

# Sistemas de información y base de datos

**CLASE 2**

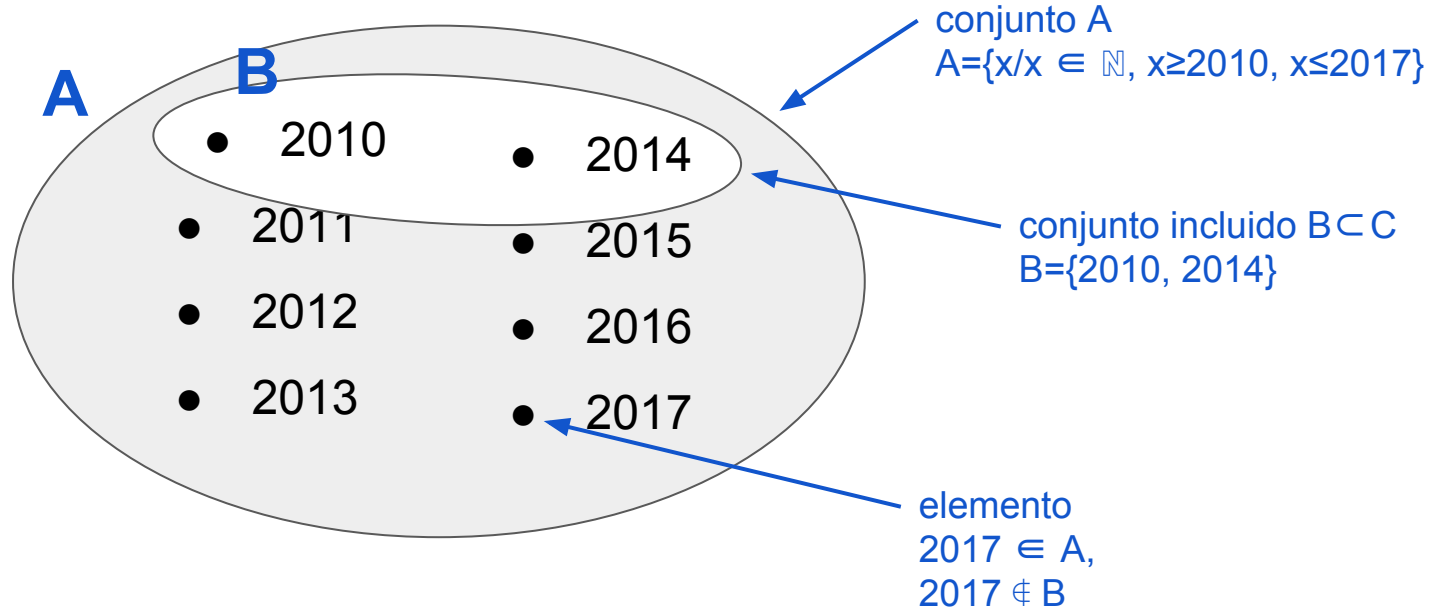
UNTREF  
2016

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

BD

## álgebra de conjuntos

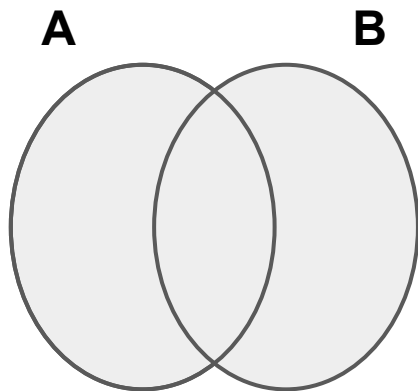
### conjuntos



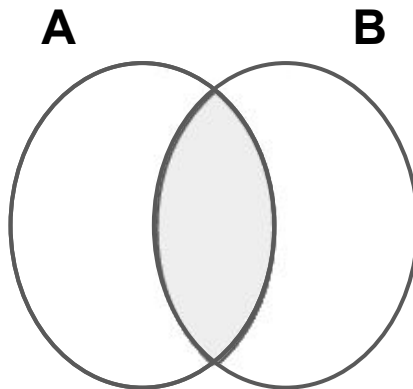
**BD**

## álgebra de conjuntos

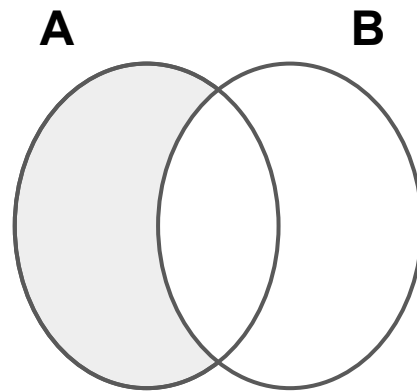
### operaciones con conjuntos



Unión  
 $A \cup B$



Intersección  
 $A \cap B$

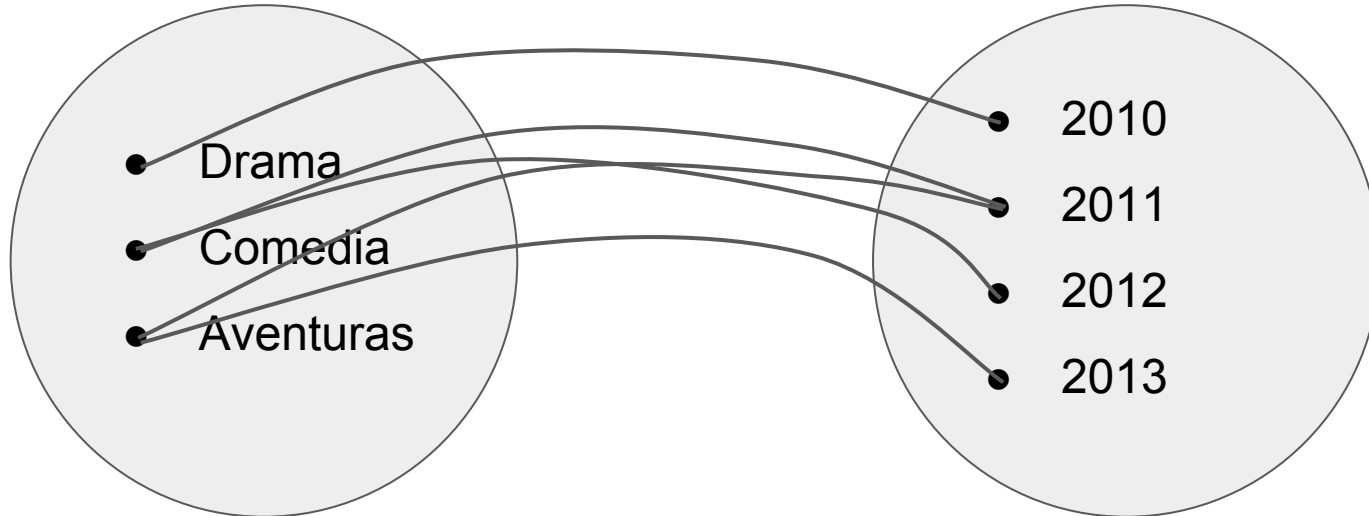


Diferencia  
 $A - B$

**BD**

## álgebra de conjuntos

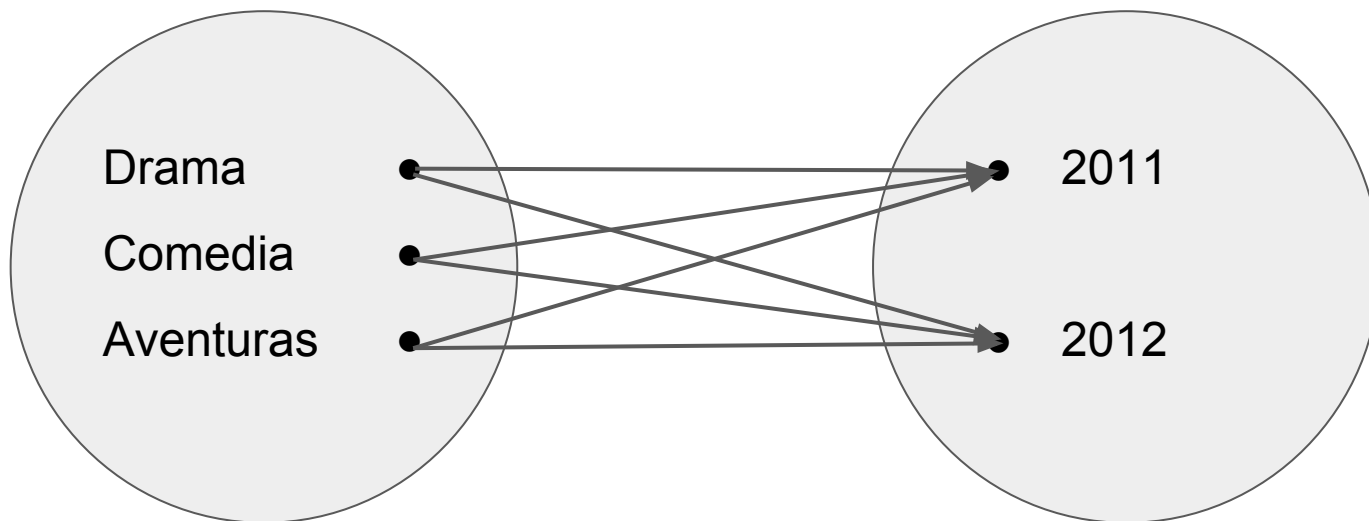
### relaciones



**BD**

## álgebra de conjuntos

### Producto Cartesiano



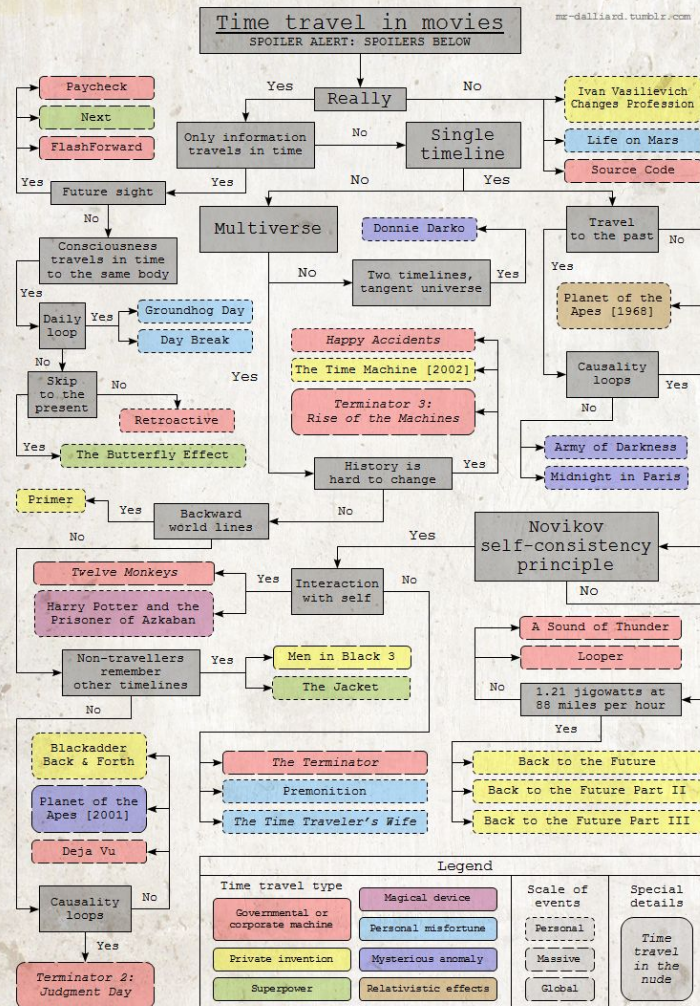
## BD

## el modelo relacional

Relaciones	tablas o relaciones entre conjuntos de datos homogéneos
