

## CAPITULO III

### MODELO Y LENGUAJES RELACIONALES

#### 3.1 Modelo Relacional

- Propuesto por E. F. Codd, en 1970.
- Tiene sólidos fundamentos teóricos.
- Está basado en una estructura de datos simple y uniforme, la **relación**.

#### Ejemplo de una relación

Rows                      Columns

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	31-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

**Tabla o relación EMP**

- **Principales ventajas del modelo relacional:**

- ♦ Simplicidad.
- ♦ Base matemática (teoría de conjuntos).
- ♦ Soporte de lenguajes declarativos.
- ♦ Facilidad de combinación de operaciones.

### 3.1.1 Conceptos, Terminología, Propiedades

- **Relación:** Subconjunto del producto cartesiano de uno o más dominios.

- **Dominio:** Conjunto de valores.

**Ejemplo:** Cardinales (azul, blanco, verde)

- $D_1 \times D_2 \times \dots \times D_k$  es el **producto cartesiano**

- ♦ dados los dominios  $D_i$   $i = 1, k$
- ♦ corresponde al conjunto de todas las  $k$ -tuplas

$(V_1, V_2, \dots, V_k)$  tal que:

$$V_1 \in D_1$$

$$V_2 \in D_2$$

$$V_k \in D_k$$

- Las relaciones son finitas (pueden ser vacías).

### **Ejemplo:**

$$D_1 = \{\text{manzana, naranja, pera}\}$$

$$D_2 = \{\text{rojo, amarillo, anaranjado}\}$$

$$\begin{aligned} D_1 \times D_2 = \{ & (\text{manzana, rojo}), (\text{manzana, amarillo}), \\ & (\text{manzana, anaranjado}), (\text{naranja, rojo}), (\text{naranja,} \\ & \text{amarillo}), (\text{naranja, anaranjado}), (\text{pera, rojo}), \\ & (\text{pera, amarillo}), (\text{pera, anaranjado}) \} \end{aligned}$$

$$\begin{aligned} R_1 = \{ & (\text{manzana, rojo}), (\text{pera, amarillo}), \\ & (\text{naranja, anaranjado}) \} \subseteq D_1 \times D_2 \end{aligned}$$

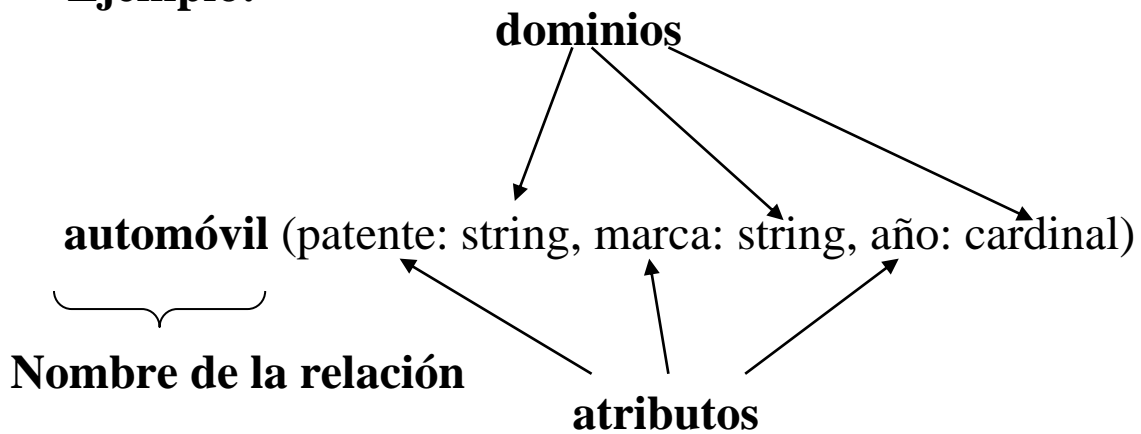
$(\text{manzana, rojo}) \in R_1$ , se denomina **tupla**.

- Una relación puede ser visualizada como una **tabla**:
  - ♦ Cada fila representa una **tupla**.
  - ♦ En cada columna hay una componente.
- Los nombres de las columnas se denominan **atributos**.

- El conjunto de atributos de una relación es el **Esquema de la Relación**.

$$R (A_1, A_2, \dots, A_k)$$

**Ejemplo:**



**Ejemplo:**

**Fruta (nombre, color)      Esquema de la relación**

nombre	color	} Conjunto de tuplas
manzana	rojo	
pera	amarillo	
naranja	anaranjado	

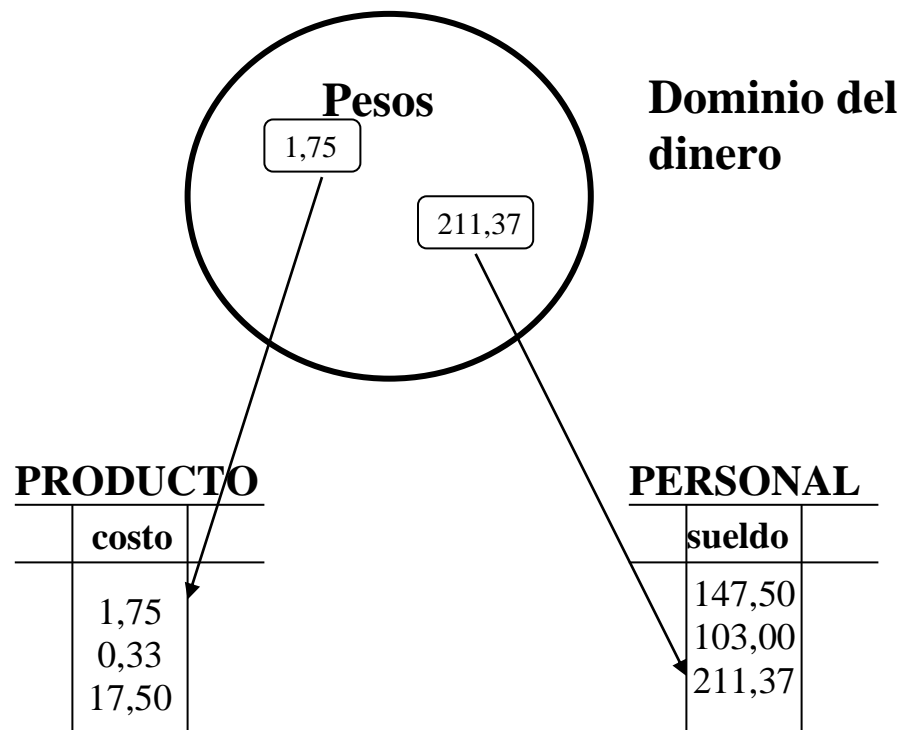
- El orden no es importante  $\Rightarrow$  diferencias respecto al modelo matemático de  $R \subseteq D_1 \times D_2 \times \dots \times D_k$ .

color	nombre
rojo	manzana
amarillo	pera
anaranjado	naranja

$(\text{rojo}, \text{manzana}) \equiv (\text{manzana}, \text{rojo})$

- **Concepto de dominio**

Los conceptos **atributo** y **dominio** son distintos.



Los atributos costo (PRODUCTO) y sueldo (PERSONAL) toman sus valores del dominio del Dinero.

## Terminología

- Relación: Tabla.
- Atributo: Columna.
- Tupla: Fila.
- Grado: Número de columnas en la tabla.
- Cardinalidad: Número de filas de la tabla.

Formal	Informal
Relación	Tabla
Tupla	Fila
Atributo	Columna

**Grado de una relación (Degree), “Aridad”:** Número de atributos (número de columnas de la tabla).

**Cardinalidad de una relación (Cardinality):** Número de tuplas (número de filas de la tabla).

## Propiedades de una relación

- a) En un sistema relacional, la tabla puede contener **sólo un tipo de registro** (tupla). Cada tupla tiene un número fijo de atributos, todos ellos con un nombre explícito.
- b) Dentro de la tabla, cada atributo es distinto y el valor asociado a cada instancia del atributo **debe ser simple** (es decir, es un **único valor**).
- c) Cada tupla de una relación es **única**, no hay tuplas duplicadas.
- d) El orden de las tuplas dentro de la tabla no está determinado.
- e) Para cada atributo definido, existe un conjunto de valores que forman el **dominio**.
- f) Se pueden crear nuevas tablas, relacionando valores de atributos procedentes de tablas ya existentes.

### 3.1.2 Restricciones en el Modelo Relacional

#### Claves (Keys):

- **Superclave:** Es un atributo o un conjunto de atributos que identifica a una única fila dentro de una tabla. No exige la característica de ser mínima.
- **Clave candidato:** Es un atributo o conjunto mínimo de atributos que identifica a una única fila dentro de la tabla.
- **Clave primaria (Primary Key):** Es una clave candidato que es escogida por el diseñador de la base de datos para identificar a las filas dentro de una tabla.
  - ◆ **Definición informal:** Es un atributo o una combinación de atributos de la tabla que puede ser usado como un identificador único para los registros de la tabla.



♦ **Definición formal:** Es un conjunto de atributos  $K_1, K_2, \dots, K_n$  de  $T$  ( $n > 0$ ) que tiene dos propiedades independientes del tiempo:

a) **Unicidad (Uniqueness).** En cualquier tiempo, no pueden existir dos registros distintos de  $T$  que tengan el mismo valor para  $K_1$ , el mismo valor para  $K_2, \dots$ , y el mismo valor para  $K_n$ .

b) **Minimalidad (Minimality).** Ninguno de los  $K_1, K_2, \dots, K_n$  puede ser descartado de  $K$  sin destruir la propiedad de unicidad.

- **Clave foránea (Foreign Key):** Es un atributo o combinación de atributos de una tabla, que es la clave primaria de otra tabla.

