

# Algebra Relacional y Cálculo Relacional de Tuplas

Dr. Gerardo Rossel

Bases de Datos

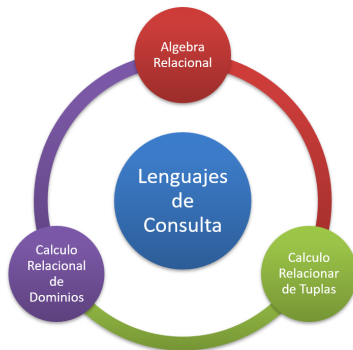
2025





# Lenguajes de Consulta

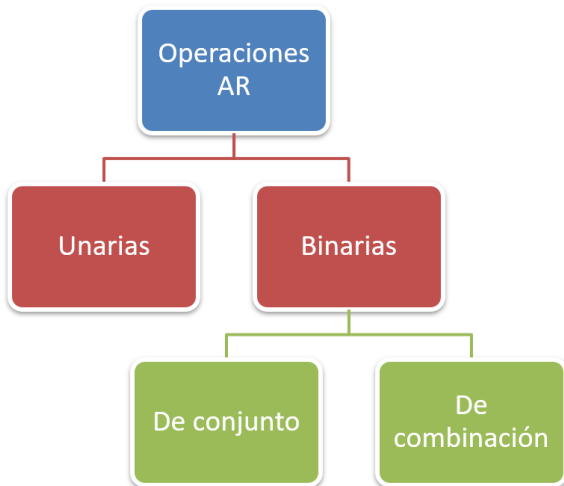
- Todo modelo de base de datos debe incluir un conjunto de operaciones para manipular la base.
- Un conjunto de operaciones fundamentales son las de recuperación de datos.



# Algebra Relacional

- Es un lenguaje formal de consultas para recuperar datos almacenados en un modelo relacional.
- Es un lenguaje **procedural**, se detallan las operaciones que deben realizarse para obtener los resultados deseados.
- Las consultas de AR:
  - Reciben como entrada una o dos instancias de relación
  - **Devuelven una instancia de relación** (las operaciones son cerradas)
- Las operaciones de AR pueden componerse . El input de una operación puede ser el resultado de una operación.

# Operaciones del AR



# Operaciones Unarias

- Proyección:  $\pi_{\langle \text{lista de atributos} \rangle}(R)$ , permite recuperar algunos atributos de una relación. Mantiene todas las tuplas origen.
- Selección:  $\sigma_{\langle \text{predicado} \rangle}(R)$ , permite recuperar algunas filas de una relación. Mantiene todos los atributos origen.
- Renombre
  - Renombra atributos:  $\rho(a1 \rightarrow a2, b1 \rightarrow b2, R)$
  - Renombra una relación:  $\rho(S, R \bowtie R)$
  - Ejemplos, relación  $A(id)$ :
    - $\rho(1 \rightarrow id1, 2 \rightarrow id2, A \times A)$
    - $\rho(B(1 \rightarrow id1, 2 \rightarrow id2), A \times A)$

# Proyección $\pi$

$$\pi \langle \text{lista de atributos} \rangle (R)$$

La operación de proyección actúa sobre una única relación R y define una nueva relación que contiene un subconjunto vertical de R, extrayendo los valores de atributos especificados y **eliminando duplicados**.

$$\pi_{\text{sex}}(\text{Staff})$$

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

# Proyección $\pi$

$$\pi_{\langle \text{lista de atributos} \rangle}(R)$$

La operación de proyección actúa sobre una única relación  $R$  y define una nueva relación que contiene un subconjunto vertical de  $R$ , extrayendo los valores de atributos especificados y **eliminando duplicados**.

$$\pi_{\text{sex}}(\text{Staff})$$

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

sex
M
F



# Selección $\sigma$

$$\sigma_{\langle predicado \rangle}(R)$$

La operación de selección actúa sobre una única relación R y define una nueva relación que contiene únicamente aquellas tuplas de R que satisfacen la condición (predicado) especificada.

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

$$\sigma_{salary > 1000}(Staff)$$

# Selección $\sigma$

$$\sigma_{\langle predicado \rangle}(R)$$

La operación de selección actúa sobre una única relación  $R$  y define una nueva relación que contiene únicamente aquellas tuplas de  $R$  que satisfacen la condición (predicado) especificada.

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

$$\sigma_{salary > 1000}(Staff)$$

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003

Renombre  $\rho$ 

Empleado

Id	Nombre	Departamento	Sueldo
1	Ana López	Ventas	75000
2	Juan Pérez	Marketing	68000
3	Marta Díaz	Finanzas	82000
4	Luis Gómez	IT	91000
5	Carla Ruiz	RRHH	70000

$$\rho(Id \rightarrow IdEmpleado, Empleado)$$

Renombre  $\rho$ 

Empleado

Id	Nombre	Departamento	Sueldo
1	Ana López	Ventas	75000
2	Juan Pérez	Marketing	68000
3	Marta Díaz	Finanzas	82000
4	Luis Gómez	IT	91000
5	Carla Ruiz	RRHH	70000

$$\rho(Id \rightarrow IdEmpleado, Empleado)$$

IdEmpleado	Nombre	Departamento	Sueldo
1	Ana López	Ventas	75000
2	Juan Pérez	Marketing	68000
3	Marta Díaz	Finanzas	82000
4	Luis Gómez	IT	91000
5	Carla Ruiz	RRHH	70000

# Operadores Binarios: De conjuntos

- Unión:  $R \cup S$
- Intersección:  $R \cap S$
- Resta:  $R - S$

## Unión Compatibles

Sólo son posibles si las relaciones son **Unión Compatibles**. Relaciones con misma cantidad de atributos y mismo dominio atributo a atributo (importa el orden).

# Unión

## Unión

Es la unión matemática de conjuntos. El resultado incluye a las tuplas que están en la primera relación, en la segunda o en ambas, eliminando tuplas duplicadas.

