

# Sistemas de información y base de datos

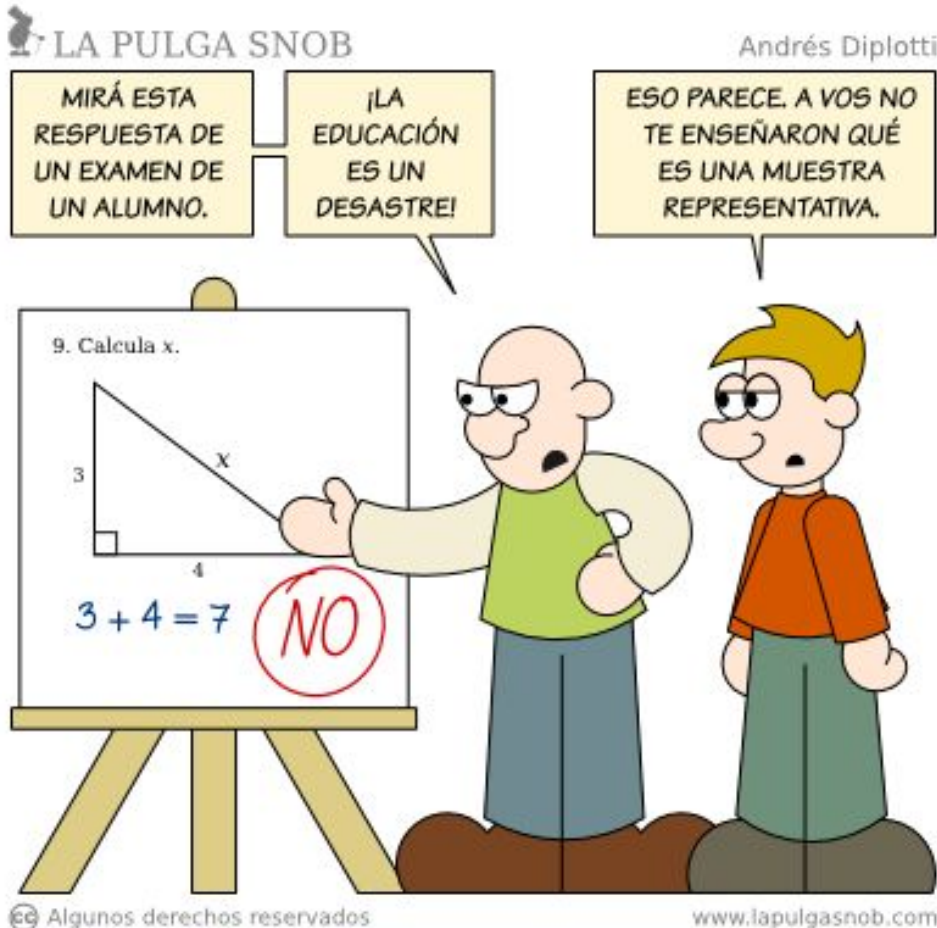
CLASE 6

UNTREF  
2018

Tere [mteresaarriola@gmail.com](mailto:mteresaarriola@gmail.com)

Emilio [emilioplatzer@gmail.com](mailto:emilioplatzer@gmail.com)

Estefi [eycapece@gmail.com](mailto:eycapece@gmail.com)