### **Ejemplo: Base de datos PARTS – SUPPLIERS**

**S** (S#, SNAME, STATUS, CITY)

**P** (P#, PNAME, COLOR, WEIGHT, CITY)

**SP** (S#, P#, QTY)      FK1: S#      FK2: P#

## Reglas de integridad

Todas las inserciones, actualizaciones y eliminaciones a bases de datos relacionales están restringidas por las siguientes dos reglas:

b) **Integridad de entidades (Entity integrity):** Una clave primaria no puede ser nula.

c) **Integridad referencial (Referential integrity):** La clave foránea puede ser nula o tener un valor que se refiere a un valor válido, que se encuentra presente como una clave primaria en alguna otra tabla.

Es decir, si una relación base R2 incluye una clave extranjera FK correspondiente a la clave primaria PK de alguna relación base R1, entonces cada valor de FK en R2 debe ser:

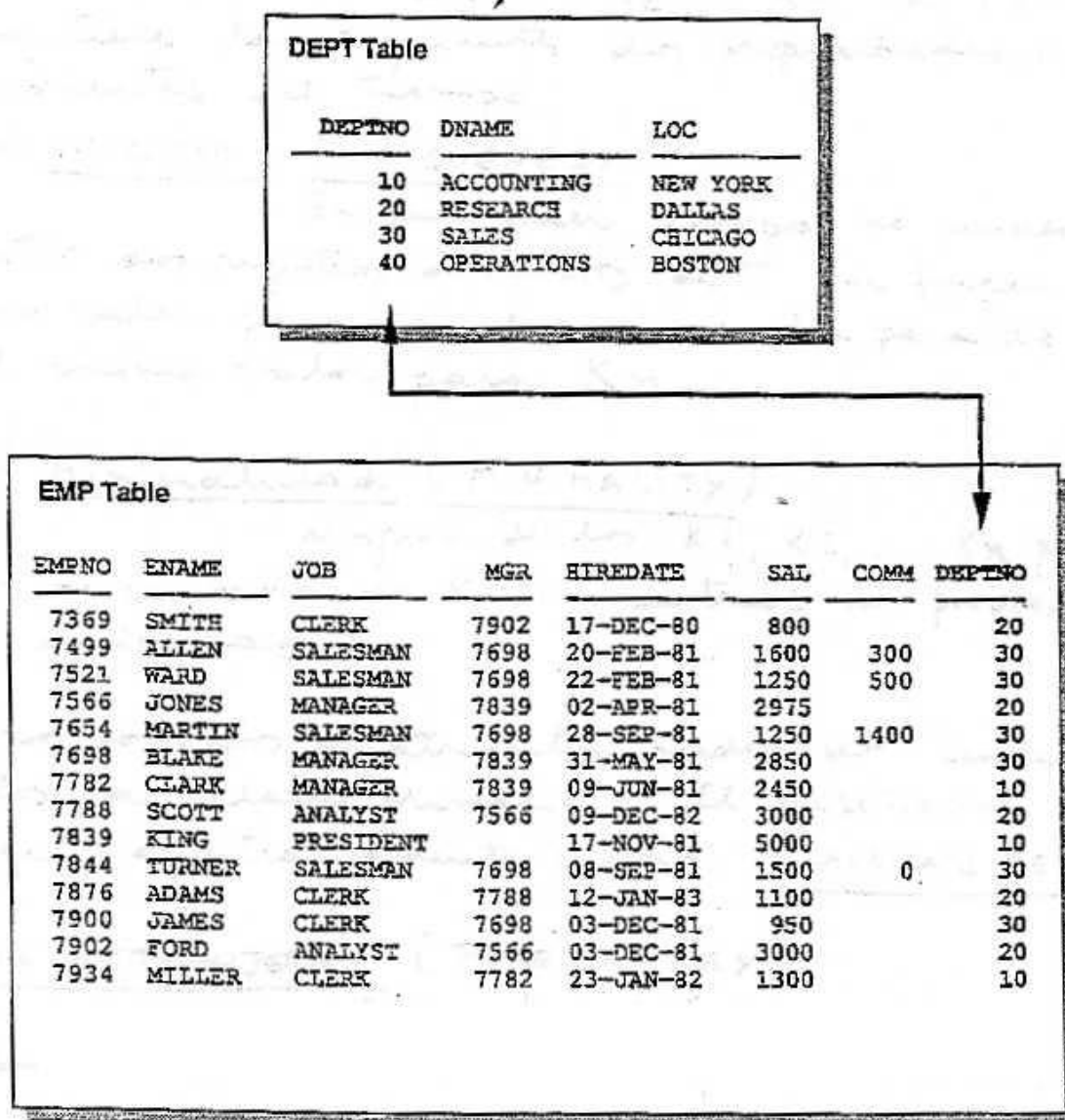
i. Igual al valor de PK en alguna tupla de R1.

ó

ii. Ser completamente nula (es decir, cada valor asociado a los atributos que participan en el valor de FK deben ser nulos).

## Ejemplo de integridad referencial

### Tablas EMP y DEPT



En valor del atributo DEPTNO de la tabla EMP para alguna tupla podría ser null. Esto significa que el empleado no está asociado a ningún departamento

## Ejemplos de restricciones de integridad

Cada valor en la columna DNAME debe ser único.

Table DEPT

DEPTNO	DNAME	LOC
20	RESEARCH	DALLAS
30	SALES	CHICAGO

Cada fila debería tener un valor para la columna ENAME.

Cada valor en la columna DEPTNO debe corresponder a un valor en la columna DEPTNO de la tabla DEPT.

Table EMP

EMPNO	ENAME	.. Other Columns ..	SAL	COMM	DEPTNO
7329	SMITH		9,000.00		20
7499	ALLEN		7,500.00	100.00	30
7521	WARD		5,000.00	200.00	30
7566	JONES		2,975.00	400.00	30

Cada fila debería tener un valor para la columna EMPNO, y el valor debe ser único.

Cada valor en la columna SAL debe ser menor que 10,000

## Reglas para mantener la integridad referencial

### Opciones:

- CASCADES
- RESTRICTED
- NULLIFIES
- DEFAULTS

### Especificación de integridad referencial en el caso de “On Delete”

### Ejemplo:

Table Name: EMPLOYEE			
Column Name	EMP_NO	EMP_NAME	EMP_DEPT_NO
Key Type	PK		FK
Nulls/Unique	NN, U	NN	
Sample Data	100	Smith	10
	310	Adams	30
	210	Brown	20
	405	Gomez	—
	378	Johnson	50

Table Name: DEPARTMENT		
Column Name	DEPT_NO	DEPT_NAME
Key Type	PK	
Nulls/Unique	NN, U	NN
Sample Data	10	Accounting
	20	Research
	30	Sales
	40	Finance
	50	Operations

Opción	Explicación de la restricción
<b>CASCADE</b>	La eliminación debería ser en cascada donde se produzca coincidencia con los empleados. Las filas de la tabla EMPLOYEE que coinciden también debieran ser borradas. <b>Ejemplo:</b> Si elimino el departamento 20, debieran eliminarse automáticamente los empleados del departamento 20
<b>RESTRICTED</b>	El borrado debiera restringirse sólo para los departamentos sin empleados.
<b>NULLIFY</b>	La clave foránea debiera quedar en nulo (válida sólo para los FK's que permitan valores nulos) cuando se elimina la PK referenciada.
<b>DEFAULT</b>	La clave foránea queda con un valor definido previamente.

**Restricción de eliminación utilizando las tablas  
EMPLOYEE y DEPARTMENT**

## Especificación de integridad referencial en el caso de “On Update”

**Ejemplo:**

Table Name: EMPLOYEE			
Column Name	EMP_NO	EMP_NAME	EMP_DEPT_NO
Key Type	PK		FK
Nulls/Unique	NN, U	NN	
Sample Data	100	Smith	10
	310	Adams	30
	210	Brown	20
	405	Gomez	—
	378	Johnson	50

Table Name: DEPARTMENT		
Column Name	DEPT_NO	DEPT_NAME
Key Type	PK	
Nulls/Unique	NN, U	NN
Sample Data	10	Accounting
	20	Research
	30	Sales
	40	Finance
	50	Operations

Opción	Explicación la de restricción
<b>CASCADE</b>	La actualización debería ser en cascada donde se produzca coincidencia con los empleados. Las filas de la tabla EMPLOYEE que coinciden también debieran ser actualizadas para reflejar el nuevo valor PK.
<b>RESTRICTED</b>	La actualización debiera restringirse sólo para los departamentos sin empleados.
<b>NULLIFY</b>	La clave foránea debiera ser nula (válida sólo para las FK's que permitan valores nulos) cuando se actualiza la PK referenciada.