Empleados1

Id	Nombre	Departamento	Sueldo
1	Ana López	Ventas	75000
2	Juan Pérez	Marketing	68000
3	Marta Díaz	Finanzas	82000

Empleados2

Id	Nombre	Departamento	Sueldo
3	Marta Díaz	Finanzas	82000
4	Luis Gómez	IT	91000
5	Carla Ruiz	RRHH	70000

 $\text{Empleados1} \cup \text{Empleados2}$ 

Id	Nombre	Departamento	Sueldo
1	Ana López	Ventas	75000
2	Juan Pérez	Marketing	68000
3	Marta Díaz	Finanzas	82000
4	Luis Gómez	IT	91000
5	Carla Ruiz	RRHH	70000

# Intersección

## Intersección

Es la intersección matemática de conjuntos. El resultado incluye a las tuplas que están en ambas relaciones.

*Empleados1*

Id	Nombre	Departamento	Sueldo
1	Ana López	Ventas	75000
2	Juan Pérez	Marketing	68000
3	Marta Díaz	Finanzas	82000

*Empleados2*

Id	Nombre	Departamento	Sueldo
3	Marta Díaz	Finanzas	82000
4	Luis Gómez	IT	91000
5	Carla Ruiz	RRHH	70000

*Empleados1*  $\cap$  *Empleados2*

Id	Nombre	Departamento	Sueldo
3	Marta Díaz	Finanzas	82000

# Resta o Diferencia

## Intersección

Es la diferencia matemática de conjuntos. El resultado incluye a las tuplas que están en la primera relación y no están en la segunda.

*Empleados1*

Id	Nombre	Departamento	Sueldo
1	Ana López	Ventas	75000
2	Juan Pérez	Marketing	68000
3	Marta Díaz	Finanzas	82000

*Empleados2*

Id	Nombre	Departamento	Sueldo
3	Marta Díaz	Finanzas	82000
4	Luis Gómez	IT	91000
5	Carla Ruiz	RRHH	70000

*Empleados1 – Empleados2*

Id	Nombre	Departamento	Sueldo
1	Ana López	Ventas	75000
2	Juan Pérez	Marketing	68000



# Operaciones Binarias de Combinación

- Producto Cartesiano:  $R \times S$ .
- Theta join:  $R \bowtie_{\langle predicado \rangle} S$ . El predicado puede usar cualquier operador lógico
- Equijoin:  $R \bowtie_{\langle predicado \rangle} S$ . El predicado sólo esta compuesto de comparaciones de igualdad. Remueve el atributo de la derecha de la relación resultado
- Junta Natural:  $R \bowtie S$ . Junta por atributos con el mismo nombre. Se mantiene solo un atributo de los duplicados.
- Right outer join  $R \bowtie\!\!\!\bowtie S$ . Conserva todas las tuplas de  $S$ . Si no se encuentra ninguna tupla de  $R$  que cumpla con la condición de *JOIN*, entonces los atributos de  $R$  en el resultado se completan en NULL.
- Left outer join  $R \bowtie\!\!\!\bowtie S$ . Conserva todas las tuplas de  $R$ . Si no se encuentra ninguna tupla de  $S$  que cumpla con condición de *JOIN*, entonces los atributos de  $S$  en el resultado se completan en NULL.
- Full outer join  $R \bowtie\!\!\!\bowtie S$ . Conserva todas las tuplas de ambas relaciones. Si no se encuentra ninguna tupla de la otra relación que cumpla con condición de *JOIN*, entonces los atributos de la otra relación en el resultado se completan en NULL
- División  $R(Z) \div S(X)$ . Si  $X \subseteq Z$ . Sea  $Y$  el conjunto de atributos de  $R$  tal que  $Y = Z - X$  (también  $Z = X \cup Y$ ). La división es una relación  $T(Y)$  es decir con atributos de  $R$  que no están en el esquema de  $S$  y una tupla  $t$  esta en la  $T(Y)$  si: **a)**  $t \in \pi_Y(R)$  y **b)** para toda tupla  $t_S \in S$  hay una tupla  $t_R \in R$  tal que  $t_R[S] = t_S[S]$  y  $t_R[R - S] = t$

# Producto cartesiano

## Producto Cartesiano

Devuelve una relación donde cada elemento es la combinación de cada tupla de la primera relación con cada tupla de la segunda. Notar que no exige ninguna relación entre las tuplas para combinarlas

*Estudiantes*

IdEst	Nombre
1	Ana López
2	Juan Pérez
3	Marta Díaz

*Cursos*

IdCurso	NombreCurso
101	Matemáticas
102	Física
103	Química

*Estudiantes × Cursos*

IdEst	Nombre	IdCurso	NombreCurso
1	Ana López	101	Matemáticas
1	Ana López	102	Física
1	Ana López	103	Química
2	Juan Pérez	101	Matemáticas
2	Juan Pérez	102	Física
2	Juan Pérez	103	Química
3	Marta Díaz	101	Matemáticas
3	Marta Díaz	102	Física
3	Marta Díaz	103	Química

# Tetha Join

Empleados

IdEmp	Nombre	Edad
1	Ana López	25
2	Juan Pérez	30
3	Marta Díaz	22

Proyectos

IdProyecto	NombreProyecto	EdadMinima
101	Alpha	24
102	Beta	28
103	Gamma	21

*Empleados* ⋈<sub>(Edad > EdadMinima)</sub> *Proyectos*

IdEmp	Nombre	Edad	IdProyecto	NombreProyecto	EdadMinima
1	Ana López	25	101	Alpha	24
1	Ana López	25	103	Gamma	21
2	Juan Pérez	30	101	Alpha	24
2	Juan Pérez	30	102	Beta	28
2	Juan Pérez	30	103	Gamma	21
3	Marta Díaz	22	103	Gamma	21

# Junta Natural

## Natural Join - Junta Natural

Combina las tuplas de dos relaciones que cumplen que todos los atributos de nombre idéntico tienen igual valor, removiendo los atributos innecesarios. O sea deja solo un atributo de los usados para junta.