## Datos abiertos

### Datos públicos

- son los datos sobre los que el público en general tiene derecho a conocer

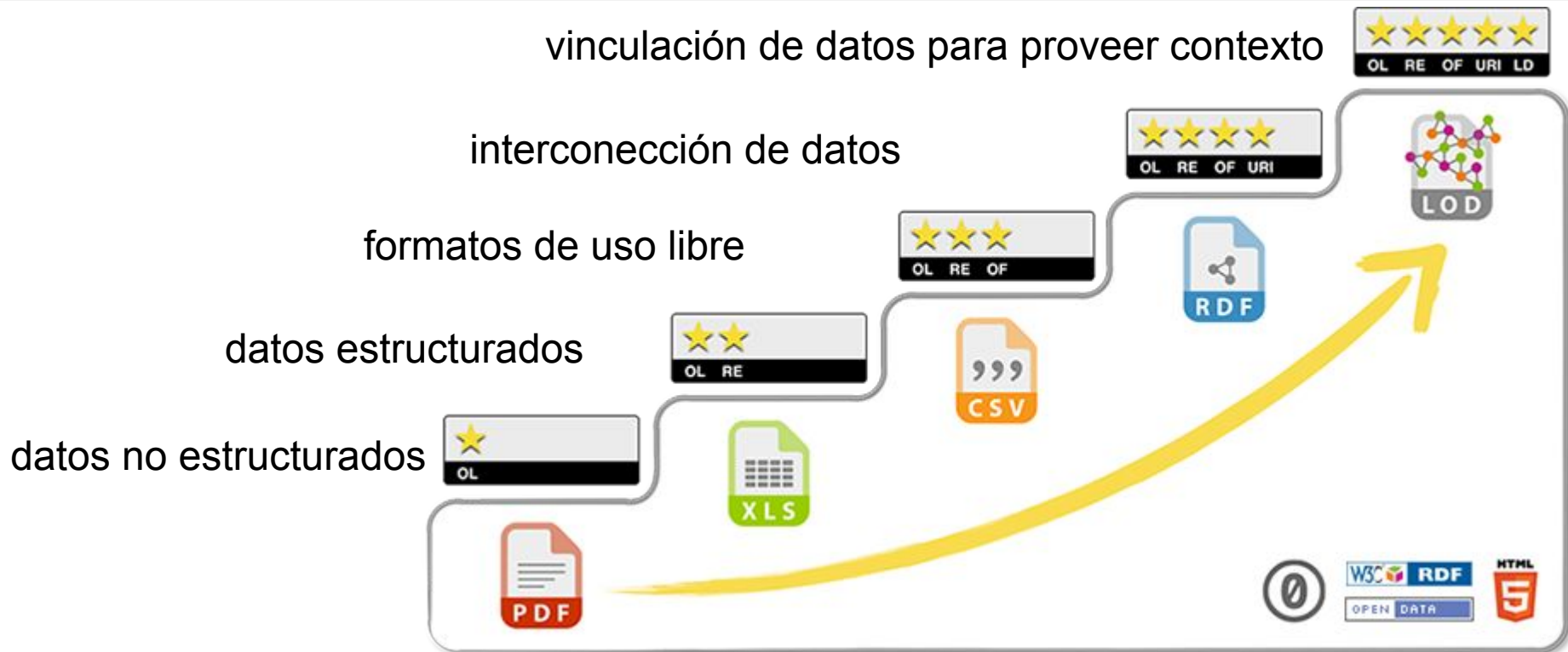
### Datos abiertos

- es la forma en que se dan a conocer esos datos públicos:
  - características tecnológicas de fácil accesibilidad (on line, formatos libres)
  - características legales (se puede copiar, usar, procesar; citando fuente)
  - procesos (permanente, oportuno)