**Restricción de actualización utilizando las tablas  
EMPLOYEE y DEPARTMENT**



## **Definición del esquema para COMPANY con SQL, sin manejo de la integridad referencial**

### **CREATE TABLE EMPLOYEE**

```
( FNAME          VARCHAR(15)  NOT NULL,  
  MINIT          CHAR,  
  LNAME          VARCHAR(15)  NOT NULL,  
  SSN            CHAR(9)      NOT NULL,  
  BDATE          DATE,  
  ADDRESS        VARCHAR(30),  
  SEX            CHAR,  
  SALARY         DECIMAL(10,2),  
  SUPERSSN       CHAR(9),  
  DNO            INT          NOT NULL,
```

**PRIMARY KEY (SSN),**

**FOREIGN KEY (SUPERSSN) REFERENCES EMPLOYEE(SSN),**

**FOREIGN KEY (DNO) REFERENCES**

**DEPARTMENT(DNUMBER) );**

**CREATE TABLE DEPARTMENT**

```
( DNAME          VARCHAR(15)  NOT NULL,  
  DNUMBER        INT          NOT NULL,  
  MGRSSN         CHAR(9)      NOT NULL,  
  MGRSTARTDATE   DATE,  
PRIMARY KEY (DNUMBER),  
UNIQUE (DNAME),  
FOREIGN KEY (MGRSSN) REFERENCES  
                                EMPLOYEE(SSN) );
```

**CREATE TABLE DEPT\_LOCATIONS**

```
( DNUMBER        INT          NOT NULL,  
  DLOCATION        VARCHAR(15)  NOT NULL,  
PRIMARY KEY (DNUMBER, DLOCATION),  
FOREIGN KEY (DNUMBER) REFERENCES  
                                DEPARTMENT(DNUMBER) );
```

**CREATE TABLE PROJECT**

```
( PNAME          VARCHAR(15)  NOT NULL,  
  PNUMBER        INT          NOT NULL,  
  PLOCATION       VARCHAR(15)  
  DNUM           INT          NOT NULL,  
PRIMARY KEY (PNUMBER),  
UNIQUE (PNAME),  
FOREIGN KEY (DNUM) REFERENCES  
                                DEPARTMENT(DNUMBER) );
```

**CREATE TABLE WORKS\_ON**

```
( ESSN           CHAR(9)      NOT NULL,  
  PNO            INT          NOT NULL,  
  HOURS          DECIMAL(3,1) NOT NULL,  
PRIMARY KEY (ESSN, PNO),  
FOREIGN KEY (ESSN) REFERENCES  
                                EMPLOYEE(SSN),  
FOREIGN KEY (PNO) REFERENCES  
                                PROJECT (PNUMBER) );
```

**CREATE TABLE DEPENDENT**

```
( ESSN                CHAR(9)          NOT NULL,  
  DEPENDENT_NAME     VARCHAR(15) NOT NULL,  
  SEX                CHAR,  
  BDATE              DATE,  
  RELATIONSHIP        VARCHAR(8),  
PRIMARY KEY (ESSN, DEPENDENT_NAME),  
FOREIGN KEY (ESSN) REFERENCES  
                                EMPLOYEE(SSN) );
```

## Esquema de COMPANY en SQL con manejo de la integridad referencial

**CREATE TABLE EMPLOYEE**

( ...,

DNO INT NOT NULL DEFAULT 1,

**CONSTRAINT EMPPK**

PRIMARY KEY (SSN),

**CONSTRAINT EMPSUPERFK**

FOREIGN KEY (SUPERSSN) REFERENCES

EMPLOYEE(SSN)

ON DELETE SET NULL ON UPDATE CASCADE,

**CONSTRAINT EMPDEPTFK**

FOREIGN KEY (DNO) REFERENCES

DEPARTMENT(DNUMBER)

ON DELETE SET DEFAULT ON UPDATE CASCADE );

**CREATE TABLE DEPARTMENT****(...,****MGRSSN CHAR(9) NOT NULL DEFAULT “8886655555”****CONSTRAINT DEPTPK****PRIMARY KEY (DNUMBER),****CONSTRAINT DEPTSK****UNIQUE (DNAME),****CONSTRAINT DEPTMGRFK****FOREIGN KEY (MGRSSN) REFERENCES EMPLOYEE(SSN)****ON DELETE SET DEFAULT ON UPDATE CASCADE;****CREATE TABLE DEPT\_LOCATIONS****(...,****PRIMARY KEY (DNUMBER, LOCATION),****FOREIGN KEY (DNUMBER) REFERENCES****DEPARTMENT(DNUMBER)****ON DELETE CASCADE ON UPDATE CASCADE);**

### 3.1.3 Documentación del Modelo Relacional

#### Formato para describir una tabla

##### TABLE INSTANCE CHARTS

Table Name:

Column Name							
Key Type							
Nulls/ Unique							
Sample Data							

Table Name:

Column Name							
Key Type							
Nulls/ Unique							
Sample Data							

Table Name:

Column Name							
Key Type							
Nulls/ Unique							
Sample Data							

## Descripción de la tabla EMPLOYEE

Table Name: EMPLOYEE

Column Name	EMPNO	FNAME	LNAME	JOB	HIREDATE	SAL	COMM	MGR	DEPTNO
Key Type	PK							FK1	FK2
Nulls/Unique	NN, U	NN	NN		NN				NN
Sample Data	7369	MARY	SMITH	CLERK	17-DEC-80	800		7902	20
	7902	HENRY	FORD	ANALYST	03-DEC-81	3000		7566	50
	7521	SUE	WARD	SALESMAN	22-FEB-81	1250	6000	7698	30
	7698	BOB	BLAKE	MANAGER	01-MAY-81	2850	10000	7839	30
	7839	BOB	KING	PRESIDENT	17-NOV-81	5000	5000		10



## Documentación

### a) Tablas

<b>Nombre Tabla:</b>			
<b>Descripción:</b>			
<b>Columna</b>	<b>Tipo de Dato</b>	<b>Primary Key / Foreign Key</b>	<b>Nulls / Unique</b>

Donde cada campo tiene los siguientes significados:

- Tipo de Dato: Indicar el tipo de dato (por ejemplo: String, Integer, Date)
- Indicar si es “Primary Key” (PK) o “Foreign Key” (FK). Si se encuentra más de un FK en la entidad se deberá indicar numéricamente la secuencia de la siguiente forma: FK<sub>1</sub>, FK<sub>2</sub>, FK<sub>3</sub>, etc.
- Indicar restricción del tipo Null o Unique, asociada al atributo

### b) Restricciones generales asociadas al modelo

## Ejemplo de Documentación del Modelo Relacional

<b>Nombre Tabla:</b> ALUMNO			
<b>Descripción:</b> Contiene los datos de los alumnos			
<b>Columna</b>	<b>Tipo de Dato</b>	<b>Primary Key / Foreign Key</b>	<b>Nulls / Unique</b>
rut	String	PK	NN, U
fechaNacimiento	Date		NN
codigoCarrera	String	FK	NN

<b>Nombre Tabla:</b> CARRERA			
<b>Descripción:</b> Contiene los datos de las carreras			
<b>Columna</b>	<b>Tipo de Dato</b>	<b>Primary Key / Foreign Key</b>	<b>Nulls / Unique</b>
codigoCarrera	String	PK	NN, U
nomCarrera	String		NN

### Restricciones generales

El código de la carrera comienza con 2 caracteres

### **3.1.4 Diseño de bases de datos relacionales, utilizando una transformación del M – E – R.**

#### **Transformación del modelo Entidad–Relacionamiento al Modelo Relacional**

1. Para cada entidad regular E en el esquema E – R, se debe crear una relación R, que incluya todos los atributos simples de E.

Incluir sólo los atributos simples que constituyan un atributo compuesto.

Escoger alguno de los atributos claves de E, como la clave primaria de R.

2. Para cada entidad débil W en el esquema E – R, que tiene un dueño E, crear una relación R; incluir todos los atributos simples (o los componentes simples de un atributo compuesto) de W como atributos de R.

Además, incluir como clave foránea de R, los atributos que componen la clave primaria de E.

La clave primaria de R es la combinación de la clave primaria de su dueño y de la clave parcial de la entidad débil W.

3. Por cada relacionamiento binario 1:1, R, en el esquema  $E - R$ , identificar las relaciones S y T, que corresponden a las entidades que participan en R.

Escoger una de las relaciones, por ejemplo S, e incluir como clave foránea en S, la clave primaria de T.

Se deben incluir todos los atributos simples (o los componentes simples de un atributo compuesto) del relacionamiento 1:1 R, como atributos de la relación S.

4. Para cada relacionamiento binario, 1:N, entre las entidades T y S y regular (no débil), R, identificar la relación S, que representa el lado “N” del relacionamiento, es decir, una tupla de T está relacionado con muchas de S y una de S está relacionada con una sola de T

Incluir como clave foránea en S, la clave primaria de la relación T, que representa la otra entidad que participa en el relacionamiento R.

Incluir todos los atributos simples (o los componentes simples de un atributo compuesto) del relacionamiento 1:N, como atributos de la relación S.

**5.** Por cada relacionamiento binario  $M:N$ ,  $R$ , en el esquema  $E - R$ , se debe crear una nueva relación  $S$ , para representar a  $R$ .

Se deben incluir como claves foráneas en  $S$ , las claves primarias de las relaciones que representan a las entidades que participan en el relacionamiento.

La combinación de estas claves primarias, formará la clave primaria de la relación  $S$ .

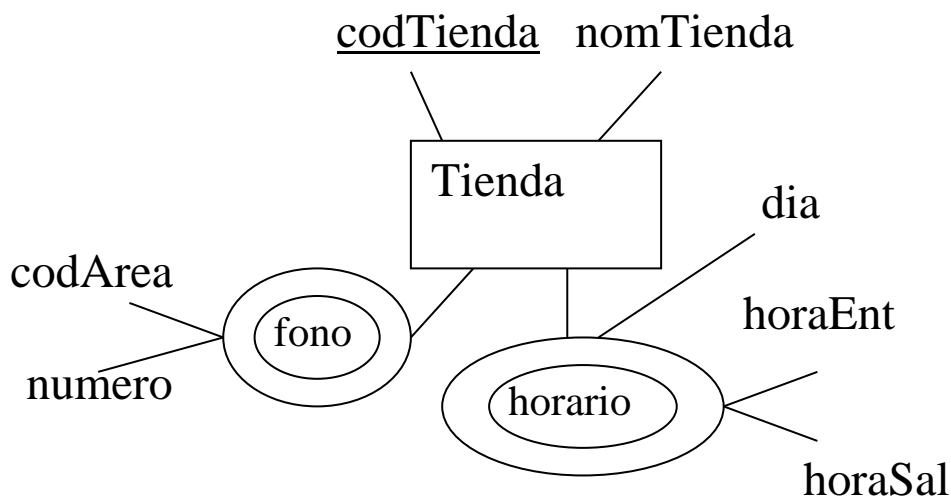
Incluir todos los atributos simples (o los componentes simples de un atributo compuesto) del relacionamiento  $N:M$ , como atributos de la relación  $S$ .