Estudiantes

IdEst	Nombre	IdCurso
1	Ana López	101
2	Juan Pérez	102
3	Marta Díaz	101

Cursos

IdCurso	NombreCurso
101	Matemáticas
102	Física
103	Química

Estudiantes ⋈ Cursos (Natural Join)

IdEst	Nombre	IdCurso	NombreCurso
1	Ana López	101	Matemáticas
2	Juan Pérez	102	Física
3	Marta Díaz	101	Matemáticas

# Left Outer Join

## Left Outer Join

Es como las juntas anteriores pero devuelve todas la tuplas de la primera relación, aún cuando no combinen con ninguna tupla de la segunda.

*Empleados*

IdEmp	Nombre	IdDept
1	Ana López	101
2	Juan Pérez	102
3	Marta Díaz	104
4	Luis Gómez	NULL

*Departamentos*

IdDept	NombreDept
101	Finanzas
102	IT
103	RRHH

*Empleados ⋈ Departamentos*

IdEmp	Nombre	IdDept	NombreDept
1	Ana López	101	Finanzas
2	Juan Pérez	102	IT
3	Marta Díaz	104	NULL
4	Luis Gómez	NULL	NULL

## Right Outer Join y Full Outer Join

Right Outer Join: Es igual solo que devuelve todas las tuplas de la relación de la derecha.

# Ejemplo de División en Álgebra Relacional

*Inscripciones*

Alumno	Curso
Ana	Matem
Ana	Física
Ana	Química
Juan	Matem
Juan	Física
Juan	Química
Marta	Matem
Marta	Química
Luis	Física
Luis	Química

*CursosRequeridos*

Curso
Matem
Física
Química

**Alumnos que tomaron todos los cursos requeridos**

*Inscripciones / CursosRequeridos*

Alumno
Ana
Juan

# Secuenciación

La forma de secuenciar operaciones es dando un nombre a las operaciones intermedias, eso se logra mediante el renombre de relaciones.  
(alternativamente se puede usar  $\leftarrow$ ).

$\rho(\text{Mayores20}, \sigma_{\text{edad} > 20}(\text{Estudiantes}))$

$\rho(\text{Resultado}, \pi_{\text{Nombre}}(\text{Mayores20}))$

Es lo mismo que:

$\rho(\text{Resultado}, \pi_{\text{nombre}}(\sigma_{\text{edad} > 20}(\text{Estudiantes})))$

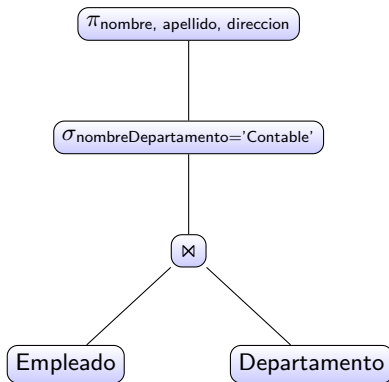
# Arbol Canónico

$\pi_{\text{nombre,apellido,direccion}}(\sigma_{\text{nombreDepartamento}='Contable'}(\text{Empleado} \bowtie \text{Departamento}))$



# Arbol Canónico

$\pi_{\text{nombre, apellido, direccion}}(\sigma_{\text{nombreDepartamento}='Contable'}(\text{Empleado} \bowtie \text{Departamento}))$



# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
- Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
- Género(idGénero, nombreGénero)
- Canal(idCanal, nombreCanal)
- Participa\_En(idActor, idSerie)
- Transmite(idCanal, idSerie)

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
  - Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
  - Género(idGénero, nombreGénero)
  - Canal(idCanal, nombreCanal)
  - Participa\_En(idActor, idSerie)
  - Transmite(idCanal, idSerie)
- 
- ¿Cómo obtengo todas las series de un actor A?

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
  - Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
  - Género(idGénero, nombreGénero)
  - Canal(idCanal, nombreCanal)
  - Participa\_En(idActor, idSerie)
  - Transmite(idCanal, idSerie)
- ¿Cómo obtengo todas las series de un actor A?
- PARTICIPA\_EN*

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
- Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
- Género(idGénero, nombreGénero)
- Canal(idCanal, nombreCanal)
- Participa\_En(idActor, idSerie)
- Transmite(idCanal, idSerie)

- ¿Cómo obtengo todas las series de un actor A?

*PARTICIPA\_EN*

$\sigma_{idActor='A'}(PARTICIPA\_EN)$

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
- Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
- Género(idGénero, nombreGénero)
- Canal(idCanal, nombreCanal)
- Participa\_En(idActor, idSerie)
- Transmite(idCanal, idSerie)

- ¿Cómo obtengo todas las series de un actor A?

*PARTICIPA\_EN*

$\sigma_{idActor='A'}(PARTICIPA\_EN)$

$\pi_{idSerie}(\sigma_{idActor='A'}(PARTICIPA\_EN))$

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
  - Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
  - Género(idGénero, nombreGénero)
  - Canal(idCanal, nombreCanal)
  - Participa\_En(idActor, idSerie)
  - Transmite(idCanal, idSerie)
- ¿Cómo lo hago si A es el nombre y no el identificador?

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
  - Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
  - Género(idGénero, nombreGénero)
  - Canal(idCanal, nombreCanal)
  - Participa\_En(idActor, idSerie)
  - Transmite(idCanal, idSerie)
- ¿Cómo lo hago si A es el nombre y no el identificador?
- PARTICIPA\_EN* ⋈ *ACTOR*



# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
- Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
- Género(idGénero, nombreGénero)
- Canal(idCanal, nombreCanal)
- Participa\_En(idActor, idSerie)
- Transmite(idCanal, idSerie)

- ¿Cómo lo hago si A es el nombre y no el identificador?

*PARTICIPA\_EN* ⋈ *ACTOR*

$\sigma_{\text{nombreActor} = 'A'}(PARTICIPA\_EN \bowtie ACTOR)$

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
- Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
- Género(idGénero, nombreGénero)
- Canal(idCanal, nombreCanal)
- Participa\_En(idActor, idSerie)
- Transmite(idCanal, idSerie)

- ¿Cómo lo hago si A es el nombre y no el identificador?