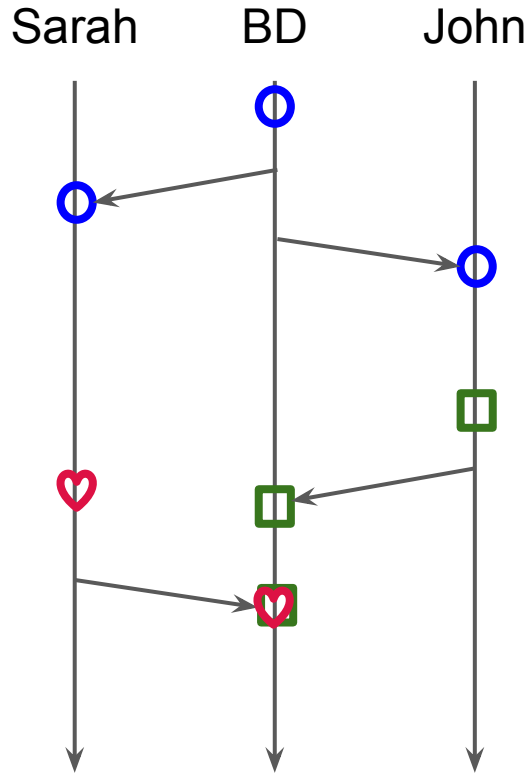
## Datos abiertos - principios básicos

- **Gratuito**, no tiene costo de adquisición
- **De libre uso**, puede utilizarse con cualquier fin (pueden haber condiciones de citar fuente, no engañar, etc)
- **Integral**, debe acompañarse con la información que permita la correcta interpretación de los datos (contexto y metadatos)
- **Oportuno**, están disponibles tan rápido como sea necesario para preservar el valor de los datos
- **No discriminatorio**, no hay restricción de acceso
- **Primario**, directos de la fuente original, con el máximo nivel de desagregación
- **Permanente**, se conservan las versiones previamente publicadas
- **Legibles por máquinas**, estructurados para permitir el procesamiento automatizado de los datos.

## Datos abiertos - nivel de apertura



# Sistemas de Información y Base de Datos



## 3 theories of TIME TRAVEL

CREATED BY HARRISON DENSMORE

### FIXED TIMELINE

IN A FIXED TIMELINE

EVEN WHEN PARTIES TRAVEL BACK IN TIME... THE FUTURE THEY LEFT CANNOT BE CHANGED. ALL EVENTS REMAIN AS FIXED POINTS IN TIME. THE ACTIONS OF THE TRAVELER IN THE PAST HAVE ALREADY BECOME PART OF HISTORY. THIS IS KNOWN AS THE NOVIKOV SELF-CONSISTENCY PRINCIPLE

**FOR EXAMPLE:** SAY YOU TRAVEL BACK IN TIME IN ORDER TO KILL ADOLF HITLER AS A BABY, IN ORDER TO PREVENT WWII.

YOU REPLACE HIM WITH A ORPHANED BABY, SO THAT THE FAMILY WILL NOT NOTICE. YOU TRAVEL BACK TO THE FUTURE, AND THE REPLACED BABY GROWS UP TO BECOME ADOLF HITLER HIMSELF.

**AS SEEN IN:** THE TERMINATOR 2 AND 3, HARRY POTTER 3, 12 MONKEYS

### DYNAMIC TIMELINE

IN A DYNAMIC TIMELINE

ALTERED EVENTS IN THE PAST HAVE DEFINITE IMPACTS ON THE PRESENT.

**FOR EXAMPLE:** IF YOU TRAVEL BACK IN TIME AND KILL YOUR GRANDFATHER... YOU ALSO PREVENT YOUR OWN BIRTH, AND YOUR EVENTUAL TRIP BACK IN TIME. IN TURN, YOUR GRANDFATHER IS NEVER KILLED, AND YOU ARE BORN AGAIN, ONLY TO GO BACK IN TIME AND KILL YOUR GRANDFATHER ANYWAY.

THIS LOOP CONTINUES INFINITELY, AND CREATES A **PARADOX**

**AS SEEN IN:** BACK TO THE FUTURE

### MULTIVERSE

THE CONCEPT OF A MULTIVERSE SUPPORTS ALTERNATE TIMELINES

IN AN INFINITE NUMBER OF PARALLEL UNIVERSES. TRAVELING INTO THE PAST CAUSES A NEW DIVERGENT TIMELINE FROM THE FIRST. BECAUSE OF THIS, THE TRAVELER CAN DO ANYTHING WITH IMPUNITY, AND ONLY THE NEW TIMELINE WILL BE AFFECTED.

**FOR EXAMPLE:** IF YOU TRAVEL BACK IN TIME AND KILL ALL YOUR GRANDPARENTS, NOTHING HAPPENS. THERE IS NO PARADOX, YOU HAVE SIMPLY CREATED A NEW TIMELINE IN WHICH YOU WILL NOT EXIST, BUT THE ORIGINAL TIMELINE IS UNAFFECTED. HOWEVER, YOU CANNOT RETURN TO YOUR ORIGINAL TIMELINE.