**6.** Para cada atributo multivaluado  $A$ , se debe crear una nueva relación  $R$ , que incluya un atributo correspondiente a  $A$ , más la clave primaria  $K$  (como una clave foránea en  $R$ ) de la relación que representa a la entidad o relacionamiento, que tiene a  $A$  como un atributo.

La clave primaria de  $R$ , es la combinación de  $A$  y  $K$ .

Si el atributo multivaluado es compuesto, se incluyen sus componentes simples y, en la mayoría de los casos, la clave primaria será la combinación de K más los componentes simples de A.

### Ejemplo



**Tienda** (codTienda, nomTienda)

**TiendaFono** (codTienda codArea, numero)

*Ambos, codArea y numero, deben ser parte de la clave, ya que la tienda podría tener varios teléfonos pertenecientes a la misma área*

**TiendaHorario** (codTienda, dia, horaEnt, horaSal)

*No es necesario que horaSal sea parte de la clave, ya que en un día en particular, por ejemplo, el lunes tiene en la mañana un horario de entrada (9:00 hrs.) y un horario de salida (14:00 hrs.) y en la tarde otro horario de entrada (17:00 hrs.), que nunca se intersectará con el horario de entrada de la mañana), y otro de salida (21:00 hrs.)*

7. Para cada relacionamiento n-ario  $R$ , con  $n$  mayor que 2, se debe crear una nueva relación  $S$ , para representar  $R$ .

Se deben incluir como claves foráneas en  $S$ , las claves primarias de las relaciones que representan a las entidades que participan en el relacionamiento.

La combinación de estas claves primarias, será usualmente la clave primaria de la relación  $S$ .

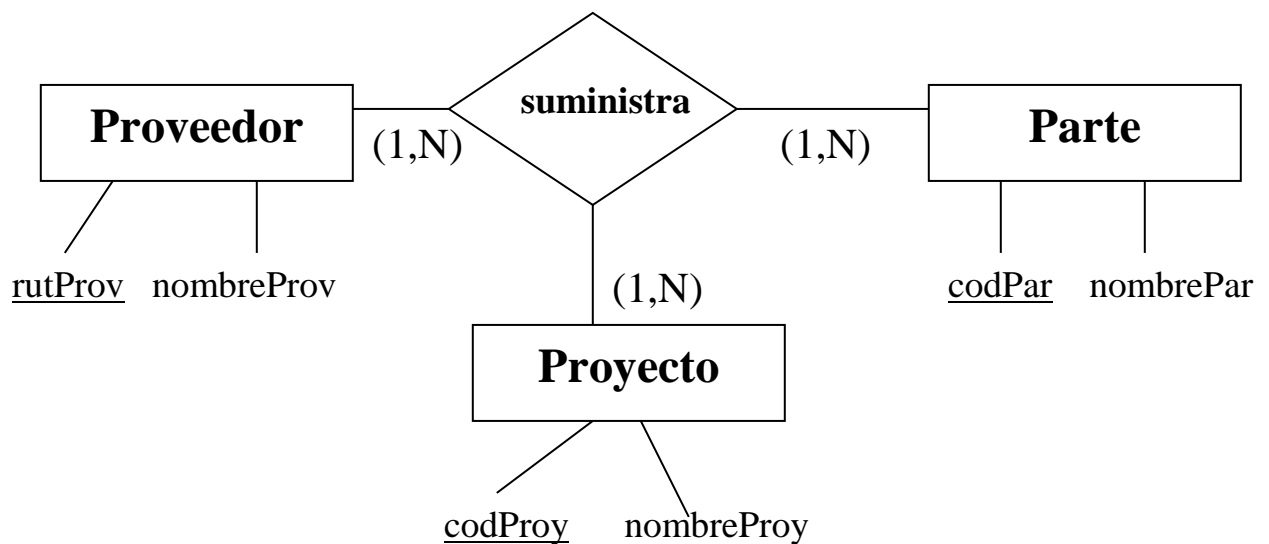
Incluir todos los atributos simples (o los componentes simples de un atributo compuesto) del relacionamiento  $N:M$ , como atributos de la relación  $S$ .

Para un relacionamiento ternario, la **clave primaria de la correspondiente tabla** es:

- Si el relacionamiento es  $N:N:N$ , la clave primaria está compuesta de los 3 atributos
- Si el relacionamiento es  $1:1:1$ , la clave primaria es cualquier combinación de 2 atributos
- Si el relacionamiento es  $1:1:N$ , la clave primaria está compuesta de 2 atributos:
  - La clave de la entidad donde está la cardinalidad  $N$

- Cualquiera de las claves de las otras entidades (que tienen cardinalidad 1)
- Si el relacionamiento es 1:N:N, la clave primaria está compuesta de 2 atributos, aquellos que son la clave de las entidades donde está la cardinalidad N

### Ejemplo



**Proveedor** (rutProv, nombreProv)

**Parte** (codPar, nombrePar)

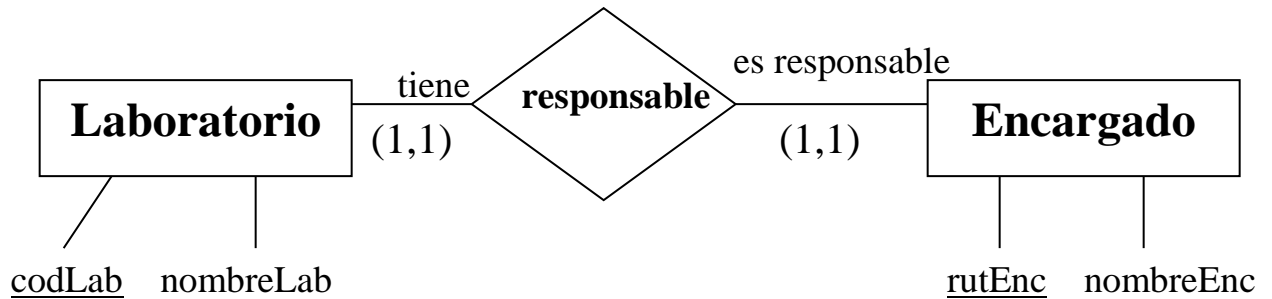
**Proyecto** (codProy, nombreProy)

**suministra** (rutProv, codPar, codProy)



## Ejemplos de transformación al modelo relacional

### (1) Relacionamiento 1:1



**Laboratorio** (codLab, nombreLab, rutEnc)

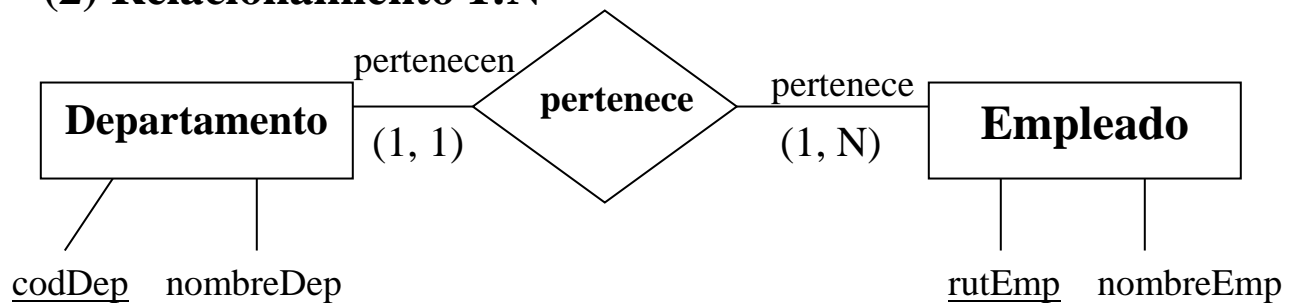
**Encargado** (rutEnc, nombreEnc)