$PARTICIPA\_EN \bowtie ACTOR$

$\sigma_{nombreActor='A'}(PARTICIPA\_EN \bowtie ACTOR)$

$\pi_{idSerie}(\sigma_{nombreActor='A'}(PARTICIPA\_EN \bowtie ACTOR))$

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
  - Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
  - Género(idGénero, nombreGénero)
  - Canal(idCanal, nombreCanal)
  - Participa\_En(idActor, idSerie)
  - Transmite(idCanal, idSerie)
- 
- ¿Y si quiero los nombres de las series en lugar de los identificadores?

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
  - Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
  - Género(idGénero, nombreGénero)
  - Canal(idCanal, nombreCanal)
  - Participa\_En(idActor, idSerie)
  - Transmite(idCanal, idSerie)
- ¿Y si quiero los nombres de las series en lugar de los identificadores?
- $$\sigma_{nombreActor='A'}(PARTICIPA\_EN \bowtie ACTOR \bowtie SERIE)$$

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
  - Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
  - Género(idGénero, nombreGénero)
  - Canal(idCanal, nombreCanal)
  - Participa\_En(idActor, idSerie)
  - Transmite(idCanal, idSerie)
- ¿Y si quiero los nombres de las series en lugar de los identificadores?

$$\sigma_{\text{nombreActor}='A'}(PARTICIPA\_EN \bowtie ACTOR \bowtie SERIE)$$

$$\rho(SERIES\_DE\_A, \sigma_{\text{nombreActor}='A'}(PARTICIPA\_EN \bowtie ACTOR \bowtie SERIE))$$

# Ejemplo de ejercicio en AR

- Actor(idActor, nombreActor, edad)
- Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)
- Género(idGénero, nombreGénero)
- Canal(idCanal, nombreCanal)
- Participa\_En(idActor, idSerie)
- Transmite(idCanal, idSerie)
- ¿Y si quiero los nombres de las series en lugar de los identificadores?

$\sigma_{\text{nombreActor}='A'}(PARTICIPA\_EN \bowtie ACTOR \bowtie SERIE)$

$\rho(SERIES\_DE\_A, \sigma_{\text{nombreActor}='A'}(PARTICIPA\_EN \bowtie ACTOR \bowtie SERIE))$

$\rho(RESULTADO, \pi_{\text{nombreSerie}}(SERIES\_DE\_A))$

# Ejercicios

- ❶ Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends”.
- ❷ Listar los nombres de los canales que transmiten todas las series de comedia.
- ❸ Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además participaron alguna vez en series que comenzaron luego del año 2000.
- ❹ Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además nunca participaron en series que comenzaron luego del año 2000.
- ❺ Listar los ids de las series que comenzaron más recientemente.
- ❻ Listar los ids de los actores que participaron en al menos 2 series.

# Soluciones - Álgebra Relacional

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie "Friends".

$$\rho(PAS, PARTICIPA\_EN \bowtie ACTOR \bowtie SERIE)$$

$$\rho(PAS', \sigma_{nombreSerie='Friends' \wedge edad > 30}(PAS))$$

$$\rho(Resultado, \pi_{nombreActor}(PAS'))$$



# Soluciones - Álgebra Relacional

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los canales que transmiten todas las series de comedia.

$$\rho(ID\_SERIE\_COMEDIA, \pi_{idSerie}(\sigma_{nombreGénero=Comedia}(SERIE \bowtie GENERO)))$$

$$\rho(ID\_CANAL\_TODAS\_COMEDIAS, TRANSMITE \div ID\_SERIE\_COMEDIA)$$

$$\rho(Resultado, \pi_{nombreCanal}(ID\_CANAL\_TODAS\_COMEDIAS \bowtie CANAL))$$

# Soluciones - Álgebra Relacional

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie "Friends" y que además participaron alguna vez en series que comenzaron luego del año 2000.

$$\rho(ID\_ACTOR\_FR\_MAY\_30, \pi_{idActor}(\sigma_{nombreSerie="Friends" \wedge edad > 30} (PARTICIPA\_EN \bowtie ACTOR \bowtie SERIE)))$$

$$\rho(ID\_ACTOR\_DESP\_2000, \pi_{idActor}(\sigma_{añoInicio > 2000} (PARTICIPA\_EN \bowtie SERIE)))$$

$$\rho(ID\_ACTOR\_1\_Y\_2, ID\_ACTOR\_FR\_MAY\_30 \cap ID\_ACTOR\_DESP\_2000)$$

$$\rho(RESULTADO, \pi_{nombreActor}(ID\_ACTOR\_1\_Y\_2 \bowtie ACTOR))$$

# Soluciones - Álgebra Relacional

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además **nunca** participaron en series que comenzaron luego del año 2000

Reutilizaremos la relación *ID\_ACTOR\_FR\_MAY\_30* del ejercicio anterior.

Obtendremos un conjunto con los actores que **si participaron en series que se iniciaron luego del 2000** y luego al conjunto de todos los actores le restaremos este conjunto.

$$\rho(ID\_ACTOR\_DESP\_2000, \pi_{idActor}(\sigma_{añoInicio > 2000}(PARTICIPA\_EN \bowtie SERIE)))$$

$$\rho(ID\_ACTOR\_NO\_DESP\_2000, \pi_{idActor}(ACTOR) - ID\_ACTOR\_DESP\_2000)$$

$$\rho(ID\_ACTOR\_1\_Y\_2, ID\_ACTOR\_FR\_MAY\_30 \cap ID\_ACTOR\_NO\_DESP\_2000)$$

$$\rho(RESULTADO, \pi_{nombreActor}(ID\_ACTOR\_1\_Y\_2 \bowtie ACTOR))$$

# Soluciones - Álgebra Relacional

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los ids de las series que comenzaron más recientemente
- Inicio más reciente indica que son aquellas tales que no haya otra que comenzó después.

$$\rho(SERIE1, SERIE)$$

$$\rho(SERIE2, SERIE)$$

Obtenemos series que no son más recientes y luego restamos

$$\rho(ID\_SERIE\_INICIO\_NO\_MAS\_RECIENTE, \pi_{SERIE1.idSerie}(\sigma_{SERIE1.añoInicio < SERIE2.añoInicio}(SERIE1 \times SERIE2)))$$

$$\rho(RESULTADO, \pi_{idSerie}(SERIE) - ID\_SERIE\_INICIO\_NO\_MAS\_RECIENTE)$$

# Soluciones - Álgebra Relacional

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los ids de los actores que participaron en **al menos 2 series**