**AS SEEN IN:** THE TERMINATOR 2 AND 3, MISFITS, STAR TREK (2009)

## BD c/SQL

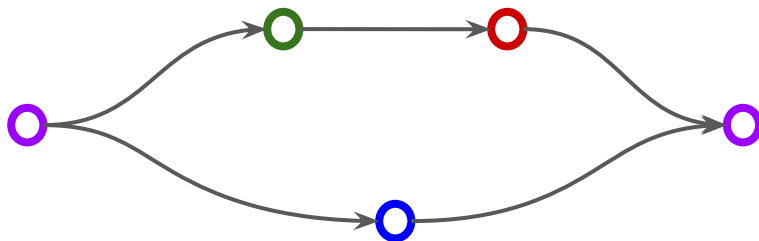
## el sistema transaccional

ACID: Atómico - Consistente - Aislado - Durable

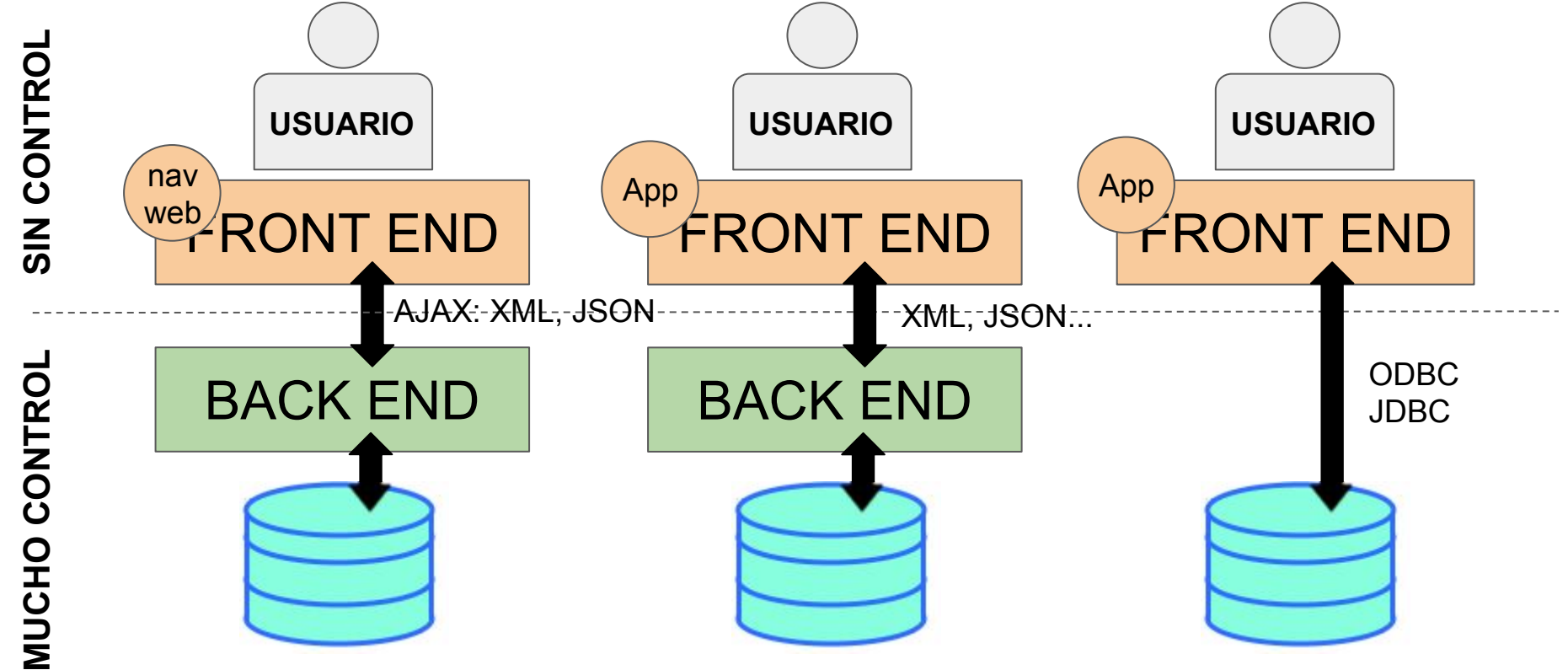
```
BEGIN TRANSACTION;  
INSERT INTO contabilidad  
  (cuenta, fecha, importe)  
VALUES (1, current_date, -100);  
INSERT INTO contabilidad  
  (cuenta, fecha, importe)  
VALUES (2, current_date, 100);  
COMMIT;
```

```
select cuenta, SUM(importe)  
  from contabilidad  
 group by cuenta;
```