ó

**Laboratorio** (codLab, nombreLab)

**Encargado** (rutEnc, nombreEnc, codLab)

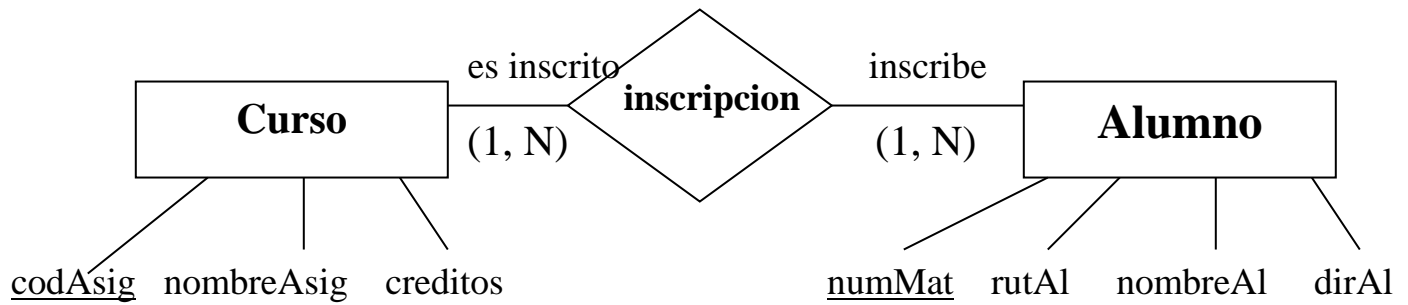
### (2) Relacionamiento 1:N



**Departamento** (codDep, nombreDep)

**Empleado** (rutEmp, nombreEmp, codDep)

### (3) Relacionamiento M:N

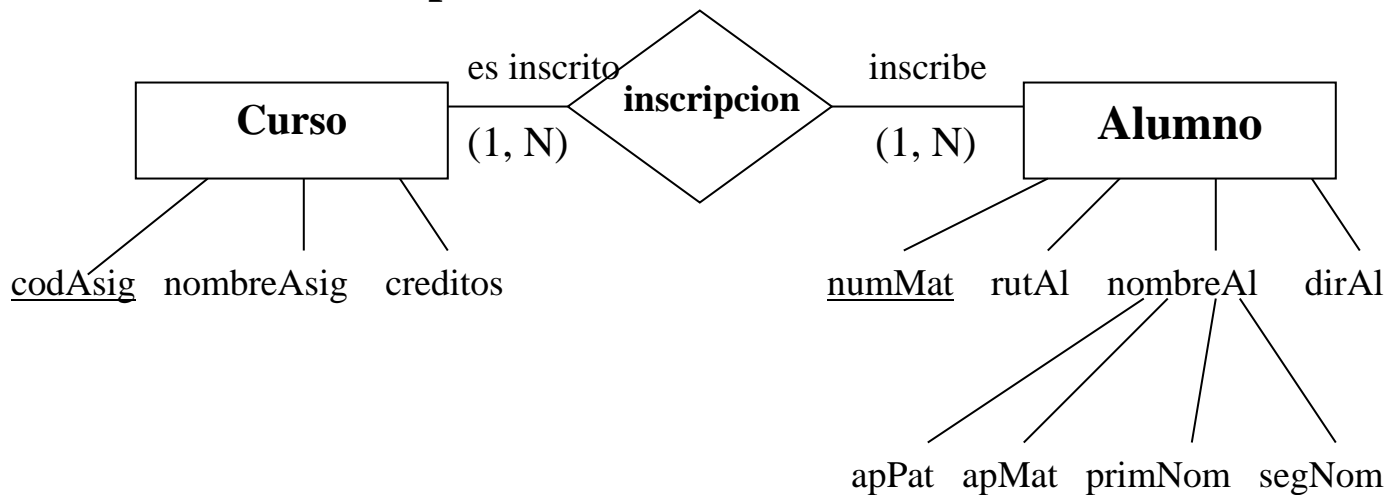


**Curso** (codAsig, nombreAsig, creditos)

**Alumno** (numMat, rutAl, nombreAl, dirAl)

**inscripción** (codAsig, numMat)

### (4) Atributo Compuesto

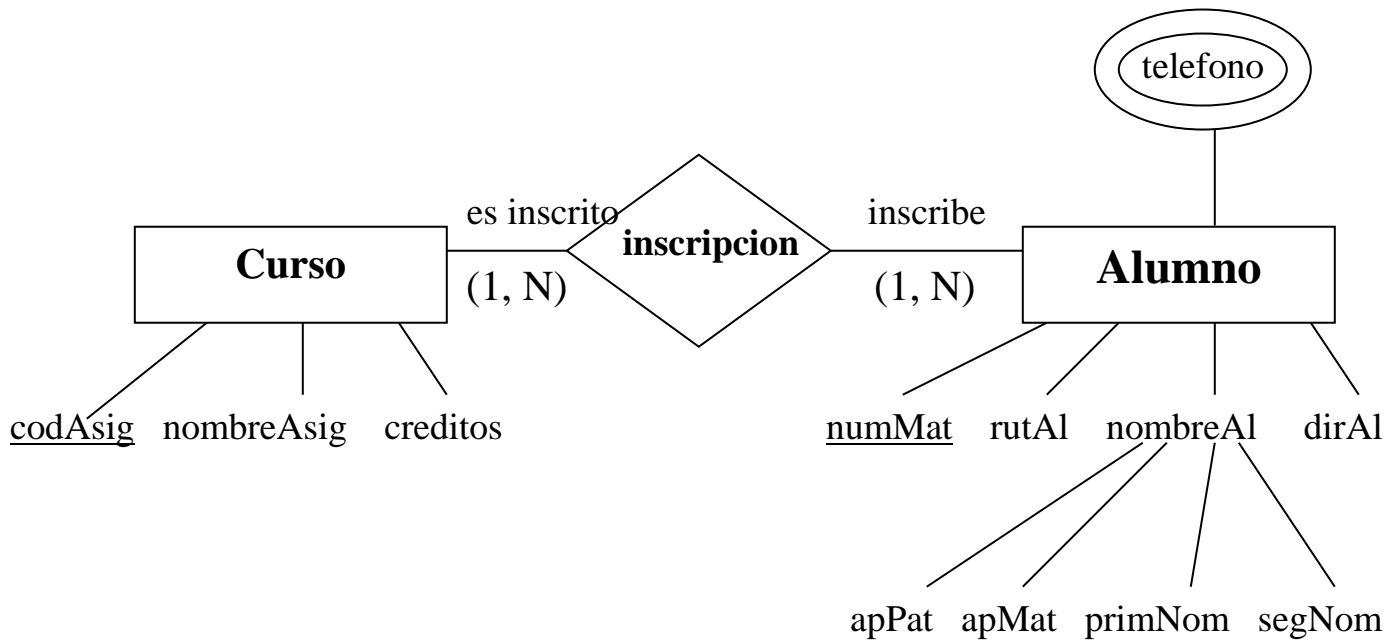


**Curso** (codAsig, nombreAsig, creditos)

**Alumno** (numMat, rutAl, apPat, apMat, primNom, segNom, dirAl)

**inscripción** (codAsig, numMat)

## (5) Atributo Multivaluado



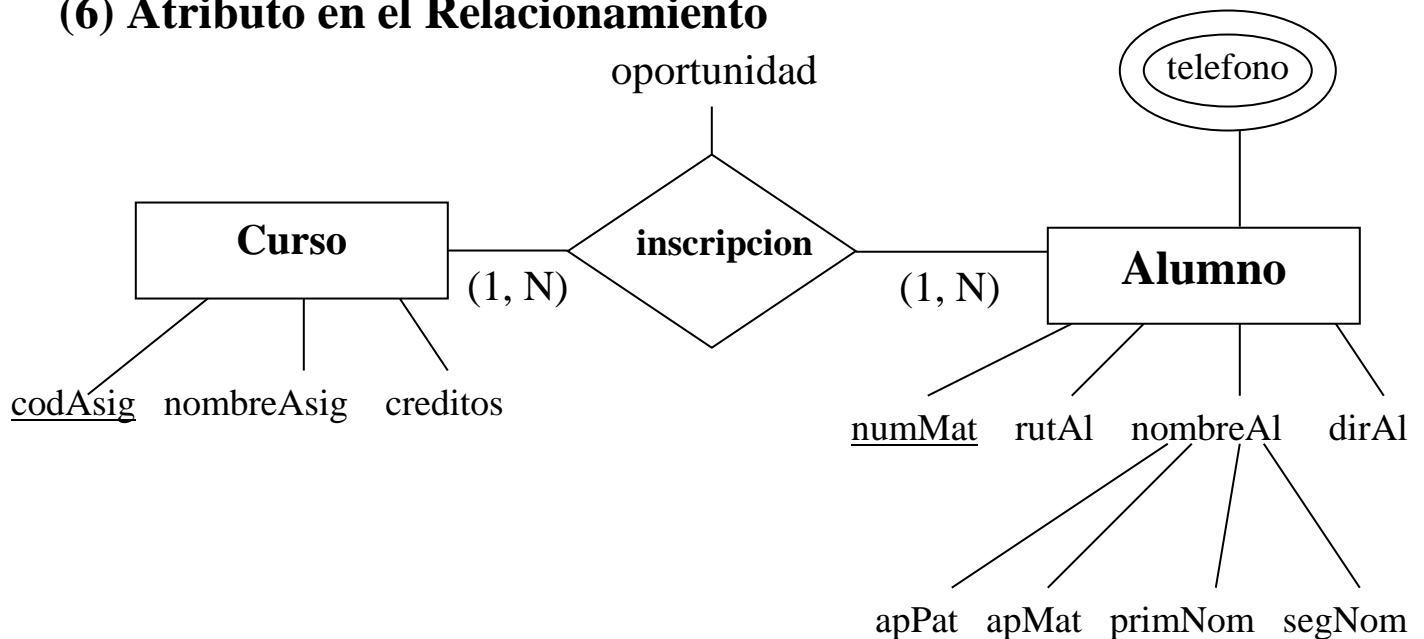
**Curso** (codAsig, nombreAsig, credits)

**Alumno** (numMat, rutAl, apPat, apMat, primNom, segNom, dirAl)

**AlumnoTelefono** (numMat, telefono)

**inscripción** (codAsig, numMat)

## (6) Atributo en el Relacionamiento



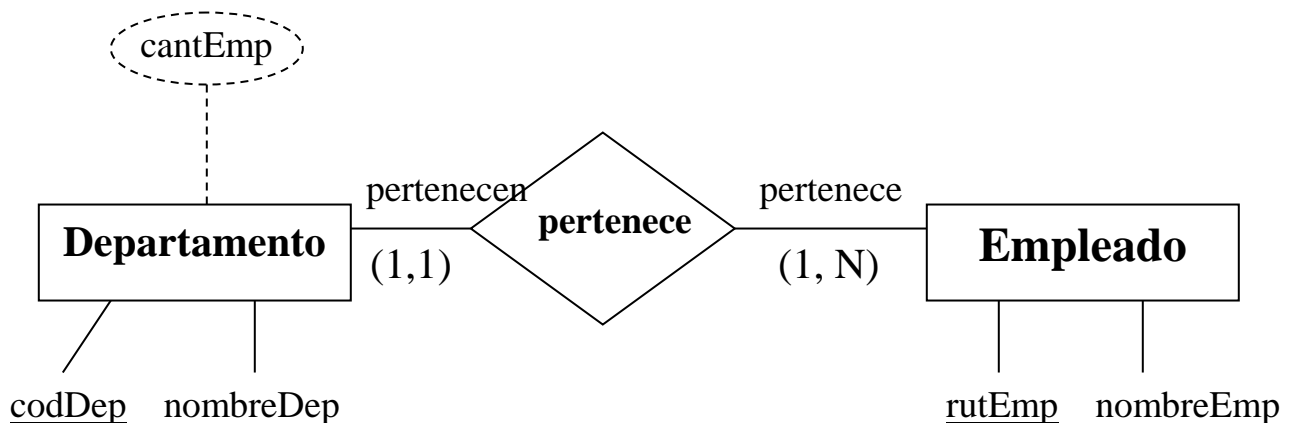
**Curso** (codAsig, nombreAsig, credits)