$\rho(P1, PARTICIPA\_EN)$

$\rho(P2, PARTICIPA\_EN)$

$\rho(RESULTADO, \pi_{idActor}(\sigma_{P1.idActor=P2.idActor \wedge P1.idSerie \neq P2.idSerie}(P1 \times P2)))$

- Para el hogar: resolver este ejercicio suponiendo que piden devolver los que participaron en **“exactamente 2”** series

# Calculo Relacional de Tuplas - CRT

- El Cálculo Relacional de Tuplas (CRT) es un lenguaje de consulta asociado al Modelo Relacional (MR).
- CRT es un lenguaje **declarativo**: describe cuáles tuplas se deben devolver pero no cómo se calculan
- Cualquier consulta escrita en AR puede también expresarse en CRT y viceversa, o sea, ambos tienen **idéntico poder expresivo**
  - AR sin las operaciones extendidas de agregación y proyección generalizada (no las veremos en la materia)
  - CRT y CRD restringido a expresiones seguras (devuelven un número finito de resultados)
- El cálculo relacional es importante por dos razones:
  - Posee una base muy firme en la lógica matemática
  - El lenguaje de consultas estándar (SQL) para los RDBMSs tiene muchos de sus fundamentos en el CRT

# Repaso: ¿Que es el Cálculo Relacional de Tuplas?

- El CRT se basa sobre la especificación de **variables tupla**.
- Cada variable tupla se extiende a lo largo de una relación y puede tomar como valor cualquier tupla de esa relación
- Una tupla se expresa del siguiente modo:

$$\{t/COND(t)\}$$

donde  $t$  es la tupla y  $COND$  es una expresión booleana que involucra a  $t$ . Ejemplo:

$$\{t/t \in EMPLEADO \wedge t.Sueldo > 50000\}$$

- Una consulta CRT tiene la forma:

$$\{t/F(t)\}$$

donde  $t$  es una tupla (variable libre) y  $F$  es la fórmula que describe a la tupla  $t$ .

- El lenguaje para escribir fórmulas es un subconjunto de la **Lógica de Primer Orden**

## ATENCIÓN

La única variable libre en  $F$  debe ser  $t$ .

# Fórmulas Atómicas

Una fórmula se compone de **fórmulas atómicas** del cálculo de predicados. Si  $R$  es una relación,  $r$  y  $s$  variables de tuplas,  $a$  y  $b$  atributos definidos en  $r$  y  $s$  respectivamente, y  $op$  es un operador del conjunto:  $\{ =, \neq, \leq, \geq, >, < \}$ . Las siguientes son fórmulas atómicas:

- $r \in R$
- $r.a \text{ op } s.b$
- $r.a \text{ op constante}$  o  $\text{constante op } r.a$

Cada una de las fórmulas atómicas anteriores se evalúa como Verdadero o Falso (**Valor de Verdad** para la fórmula atómica)



# Fórmulas

Las fórmulas se definen recursivamente mediante las siguientes reglas:

- cualquier fórmula atómica es una fórmula
- $\neg p$ ,  $p \wedge q$ ,  $p \vee q$ ,  $p \implies q$
- $\exists r(p(r))$  Donde  $r$  es variable de tupla
- $\forall r(p(r))$  Donde  $r$  es variable de tupla

En donde  $p$  y  $q$  son formulas y  $p(r)$  denota una formula en la cual aparece la variable  $r$

# Fórmulas

Las fórmulas se definen recursivamente mediante las siguientes reglas:

- cualquier fórmula atómica es una fórmula
- $\neg p, p \wedge q, p \vee q, p \implies q$
- $\exists r(p(r))$  Donde  $r$  es variable de tupla
- $\forall r(p(r))$  Donde  $r$  es variable de tupla

En donde  $p$  y  $q$  son formulas y  $p(r)$  denota una formula en la cual aparece la variable  $r$  . Ejemplos:

$F_1 : d.\text{Nombre} = \text{'José Perez'}$

$F_2 : ( \exists e)(e \in E \wedge d.\text{Dni} = e.\text{Documento})$

$F_3 : ( \forall d)(d \in D \wedge d.\text{NroLicencia} = \text{'333445555'})$

$d$  es **variable libre** en  $F_1$  y  $F_2$  y **cuantificada** en  $F_3$

# Fórmulas - Ejemplo

Empleado(idEmpleado, nombre, apellido, dirección, idDpto)

Departamento(idDpto, nombreDpto)

Listar nombre, apellido y dirección de todos los empleados que trabajan para el Departamento 'Contable':

# Fórmulas - Ejemplo

Empleado(idEmpleado, nombre, apellido, dirección, idDpto)  
 Departamento(idDpto, nombreDpto)

Listar nombre, apellido y dirección de todos los empleados que trabajan para el Departamento 'Contable':

{t /

# Fórmulas - Ejemplo

Empleado(idEmpleado, nombre, apellido, dirección, idDpto)  
 Departamento(idDpto, nombreDpto)

Listar nombre, apellido y dirección de todos los empleados que trabajan para el Departamento 'Contable':

$\{t \mid \exists e, d$

# Fórmulas - Ejemplo

Empleado(idEmpleado, nombre, apellido, dirección, idDpto)  
 Departamento(idDpto, nombreDpto)

Listar nombre, apellido y dirección de todos los empleados que trabajan para el Departamento 'Contable':

$$\{t \mid \exists e, d \ (e \in \text{Empleado} \wedge d \in \text{Departamento})$$

# Fórmulas - Ejemplo

Empleado(idEmpleado, nombre, apellido, dirección, idDpto)  
 Departamento(idDpto, nombreDpto)

Listar nombre, apellido y dirección de todos los empleados que trabajan para el Departamento 'Contable':

$$\{t \mid \exists e, d \ (e \in \text{Empleado} \wedge d \in \text{Departamento} \wedge e.\text{idDpto} = d.\text{idDpto} \wedge d.\text{nombreDpto} = \text{'Contable'})\}$$

# Fórmulas - Ejemplo

Empleado(idEmpleado, nombre, apellido, dirección, idDpto.)  
 Departamento(idDpto, nombreDpto)

Listar nombre, apellido y dirección de todos los empleados que trabajan para el Departamento 'Contable':

$$\{t \mid \exists e, d \ (e \in \text{Empleado} \wedge d \in \text{Departamento} \wedge e.\text{idDpto} = d.\text{idDpto} \wedge d.\text{nombreDpto} = \text{'Contable'} \wedge t.\text{nombre} = e.\text{nombre} \wedge t.\text{apellido} = e.\text{apellido} \wedge t.\text{dirección} = e.\text{dirección}) \}$$



# Consultas Seguras/Inseguras

## Consultas Inseguras

Es posible definir consultas que devuelvan infinitos resultados las llamamos: **consultas inseguras**.