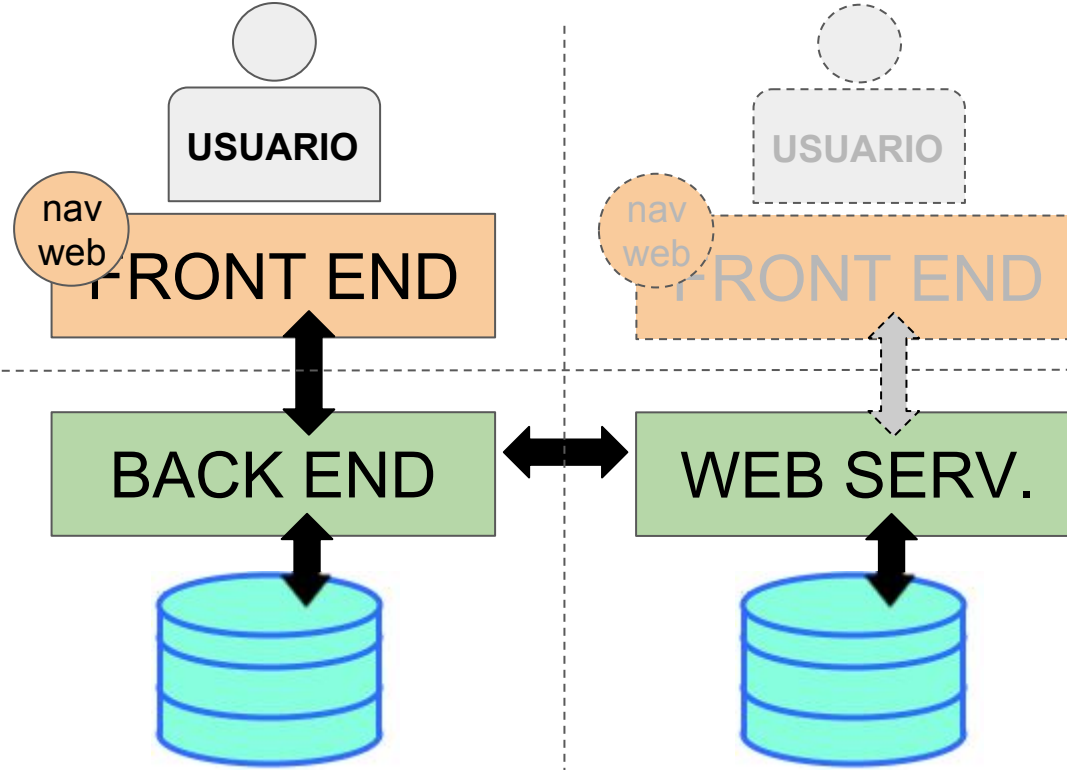
El tiempo se bifurca



## Arquitectura



## Arquitectura - web services





## Confiabilidad - Precisión

- ¿Por qué vamos a usar una **base de datos relacional** basada en **SQL**?
- ¿qué tengo que considerar al utilizar una **f fuente de datos externa**? ¿cómo se transfieren esos datos a una BD?
- ¿Qué es la **pk**? ¿cómo se elige?
- ¿qué es la **integridad referencial**?
- ¿a qué se llama una **operación atómica**?