**Alumno** (numMat, rutAl, apPat, apMat, primNom, segNom, dirAl)

**AlumnoTelefono** (numMat, telefono)

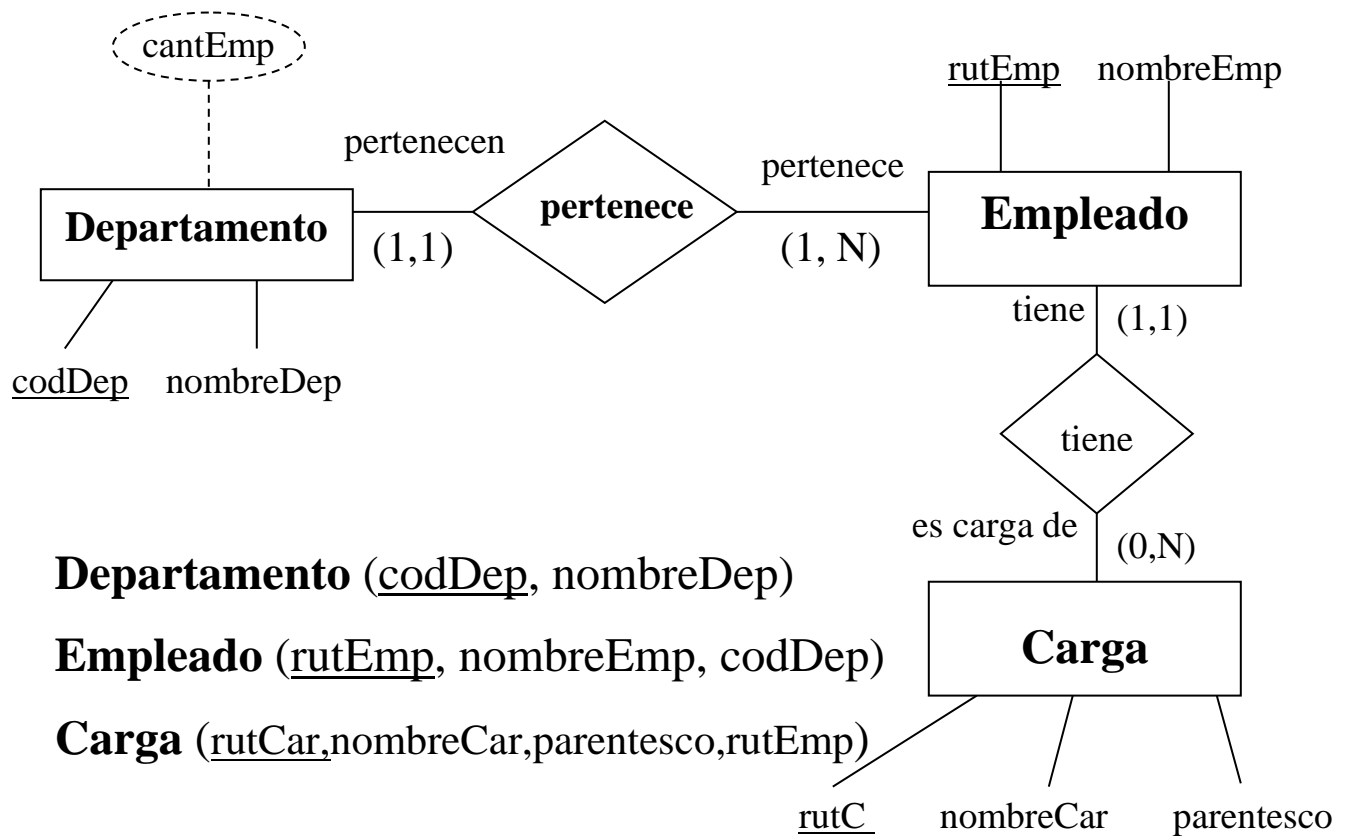
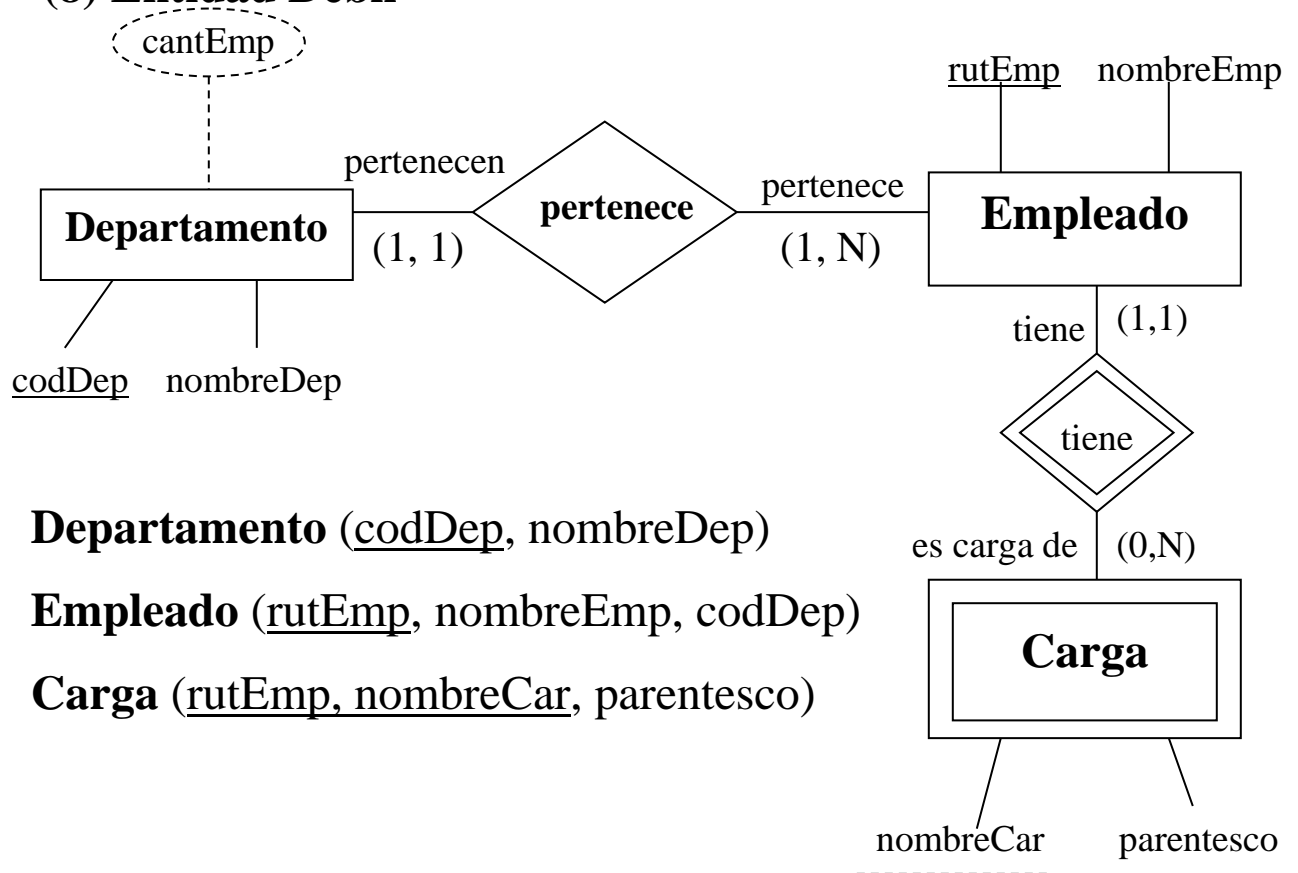
**inscripción** (codAsig, numMat, oportunidad)