- Ejemplo:  $\{t / \neg(t \in Actor)\}$

# Consultas Seguras/Inseguras

## Consultas Inseguras

Es posible definir consultas que devuelvan infinitos resultados las llamamos: **consultas inseguras**.

- Ejemplo:  $\{t / \neg(t \in Actor)\}$

Es insegura porque produce todas las tuplas del universo que no son tuplas de actores, que obviamente resultan infinitas

# Consultas Seguras/Inseguras

## Dominio de una Fórmula

Es posible definir el dominio de una fórmula  $F$  que denotamos como  $dom(F)$  y es el conjunto de todos los valores constantes que aparecen en  $F$  y todos los valores de las relaciones a los que hace  $F$  referencia.

# Consultas Seguras/Inseguras

## Dominio de una Fórmula

Es posible definir el dominio de una fórmula  $F$  que denotamos como  $dom(F)$  y es el conjunto de todos los valores constantes que aparecen en  $F$  y todos los valores de las relaciones a los que hace  $F$  referencia.

- Los valores que aparecen explícitamente en  $F$
- Los valores que aparecen en todas las relaciones cuyos nombres aparecen en  $F$
- $dom(t \in Actor \wedge t.edad > 18)$  ???
- $dom(\neg(t \in Actor))$  ???

# Consultas Seguras/Inseguras

## Dominio de una Fórmula

Es posible definir el dominio de una fórmula  $F$  que denotamos como  $dom(F)$  y es el conjunto de todos los valores constantes que aparecen en  $F$  y todos los valores de las relaciones a los que hace  $F$  referencia.

- Los valores que aparecen explícitamente en  $F$
- Los valores que aparecen en todas las relaciones cuyos nombres aparecen en  $F$
- $dom(t \in Actor \wedge t.edad > 18)$  ???
- $dom(\neg(t \in Actor))$  ???

## Ojo!

En ambos casos el dominio son todos los valores de Actor!!! (en el primer caso se agrega además la cte 18)

# CRT: Consultas Seguras/Inseguras

## Consulta Segura

Se dice que una consulta  $\{t/F(t)\}$  es segura si todos los valores que aparecen en el resultado pertenecen al  $dom(F)$

El cálculo relacional de tuplas restringido a expresiones seguras es equivalente en potencia expresiva al álgebra relacional básica

# CRT: Consultas Seguras/Inseguras

## Consulta Segura

Se dice que una consulta  $\{t/F(t)\}$  es segura si todos los valores que aparecen en el resultado pertenecen al  $dom(F)$

El cálculo relacional de tuplas restringido a expresiones seguras es equivalente en potencia expresiva al álgebra relacional básica

## Ojo!

Solo vamos a trabajar con consultas seguras. No tiene sentido trabajar con consultas inseguras.

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)



# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- ¿Cómo obtengo todas las series de un actor A?

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- ¿Cómo obtengo todas las series de un actor A?

$$\{t / \exists p (p \in PARTICIPA\_EN)\}$$

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- ¿Cómo obtengo todas las series de un actor A?

$$\{t/\exists p (p \in PARTICIPA\_EN)\}$$
$$\{t/\exists p (p \in PARTICIPA\_EN \wedge p.idActor = 'A')\}$$

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- ¿Cómo obtengo todas las series de un actor A?

$$\{t/\exists p (p \in PARTICIPA\_EN)\}$$
$$\{t/\exists p (p \in PARTICIPA\_EN \wedge p.idActor = 'A')\}$$
$$\{t/\exists p (p \in PARTICIPA\_EN \wedge p.idActor = 'A' \wedge t.idSerie = p.idSerie)\}$$

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- ¿Cómo lo hago si A es el nombre y no el identificador?

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- ¿Cómo lo hago si A es el nombre y no el identificador?

$$\{t / \exists a, p (a \in ACTOR \wedge p \in PARTICIPA\_EN)\}$$

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- ¿Cómo lo hago si A es el nombre y no el identificador?

$$\{t/\exists a, p (a \in \text{ACTOR} \wedge p \in \text{PARTICIPA\_EN})\}$$

$$\{t/\exists a, p (a \in \text{ACTOR} \wedge p \in \text{PARTICIPA\_EN} \\ \wedge a.\text{nombreActor} = 'A' \wedge a.\text{idActor} = p.\text{idActor})\}$$

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- ¿Cómo lo hago si A es el nombre y no el identificador?

$$\{t/\exists a, p (a \in ACTOR \wedge p \in PARTICIPA\_EN)\}$$

$$\{t/\exists a, p (a \in ACTOR \wedge p \in PARTICIPA\_EN$$

$$\wedge a.nombreActor = 'A' \wedge a.idActor = p.idActor$$

$$\wedge t.idSerie = p.idSerie)\}$$



# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- ¿Y si quiero los nombres de las series en lugar de los identificadores?