## El lenguaje SQL

- es un lenguaje de manejo de base de datos que permite:
  - definir la estructura objetos de la base de datos que almacenarán los datos:
    - creando tablas y columnas
    - definir las restricciones que se aplican a esos datos:
      - el tipo de las columnas
      - las columnas (o conjuntos de columnas) que no pueden contener datos repetidos (claves)
      - los valores permitidos para las columnas (puede haber restricción de longitud, cantidad de decimales, rangos válidos)
    - definir las relaciones entre las tablas
      - campos de una tabla que deben ser claves de la otra
  - agregar, modificar y borrar datos de las tablas
    - de a uno o de a varios
    - pudiendo basarse en datos de la misma u otras tablas de la base
  - recuperar los datos almacenados:
    - eligiendo qué registros o columnas mostrar, basándose en datos de una o más tablas, incluyendo un filtro y/o agrupando los datos

## Bases de datos basadas en SQL

- Hay un software que es el motor de la base de datos que puede correr en un servidor o en una máquina común que:
  - Almacena una o más bases de datos relacionales y transaccionales.
  - Recibe órdenes SQL, con las que:
    - actualiza la base de datos
    - devuelve parte de la información almacenada en la base
  - Cumple A.C.I.D.
  - Almacena metadatos de la estructura (nombres de campos, tipos, relaciones, restricciones)
- Ejemplo de motores de base de datos: Postgresql, Oracle, Sql-Server, MySQL, SQLite, MS-Access

## clave primaria / clave secundaria

- Se define como clave a una columna o conjunto de columnas cuyos valores no se repiten en la tabla
- Cada tabla puede tener una (y solo una) clave principal (definida con PRIMARY KEY) y cualquier cantidad de claves secundarias (definida con UNIQUE).
- Se llama clave parcial a una clave que puede tener algunos campos en null (por ejemplo el número de carnet de biblioteca de un alumno es una clave parcial, no se puede repetir, pero algunos alumnos podrían no tener carnet de biblioteca). En SQL se usa UNIQUE para definir claves parciales.
- Una clave principal no puede ser clave parcial (no puede haber nulls).
- Una clave secundaria puede o no ser parcial (dependerá de si las columnas que la componen están definidas con NOT NULL).

```
create table alumnos(  
    id integer,  
    dni integer not null,  
    nombre text not null,  
    carnet_biblio integer  
);
```

```
alter table alumnos  
    add primary key (id),  
    add unique (dni),  
    add unique (carnet_biblio);
```

clave primaria  
(not null implícito)

claves secundarias

clave parcial  
(no tiene not null)

## **una base de datos A.C.I.D.**

- **Atomicidad**
  - Cuando una acción u operación consiste en una serie de pasos la base de datos garantiza que si no pueden ejecutarse la totalidad de ellos la base de datos quedará en el mismo estado en que estaba antes de empezar la acción. Actúa como si la acción deseada fuera indivisible.
- **Consistencia**
  - La base de datos garantiza que todas las operaciones (update, insert, delete) dejan a la base de datos cumpliendo todas las reglas de integridad (claves primarias, relaciones y restricciones se siguen cumpliendo)
- **Aislamiento (Isolation)**
  - La base de datos garantiza que si dos operaciones se pueden ejecutar simultáneamente ninguna de ellas molestará a la otra (por ejemplo permitiendo modificar datos que se suponen no deberían cambiar durante cada operación).
- **Persistencia (Durabilidad)**
  - La base de datos garantiza que ante una falla del sistema (p.ej: corte de luz) los datos que la base de datos informó que fueron grabados se puedan encontrar.