Atributos	columnas o cada uno de los conjuntos sobre los que se basa la relación
Tuplas	filas o cada uno de los elementos de una relación
Dominios	tipos de las columnas o universos de los conjuntos
Instancia	contenido de la relación en algún momento del tiempo
Claves	restricciones sobre la relación cuando no se concibe que determinados conjuntos de atributos no pueden tener valores repetidos en ninguna instancia

# Sistemas de Información y Base de D

año	nombre	duración	género
1968	El planeta de los simios	112	CF
1984	Terminator	108	CF
1985	Volver al Futuro	116	CF
1989	Volver al Futuro 2	108	CF
1990	Volver al Futuro 3	116	CF
1991	Terminator 2: El juicio final	137	CF
1993	El día de la marmota	119	Comedia F
1995	12 Monos	129	CF
2001	El planeta de los simios	119	CF
2003	Terminator 3: la rebelión de las máquinas	109	CF



## BD

## la tarea de definir la Clave primaria

- Clave es un concepto matemático: un conjunto de atributos  $C_1, C_2, C_n$  es clave si todos los demás atributos dependen funcionalmente de ellos.  $C_1, C_2, C_n \rightarrow A_i$
- La identificación de las claves es conceptual, no solo tiene que cumplirse la dependencia funcional en los datos actuales, tiene que ser inconcebible que en algún momento la dependencia funcional no se cumpla
- Llevado a tablas reales en la base de datos:
  - Hay que definir una clave como clave primaria (PK)
  - No es obligatorio (pero sí conveniente) definir la PK
  - A veces se definen PK autonuméricas (ID) aunque no sea necesario