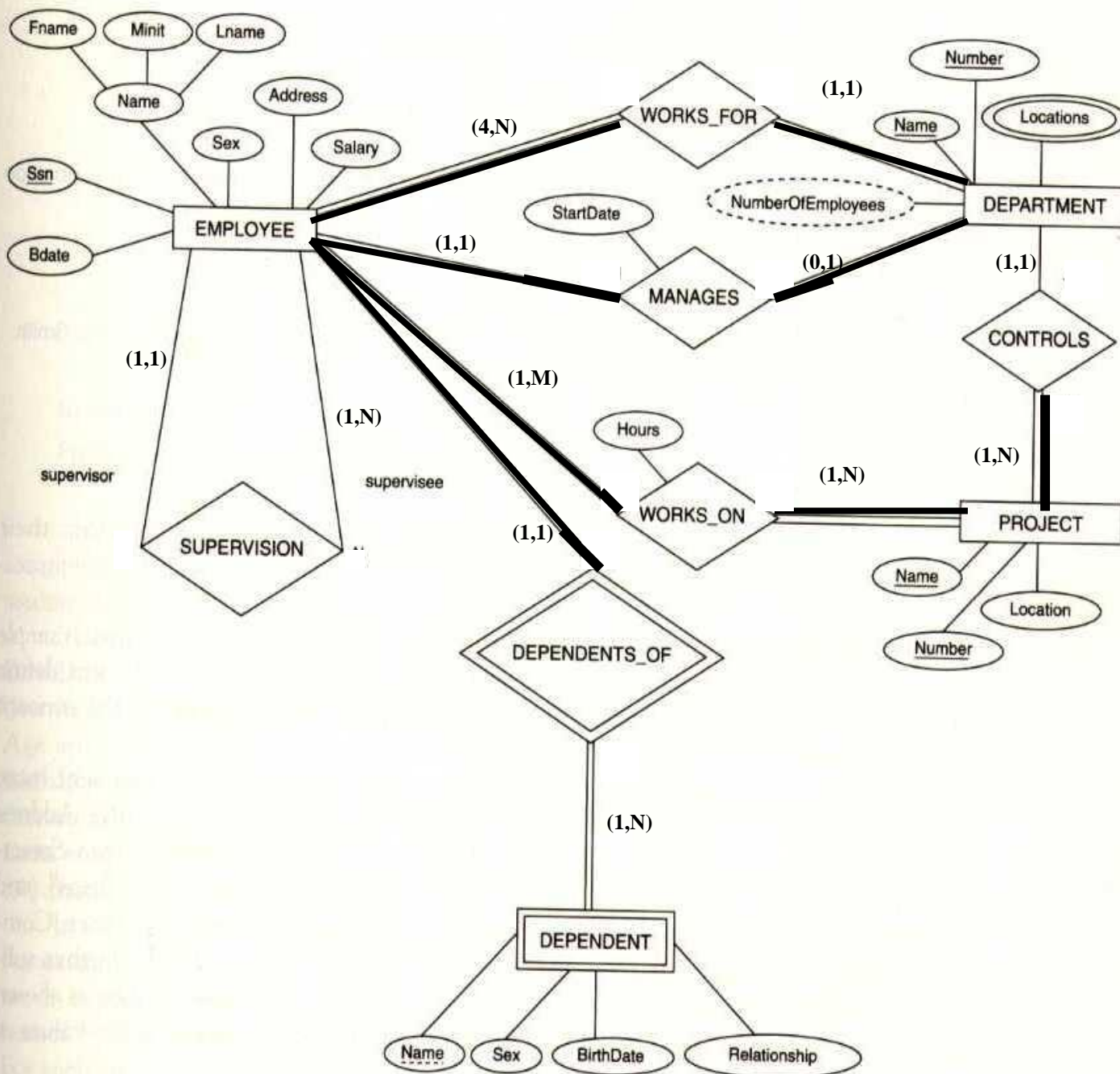
## (7) Atributo Derivado

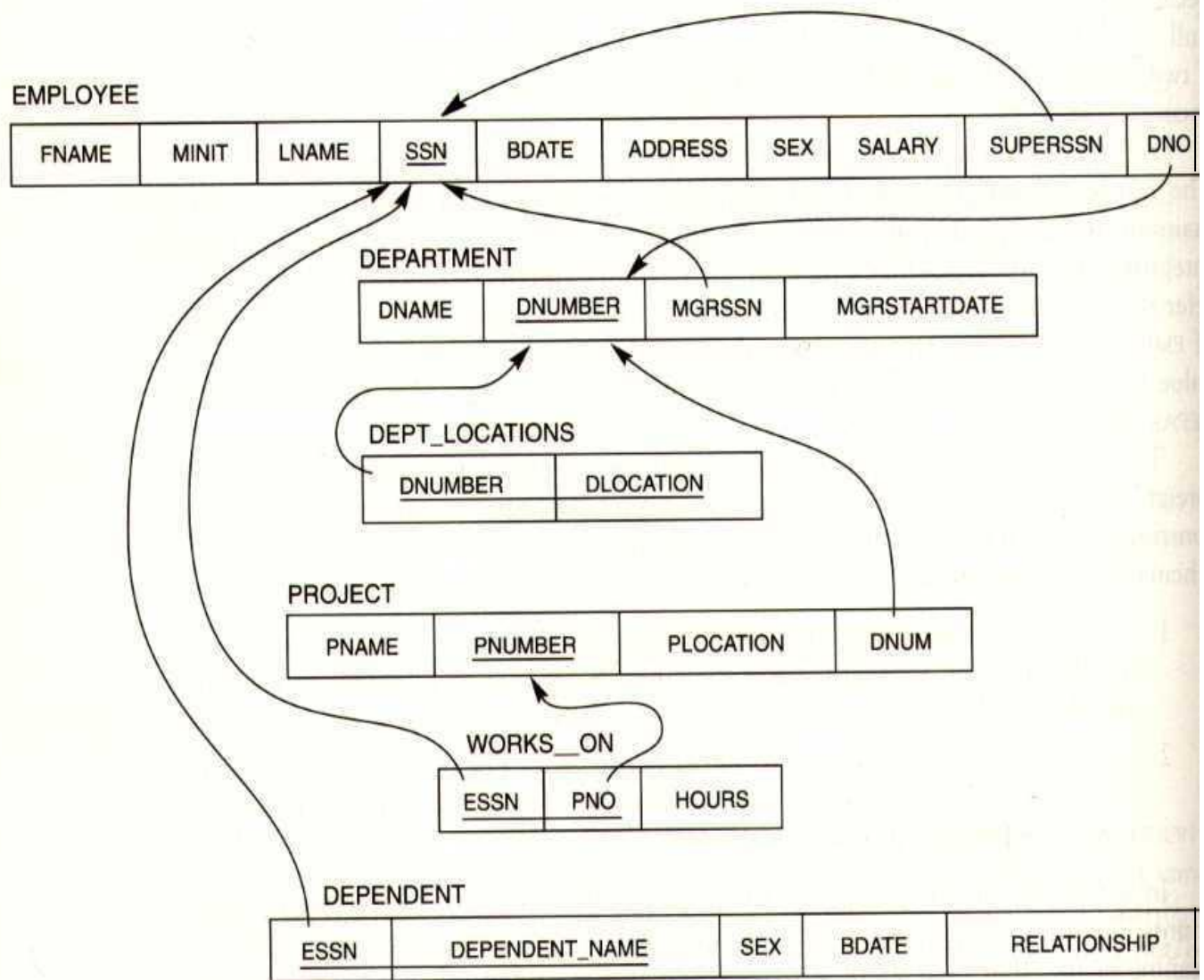


**Departamento** (codDep, nombreDep)

**Empleado** (rutEmp, nombreEmp, codDep)

**(8) Entidad Débil**

**Ejemplo:****Modelo E- R para la base de datos COMPANY**



**Esquema de la base de datos COMPANY**

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	09-JAN-55	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	08-DEC-45	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	19-JUL-58	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	20-JUN-31	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	15-SEP-52	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	31-JUL-62	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	29-MAR-59	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	10-NOV-27	450 Stone, Houston, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	22-MAY-78
	Administration	4	987654321	01-JAN-85
	Headquarters	1	888665555	19-JUN-71

WORKS_ON	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	05-APR-76	DAUGHTER
	333445555	Theodore	M	25-OCT-73	SON
	333445555	Joy	F	03-MAY-48	SPOUSE
	987654321	Abner	M	29-FEB-32	SPOUSE
	123456789	Michael	M	01-JAN-78	SON
	123456789	Alice	F	31-DEC-78	DAUGHTER
	123456789	Elizabeth	F	05-MAY-57	SPOUSE

**Instancia de la base de datos COMPANY**



## 8. Relacionamiento Superclase / Subclase y Especialización (Generalización)

Convertir cada especialización con  $m$  subclases  $\{S_1, S_2, \dots, S_m\}$  y superclase  $C$ , donde los atributos de  $C$  son  $\{k, a_1, a_2, \dots, a_n\}$  y  $k$  es la clave primaria en esquemas relacionales, utilizando una de las cuatro siguientes opciones:

8. a Crear una relación  $L$  para  $C$ , con los atributos  $\text{Attrs}(L) = \{k, a_1, a_2, \dots, a_n\}$  y  $\text{PK}(L) = k$ .  
 Crear una relación  $L_i$  para cada subclase  $S_i$ ,  $1 \leq i \leq m$ , con los atributos  $\text{Attrs}(L_i) = \{k\} \cup \{\text{atributos de } S_i\}$  y  $\text{PK}(L_i) = k$ .
8. b Crear una relación  $L_i$  para cada subclase  $S_i$ ,  $1 \leq i \leq m$ , con los atributos  $\text{Attrs}(L_i) = \{\text{atributos de } S_i\} \cup \{k, a_1, a_2, \dots, a_n\}$  y  $\text{PK}(L_i) = k$ .