## SQL

### el lenguaje

#### clasificación principal

#### clasificación extendida

**DDL**  
definition

**DML**  
manipulation

**DCL**  
data control

**TCL**  
task control

CREATE TABLE  
ALTER FUNCTION  
DROP VIEW

SELECT  
INSERT INTO  
UPDATE  
DELETE  
TRUNCATE

GRANT  
REVOKE

BEGIN TRANSACTION  
COMMIT  
ROLLBACK

*TRUNCATE*

## SQL

## el lenguaje

## clasificación principal

### DML manipulation

#### instrucciones

**SELECT** obtiene datos de la base de datos

**INSERT INTO** agregar registros nuevos en una tabla

**UPDATE** cambia registros existentes de una tabla

**DELETE** borra registros de una tabla

**TRUNCATE** vacía el contenido de una tabla

#### cláusulas

**JOIN** junta tablas en un registro

**UNION** agrega registros a un resultado

**GROUP BY** agrupa registros

**ORDER BY** ordena registros

**FROM** tablas involucradas

**WHERE** filtro de registros

**SELECT** subconsultas

## SQL

## el lenguaje

## clasificación principal

### DDL definition

CREATE TABLE  
ALTER FUNCTION  
DROP VIEW

crea, modifica o elimina  
tablas, funciones o vistas

CREATE TABLE  
ALTER TABLE

crea una tabla a partir de la definición o de un SELECT  
modifica la estructura de una tabla agregando, quitando  
o modificando:

- campos (nombres y tipos) COLUMN
- la clave primaria PRIMARY KEY
- claves secundarias UNIQUE
- relaciones con otras tablas REFERENCES
- restricciones CONSTRAINT

TRUNCATE



## SQL

**el lenguaje**



**clasificación extendida**



**DCL**  
data control

GRANT  
REVOKE

otorga  
quita

→ permisos de acceso y modificación  
→ de los objetos de la base de datos

## SQL

**el lenguaje**



**clasificación extendida**



**TCL**  
**task control**

comienza una transacción	BEGIN TRANSACTION
finaliza una transacción	COMMIT
aborta una transacción	ROLLBACK

## SELECT

### orden de ejecución lenguaje

4. se eligen las columnas

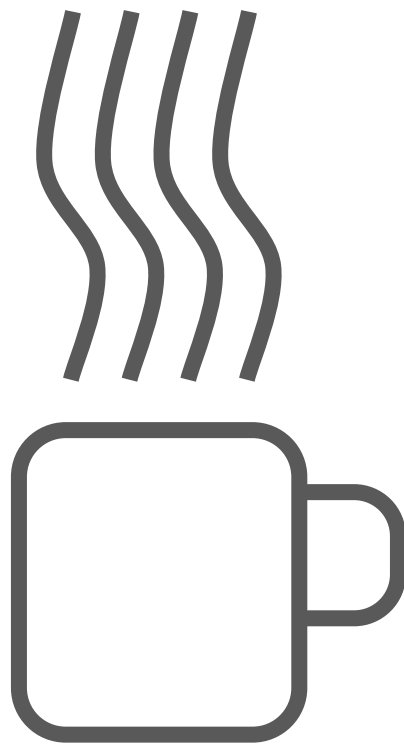
```
SELECT v2_2, count(*) as cant  
FROM hogares h  
    INNER JOIN miembros m  
    ON m.id=h.id AND m.nhogar=h.nhogar  
WHERE nhogar=1  
GROUP BY v2_2  
ORDER BY v2_2;
```

1. se conforma la fuente de datos

2. se filtran los registros

3. se determina si hay agrupamiento y cuál

5. se ordena



## Secreto estadístico

- Protegido por una ley especial
- La información que se *releva* con fines estadísticos debe publicarse en compilaciones que no permitan identificar individuos o empresas.
- La ley es fundamental para la existencia de estadística pública
- Contestar es obligatorio, no hacerlo está multado
- Protege el secreto comercial y patrimonial
- Tiene implicaciones prácticas:
  - Cada empleados y proveedor “firma” un compromiso
  - Debe prestarse atención a la seguridad informática

Ley Nacional 17.622  
Decreto reglamentario 3.110/70  
Ordenanza Municipal 35.386/79  
Ley de la C.A.B.A 451

## Confiabilidad - Seguridad

- ¿qué hay que tener en cuenta al analizar una **licencia**?
- ¿qué es el **software libre**?
- ¿por qué es importante la **seguridad**? ¿qué aspectos?
- ¿qué es el **front-end** y qué es el **back-end**? ¿dónde está la Base de Datos?
- ¿cuáles son específicos de los sistemas estadísticos?
- ¿qué tienen que ver con las **leyes** relacionadas?