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- ¿Y si quiero los nombres de las series en lugar de los identificadores?  
 $\{t / \exists a, s, p (a \in ACTOR \wedge s \in SERIE \wedge p \in PARTICIPA\_EN$

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- ¿Y si quiero los nombres de las series en lugar de los identificadores?

$$\{t / \exists a, s, p (a \in ACTOR \wedge s \in SERIE \wedge p \in PARTICIPA\_EN \\ \wedge a.idActor = p.idActor \wedge s.idSerie = p.idSerie)$$

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- ¿Y si quiero los nombres de las series en lugar de los identificadores?

$$\begin{aligned}
 &\{t / \exists a, s, p (a \in ACTOR \wedge s \in SERIE \wedge p \in PARTICIPA\_EN \\
 &\quad \wedge a.idActor = p.idActor \wedge s.idSerie = p.idSerie \\
 &\quad \wedge a.nombreActor = 'A')
 \end{aligned}$$

# Ejercicios Simples

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- ¿Y si quiero los nombres de las series en lugar de los identificadores?

$$\begin{aligned}
 &\{t / \exists a, s, p (a \in ACTOR \wedge s \in SERIE \wedge p \in PARTICIPA\_EN \\
 &\quad \wedge a.idActor = p.idActor \wedge s.idSerie = p.idSerie \\
 &\quad \wedge a.nombreActor = 'A' \\
 &\quad \wedge t.nombreSerie = s.nombreSerie)\}
 \end{aligned}$$

# Ejercicios

- ❶ Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends”.
- ❷ Listar los nombres de los canales que transmiten todas las series de comedia.
- ❸ Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además participaron alguna vez en series que comenzaron luego del año 2000.
- ❹ Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además nunca participaron en series que comenzaron luego del año 2000.
- ❺ Listar los ids de las series que comenzaron más recientemente.
- ❻ Listar los ids de los actores que participaron en al menos 2 series.

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie "Friends".

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie "Friends".

$$\{t/\exists a, s, p(a \in ACTOR \wedge a.edad > 30 \wedge s \in SERIE \wedge s.nombreSerie = "Friends" \wedge p \in PARTICIPA\_EN \wedge p.idActor = a.idActor \wedge p.idSerie = s.idSerie \wedge t.nombreActor = a.nombreActor)\}$$



# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)

Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)

Género(idGénero, nombreGénero)

Canal(idCanal, nombreCanal)

Participa\_En(idActor, idSerie)

Transmite(idCanal, idSerie)

- Listar los nombres de los canales que transmiten todas las series de comedia.

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los canales que transmiten todas las series de comedia.

$$\{t/\exists c(c \in CANAL \wedge t.nombreCanal = c.nombreCanal \wedge \\
 \forall s(esSerieComedia(s) \implies \exists tra(tra \in TRANSMITE \wedge tra.idCanal = \\
 c.idCanal \wedge tra.idSerie = s.idSerie))))\}$$

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los canales que transmiten todas las series de comedia.

$$\{t/\exists c(c \in CANAL \wedge t.nombreCanal = c.nombreCanal \wedge \forall s(esSerieComedia(s) \implies \exists tra(tra \in TRANSMITE \wedge tra.idCanal = c.idCanal \wedge tra.idSerie = s.idSerie))))\}$$

Definimos *esSerieComedia* como:

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los canales que transmiten todas las series de comedia.

$$\{t/\exists c(c \in CANAL \wedge t.nombreCanal = c.nombreCanal \wedge \forall s(esSerieComedia(s) \implies \exists tra(tra \in TRANSMITE \wedge tra.idCanal = c.idCanal \wedge tra.idSerie = s.idSerie))))\}$$

Definimos *esSerieComedia* como:

$$esSerieComedia(s) = s \in SERIE \wedge \exists g(g \in GENERO \wedge g.idGenero = s.idGenero \wedge g.nombreGenero = "Comedia")$$

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los canales que transmiten todas las series de comedia.

$$\{t/\exists c(c \in CANAL \wedge t.nombreCanal = c.nombreCanal \wedge \forall s(esSerieComedia(s) \implies \exists tra(tras \in TRANSMITE \wedge tras.idCanal = c.idCanal \wedge tras.idSerie = s.idSerie))))\}$$

Definimos *esSerieComedia* como:

$$esSerieComedia(s) = s \in SERIE \wedge \exists g(g \in GENERO \wedge g.idGenero = s.idGenero \wedge g.nombreGenero = "Comedia")$$

- ESTAMOS ASUMIENDO QUE EXISTEN SERIES DE COMEDIA.

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los canales que transmiten todas las series de comedia.

$$\{t/\exists c(c \in \text{CANAL} \wedge t.\text{nombreCanal} = c.\text{nombreCanal} \wedge \forall s(esSerieComedia(s) \implies \exists tra(tra \in \text{TRANSMITE} \wedge tra.\text{idCanal} = c.\text{idCanal} \wedge tra.\text{idSerie} = s.\text{idSerie})))\}$$

Definimos *esSerieComedia* como:

$$esSerieComedia(s) = s \in \text{SERIE} \wedge \exists g(g \in \text{GENERO} \wedge g.\text{idGenero} = s.\text{idGenero} \wedge g.\text{nombreGenero} = \text{"Comedia"})$$

- ESTAMOS ASUMIENDO QUE EXISTEN SERIES DE COMEDIA.

$$\{t/\exists c(c \in \text{CANAL} \wedge t.\text{nombreCanal} = c.\text{nombreCanal} \wedge \forall s(esSerieComedia(s) \implies \exists tra(tra \in \text{TRANSMITE} \wedge tra.\text{idCanal} = c.\text{idCanal} \wedge tra.\text{idSerie} = s.\text{idSerie})) \wedge \exists s'(esSerieComedia(s')) )\}$$

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa-En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además participaron alguna vez en series que comenzaron luego del año 2000.

