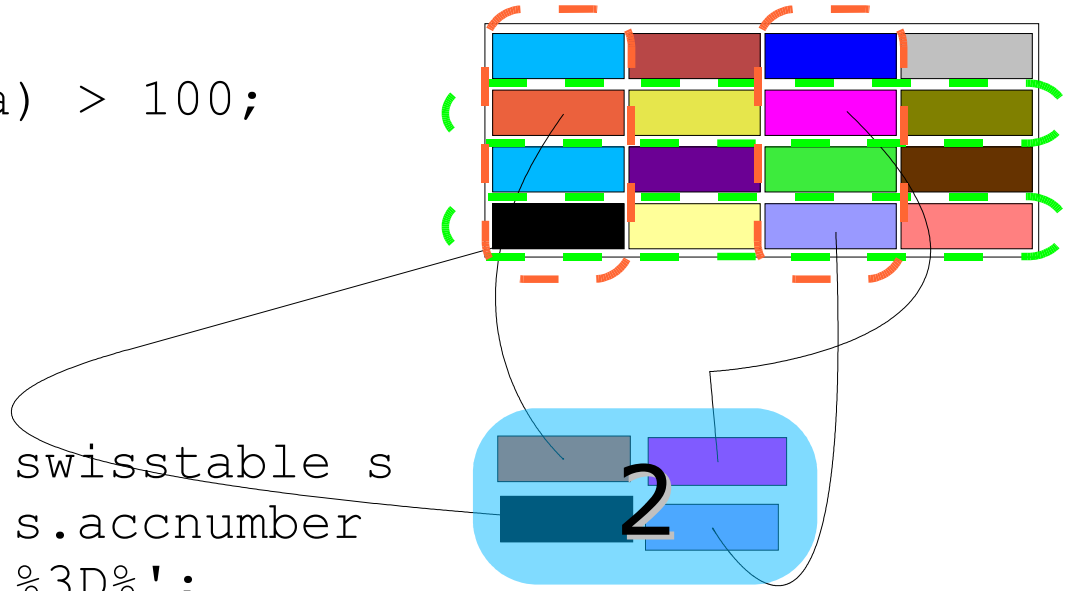
SQL: Recuperando datos (II)

Consulta de agregación simple

```
SELECT COUNT(*)
FROM SWISSTABLE
WHERE LENGTH(secuencia) > 100;
```

Consulta agregación con *join*

```
SELECT COUNT(*)
FROM rel_swiss_pdb p, swisstable s
WHERE p.accnumber_r = s.accnumber
AND description LIKE '%3D%';
```



SQL: Recuperando datos (III)

Consulta de agregación ampliada

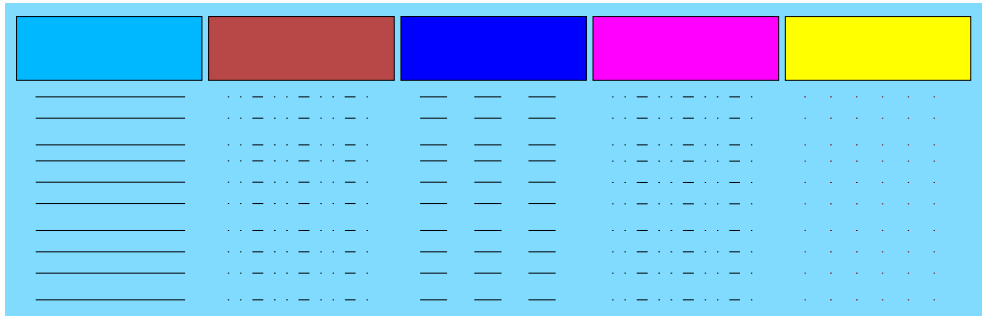
```
SELECT s.id, COUNT(p.pdbcode)
FROM rel_swiss_pdb p, swisstable s
WHERE p.accnumber_r = s.accnumber
AND description LIKE '%3D%'
GROUP BY 1;
```

Id	COUNT
Id_A	2
Id_B	1
Id_C	3

Id	Accnumber	Secuencia	Molweight	Description
Id_A	A	QWEF	38	Blah 3D
Id_B	B	ADSFQ	174	3D Bleh
Id_C	C	SGF	23	Bl 3D ih
Id_D	D	WEWH	229	Bloh
Id_E	E	NMEGY	151	Bluh
Id_F	F	PEUUH	79	Jaja 3D 2

Accnumber_r	Pdbcode
A	P
C	Q
E	R
B	S
C	T
C	U
A	V

Borrado de tablas. Permisos



```
DROP TABLE REL_SWISS_PDB;

GRANT SELECT,INSERT,UPDATE,DELETE
     ON SWISSTABLE TO pepe;

REVOKE SELECT ON SWISSTABLE TO PUBLIC;
```

SQL: Manipulación de datos

Inserción

```
INSERT INTO SWISSTABLE VALUES
     ('P9876',NULL,'LSQSDARESM',18.15,'ID_RAT');
INSERT INTO SWISSTABLE (accnumber,id,molweight, secuencia)
     VALUES ('P9876','ID_RAT',18.15,'LSQSDARESM');
```

Borrado

```
DELETE FROM SWISSTABLE
     WHERE accnumber LIKE 'P98%';
```

Actualización

```
UPDATE SWISSTABLE SET
     molweight = molweight + 1.0
WHERE description IS NULL;
```

Diseño de una base de datos ¿Arte o Ciencia?

- El diseño de la base de datos influye tanto en qué se puede almacenar y consultar, como en los métodos de consulta.
- Existe multitud de herramientas para realizar el diseño a alto nivel de una base de datos. Adicionalmente, existen varias metodologías de diseño, que proporcionan las directrices básicas.
- El diseñador debe conocer tanto el dominio del problema, como el dominio de uso de la futura base de datos.

Recomendaciones de Diseño

- La base de datos tiene que tener una estructura con la complejidad necesaria: ni más ni menos.
- La base de datos tiene que ser funcional: servir para lo que se ha diseñado.
- El diseñador debe tomarse su tiempo para sopesar los pros y los contras del diseño que ha realizado. Es más fácil cambiar la estructura de una base de datos cuando no tiene datos :-)

Interfaces de programación

- Toda base de datos necesita de una serie de programas que realicen las tareas para las que fue diseñada la base de datos.
- Según el lenguaje de programación en el que se encuentre escrito cada programa, habrá que usar uno u otro interfaz de acceso a la base de datos.
- No todos los interfaces están disponibles para todos los SGBDRs y plataformas.
- Interfaces clásicos son: ODBC, JDBC, DBI, ADO, etc...

Referencias

- Manual de PostgreSQL
- Manual de MySQL
- http://www.sqlcourse.com/
- http://www.sqlcourse2.com/
- http://www.w3schools.com/sql/default.asp
- “Mastering SQL”, Martin Gruber, Ed. Sybex
- “SQL for Smarties”, Joe Celko