## **BD c/SQL**

## **el contenido almacenado**

Datos	los datos propiamente dichos
Metadatos	información sobre la estructura de la BD
bitácora	los cambios en la BD
estadísticas	estadística sobre información almacenada
índices	información duplicada para acelerar los resultados de las consultas

## BD c/SQL

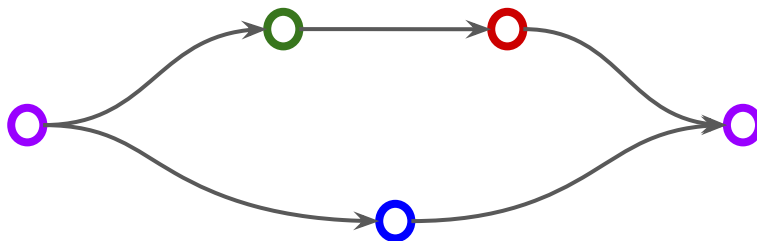
## el sistema transaccional

ACID: Atómico - Consistente - Aislado - Durable

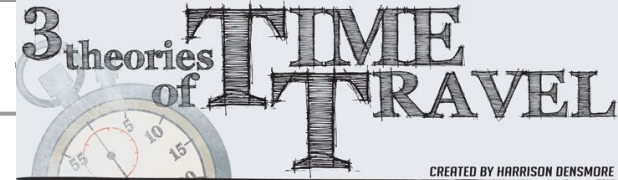
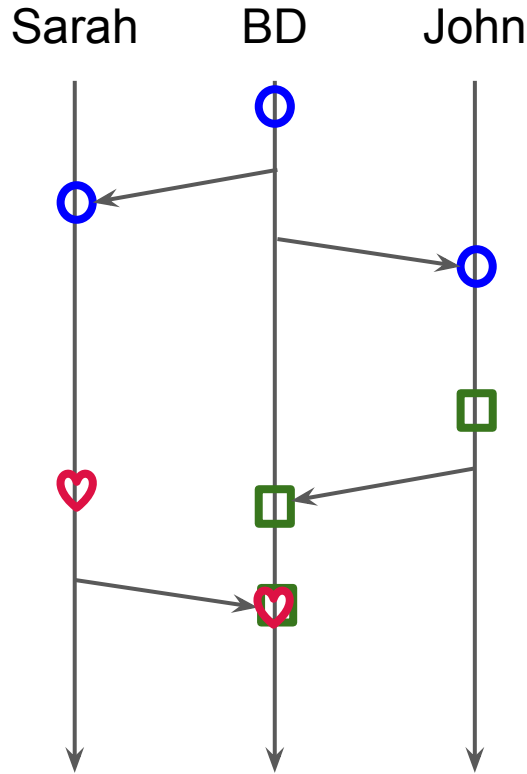
```
BEGIN TRANSACTION;  
INSERT INTO plan  
  SELECT mes, detalle, monto*w  
  FROM plan, modo_detallar  
  WHERE momento IS  
NULL;  
DELETE FROM plan  
  WHERE momento IS NULL;  
COMMIT;  
select * from plan;
```

```
INSERT INTO presupuesto  
  SELECT 'plan', sum(monto)  
  FROM plan;
```