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa-En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además participaron alguna vez en series que comenzaron luego del año 2000.

$$\{t/\exists a(a \in \text{ACTOR} \wedge \text{mayorde30yEnFriends}(a) \wedge \text{actuoDesp2000}(a) \wedge t.\text{nombreActor} = a.\text{nombreActor})\}$$



# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además participaron alguna vez en series que comenzaron luego del año 2000.

$$\{t/\exists a(a \in \text{ACTOR} \wedge \text{mayorde30yEnFriends}(a) \wedge \text{actuoDesp2000}(a) \wedge t.\text{nombreActor} = a.\text{nombreActor})\}$$

$$\text{mayorde30yEnFriends}(a) = a.\text{edad} > 30 \wedge \exists s, p(s \in \text{SERIE} \wedge s.\text{nombreSerie} = \text{"Friends"} \wedge p \in \text{PARATICIPA\_EN} \wedge p.\text{idActor} = a.\text{idActor} \wedge p.\text{idSerie} = s.\text{idSerie})$$

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además participaron alguna vez en series que comenzaron luego del año 2000.

$$\{t/\exists a(a \in \text{ACTOR} \wedge \text{mayorde30yEnFriends}(a) \wedge \text{actuoDesp2000}(a) \wedge t.\text{nombreActor} = a.\text{nombreActor})\}$$

$$\text{mayorde30yEnFriends}(a) = a.\text{edad} > 30 \wedge \exists s, p(s \in \text{SERIE} \wedge s.\text{nombreSerie} = \text{"Friends"} \wedge p \in \text{PARTICIPA\_EN} \wedge p.\text{idActor} = a.\text{idActor} \wedge p.\text{idSerie} = s.\text{idSerie})$$

$$\text{actuoDesp2000}(a) = \exists s, p(s \in \text{SERIE} \wedge s.\text{añoInicio} > 2000 \wedge p \in \text{PARTICIPA\_EN} \wedge p.\text{idActor} = a.\text{idActor} \wedge p.\text{idSerie} = s.\text{idSerie})$$

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además **nunca** participaron en series que comenzaron luego del año 2000

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además **nunca** participaron en series que comenzaron luego del año 2000

$$\{t/\exists a(a \in ACTOR \wedge \text{mayorde30yEnFriends}(a) \wedge \text{nuncaEnSerieDesp2000}(a) \wedge t.\text{nombreActor} = a.\text{nombreActor})\}$$

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los nombres de los actores mayores de 30 años que participan en la serie “Friends” y que además **nunca** participaron en series que comenzaron luego del año 2000

$$\{t/\exists a(a \in \text{ACTOR} \wedge \text{mayorde30yEnFriends}(a) \wedge \text{nuncaEnSerieDesp2000}(a) \wedge t.\text{nombreActor} = a.\text{nombreActor})\}$$

$$\text{nuncaEnSerieDesp2000}(a) = \neg \exists s(s \in \text{SERIE} \wedge s.\text{añoInicio} > 2000 \wedge \exists p(p \in \text{PARTICIPA\_EN} \wedge p.\text{idActor} = a.\text{idActor} \wedge p.\text{idSerie} = s.\text{idSerie}))$$

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los ids de las series que comenzaron más recientemente.

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los ids de las series que comenzaron más recientemente.

$$\{t / \exists s (s \in \text{Serie} \wedge t.\text{idSerie} = s.\text{idSerie} \wedge \text{inicioMasReciente}(s))\}$$

- Inicio más reciente indica que son aquellas tales que no haya otra que comenzó después.

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los ids de las series que comenzaron más recientemente.

$$\{t / \exists s (s \in \text{Serie} \wedge t.\text{idSerie} = s.\text{idSerie} \wedge \text{inicioMasReciente}(s))\}$$

- Inicio más reciente indica que son aquellas tales que no haya otra que comenzó después.

$$\text{inicioMasReciente}(s) = \neg \exists s' (s' \in \text{SERIE} \wedge s'.\text{añoInicio} > s.\text{añoInicio})$$



# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los ids de los actores que participaron en **al menos 2 series**.

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los ids de los actores que participaron en **al menos 2 series**.

$$\{t / \exists a(a \in ACTOR \wedge t.idActor = a.idActor \wedge partEnAlMenosDosSeries(a))\}$$

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
Género(idGénero, nombreGénero)  
Canal(idCanal, nombreCanal)  
Participa\_En(idActor, idSerie)  
Transmite(idCanal, idSerie)

- Listar los ids de los actores que participaron en **al menos 2 series**.

$\{t/\exists a(a \in ACTOR \wedge t.idActor = a.idActor \wedge partEnAlMenosDosSeries(a))\}$

- Debería haber 2 tuplas diferentes en *PARTICIPA\_EN* para el mismo actor

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los ids de los actores que participaron en **al menos 2 series**.

$$\{t/\exists a(a \in ACTOR \wedge t.idActor = a.idActor \wedge partEnAlMenosDosSeries(a))\}$$

- Debería haber 2 tuplas diferentes en *PARTICIPA\_EN* para el mismo actor

$$partEnAlMenosDosSeries(a) = \exists p, p'(p \in PARTICIPA\_EN \wedge p' \in PARTICIPA\_EN \wedge p' \neq p \wedge p.idActor = a.idActor \wedge p'.idActor = a.idActor)$$

# Soluciones - Cálculo Relacional de Tuplas

Actor(idActor, nombreActor, edad)  
 Serie(idSerie, nombreSerie, idGénero, añoInicio, añoFin)  
 Género(idGénero, nombreGénero)  
 Canal(idCanal, nombreCanal)  
 Participa\_En(idActor, idSerie)  
 Transmite(idCanal, idSerie)

- Listar los ids de los actores que participaron en **al menos 2 series**.

$$\{t/\exists a(a \in ACTOR \wedge t.idActor = a.idActor \wedge partEnAlMenosDosSeries(a))\}$$

- Debería haber 2 tuplas diferentes en *PARTICIPA\_EN* para el mismo actor

$$partEnAlMenosDosSeries(a) = \exists p, p'(p \in PARTICIPA\_EN \wedge p' \in PARTICIPA\_EN \wedge p' \neq p \wedge p.idActor = a.idActor \wedge p'.idActor = a.idActor)$$

- Para el hogar: Resolver este ejercicio suponiendo que piden devolver los que participaron en **“exactamente 2”** series

# Bibliografía principal

- Ramez Elmasri and Shamkant B. Navathe. 2016. **Fundamentals of Database Systems (7th ed.)**. Pearson.
- Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. 2008. **Database Systems: The Complete Book (2nd ed.)**. Prentice Hall Press, Upper Saddle River, NJ, USA.
- Jeffrey D. Ullman **Principles of Database and Knowledge-Base Systems** Computer Science Press; 1st edition (January 1, 1990)
- CARTILLAS DE NOTACION DE LA MATERIA