El tiempo se bifurca



# Sistemas de Información y Base de Datos



## 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,

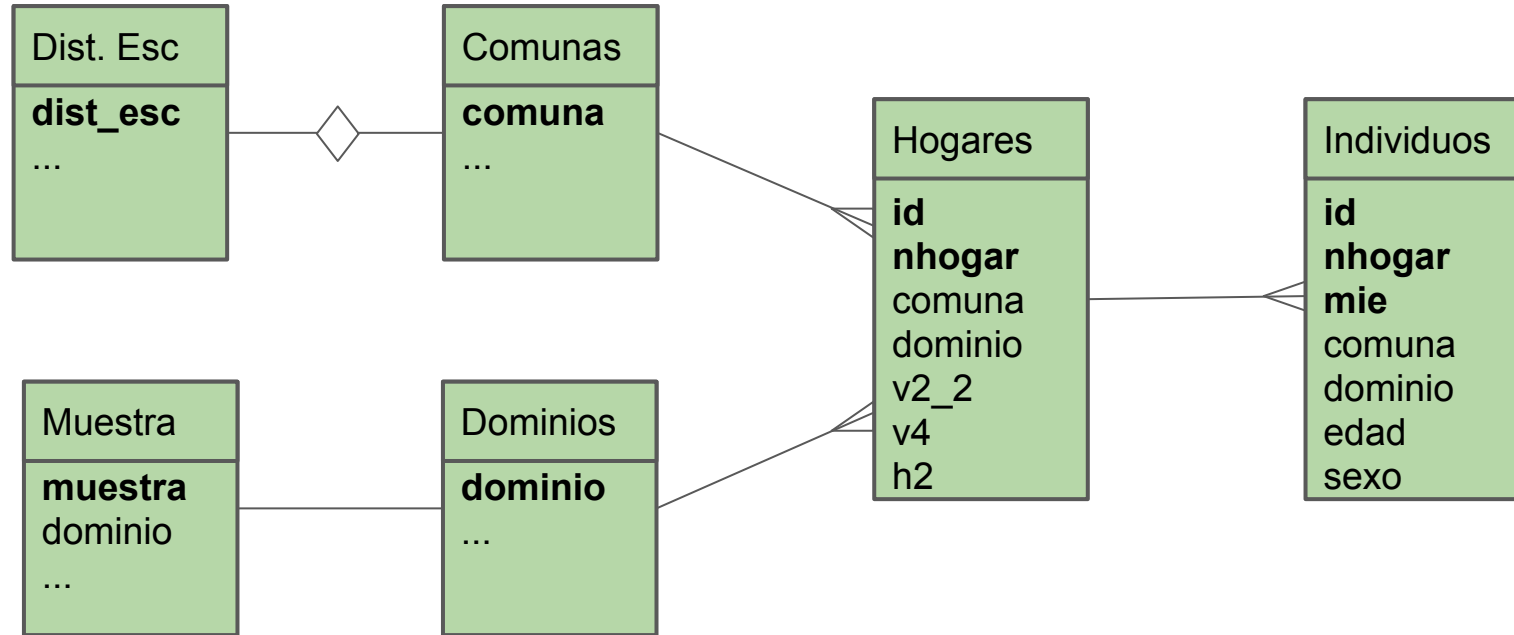


NOTHING HAPPENS. THERE IS NO PARADOX, YOU HAVE SIMPLY CREATED A NEW TIMELINE IN WHICH YOU WILL NOT EXIST, BUT THE ORIGINAL TIMELINE IS UNAFFECTED. HOWEVER, YOU CANNOT RETURN TO YOUR ORIGINAL TIMELINE.

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

**BD**

## el modelo de entidad relación



## 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

### SELECT simple

Lista de campos: \* = todos

Tabla de origen

```
SELECT *  
FROM eah2015_usuarios_hog  
WHERE v2_2 = 5  
ORDER BY id;
```

Orden

Filtro  
¿qué registros quiero?  
(si no está quiero todos)

## SQL

### SELECT agrupado

Lista de campos o expresiones

alias

```
SELECT v2_2, count(*) as cant
FROM eah2015_usuarios_hog
WHERE nhogar=1
GROUP BY v2_2
ORDER BY v2_2;
```

Agrupamiento

Función de agregación

COUNT, SUM, AVG,  
MAX, MIN

## SQL

## SELECT c/subconsultas

```
WITH viviendas AS (  
  SELECT CASE v2_2 WHEN 1 THEN 'casa'  
            WHEN 2 THEN 'dept'  
            ELSE 'otros'  
  END AS tipo_viv,  
  fexp  
  FROM eah2015_usuarios_hog  
  WHERE nhogar=1)  
SELECT tipo_viv,  
       SUM(fexp)*100.0/(SELECT SUM(fexp) FROM viviendas)  
FROM viviendas  
GROUP BY tipo_viv  
ORDER BY tipo_viv;
```

subconsulta

reclasificación

uso de la  
subconsulta



## SQL

## SELECT c/juntas

```
WITH personas AS (  
    SELECT CASE v2_2 WHEN 1 THEN 'casa'  
              WHEN 2 THEN 'dept'  
              ELSE 'otros'  
            END AS tipo_viv,  
            i.fexp  
    FROM eah2015_usuarios_hog h  
    INNER JOIN eah2015_usuarios_ind i ON h.id=i.id  
)  
SELECT tipo_viv, SUM(fexp)  
      SUM(fexp)*100.0/(SELECT SUM(fexp) FROM personas),  
FROM personas  
GROUP BY tipo_viv;
```

La junta

La condición