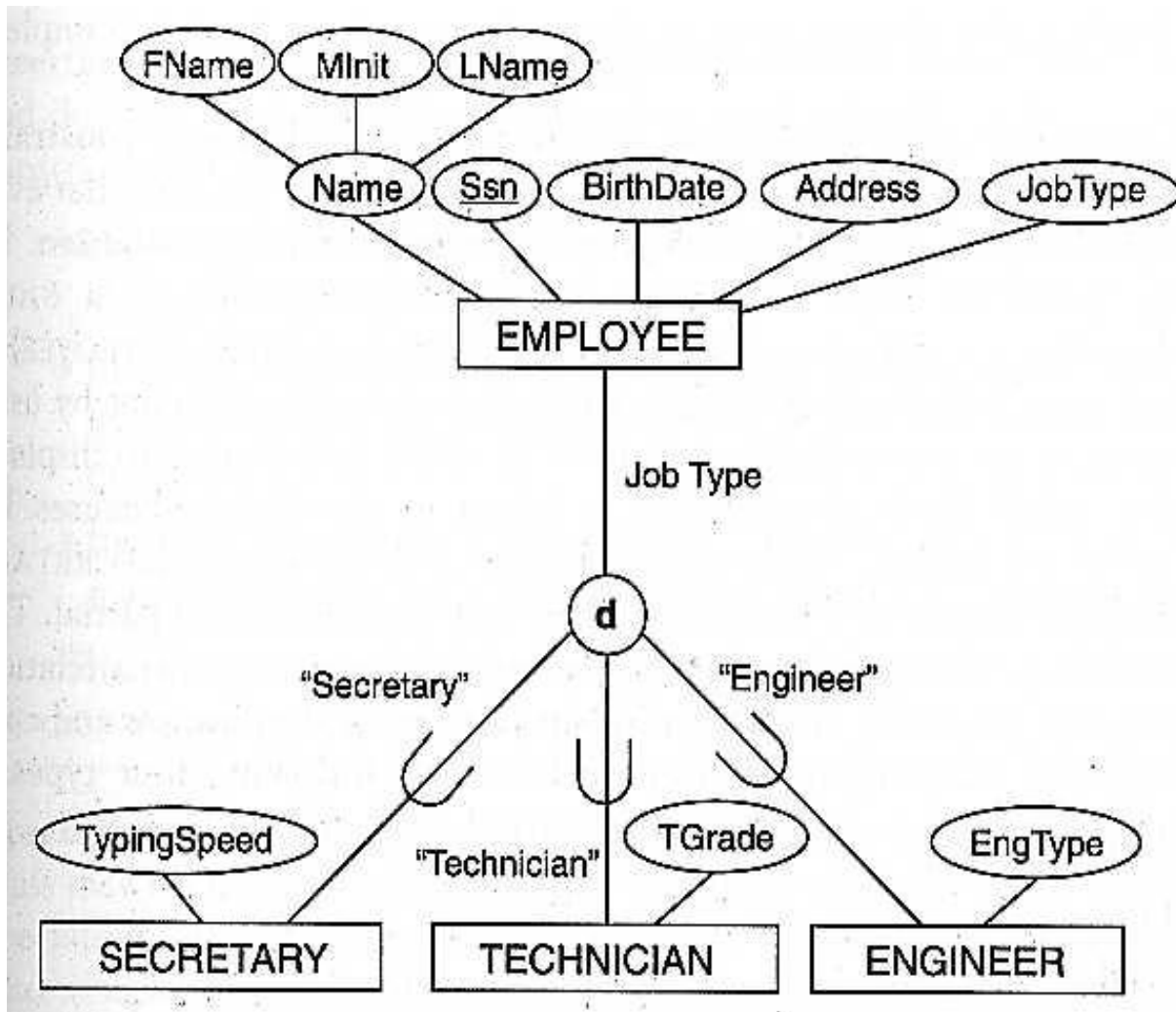
- 8. c** Crear una única relación  $L$  con los atributos  $\text{Attrs}(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos de } S_1\} \cup \{\text{atributos de } S_2\} \cup \dots \{\text{atributos de } S_m\} \cup \{t\}$  y  $\text{PK}(L) = k$ .

Esta opción es para una especialización cuyas subclases son disjuntas, y  $t$  es un atributo que indica a la subclase a la cual pertenece, si la hay. Esta opción puede generar una gran cantidad de valores null.

- 8. d** Crear una única relación  $L$ , con los atributos  $\text{Attrs}(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos de } S_1\} \cup \{\text{atributos de } S_2\} \cup \dots \{\text{atributos de } S_m\} \cup \{t_1, t_2, t_3, \dots, t_m\}$  y  $\text{PK}(L) = k$ .

Esta opción es para especialización cuyas subclases son solapadas, y cada  $t_i$ ,  $1 \leq i \leq m$ , es un atributo booleano que indica si una tupla pertenece a la subclase.

También puede generar una gran cantidad de valores null.

**Ejemplo:**

**Especialización definida por atributo, para el atributo JobType de la tabla EMPLOYEE**

## Transformación al modelo relacional de la especialización de EMPLOYEE

### EMPLOYEE

<u>SSN</u>	Fname	Minit	Lname	BirthDate	Address	JobType
------------	-------	-------	-------	-----------	---------	---------

### SECRETARY

<u>SSN</u>	TypingSpeed
------------	-------------

### TECHNICIAN

<u>SSN</u>	Tgrade
------------	--------

### ENGINEER

<u>SSN</u>	EngType
------------	---------

**Transformación utilizando opción 8.a**

**SECRETARY**

<u>SSN</u>	TypingSpeed	Fname	Minit	LName	Birth Date	Address

**TECHNICIAN**

<u>SSN</u>	TGrade	FName	Minit	LName	BirthDate	Address

**ENGINEER**

<u>SSN</u>	EngType	FName	Minit	LName	BirthDate	Address

**Transformación utilizando opción 8.b****EMPLOYEE**

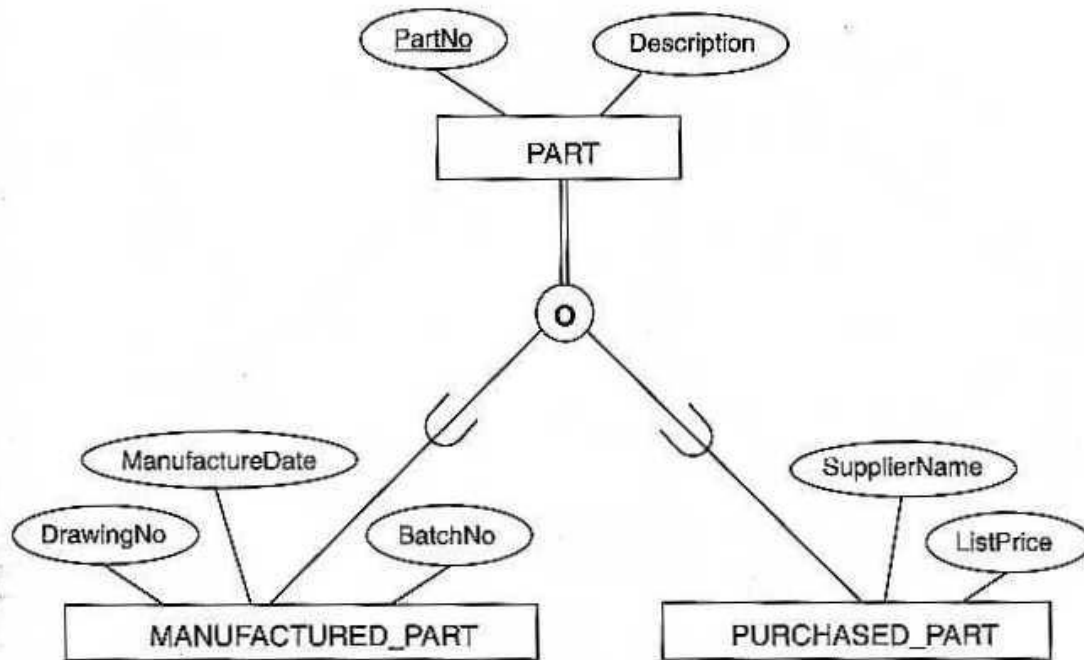
<u>SSN</u>	Fname	Minit	LName	Birth Date	Address	JobType
Typing Speed	EngType	TGrade				

**Transformación utilizando opción 8.c**

**EMPLOYEE**

<u>SSN</u>	Fname	Minit	LName	BirthDate	Address
Type	TGrade	Eng	SType	SEngType	STGrade
Speed		Type	Speed		

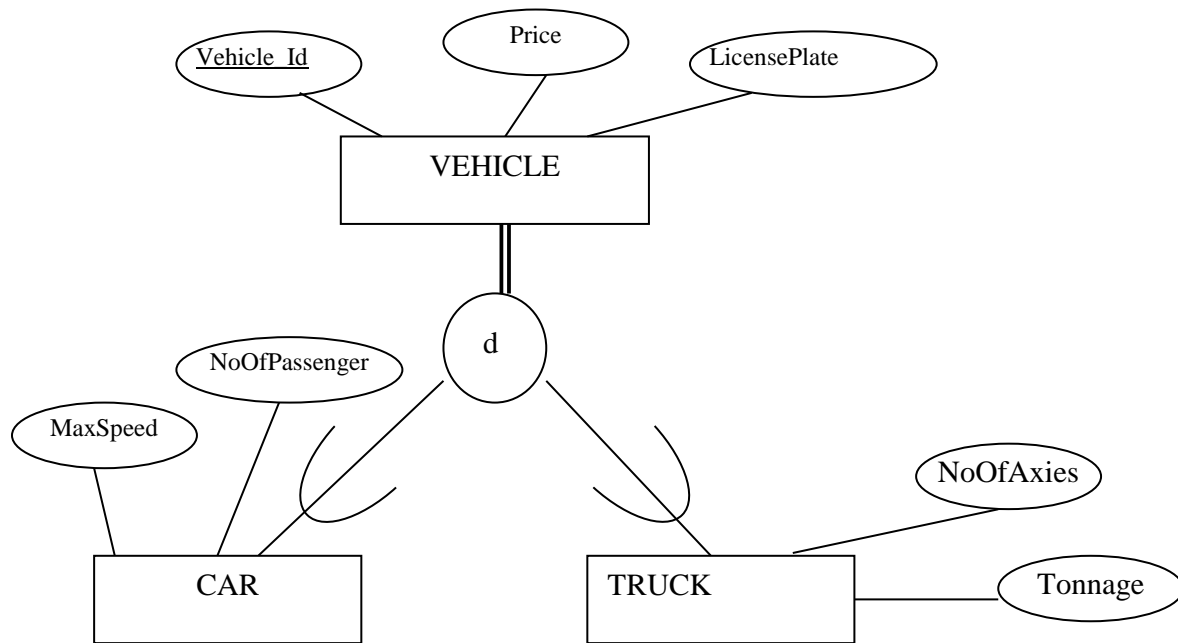
**Transformación utilizando opción 8.d y suponiendo  
subclases solapadas**

**Ejemplo:****Especialización con subclases no disjuntas (solapadas)**

PART

<u>PartNo</u>	Description	MFlag	DrawingNo	Manufacturate Date
BatchNo	PFlag	Supplier Name	List Price	

**Transformación al modelo relacional de una especialización con subclases solapadas. Se agregan dos campos de tipo booleano (Mflag y Pflag)**

**Ejemplo:**

**Especialización de VEHICLE como una especialización total disjunta**

**CAR**

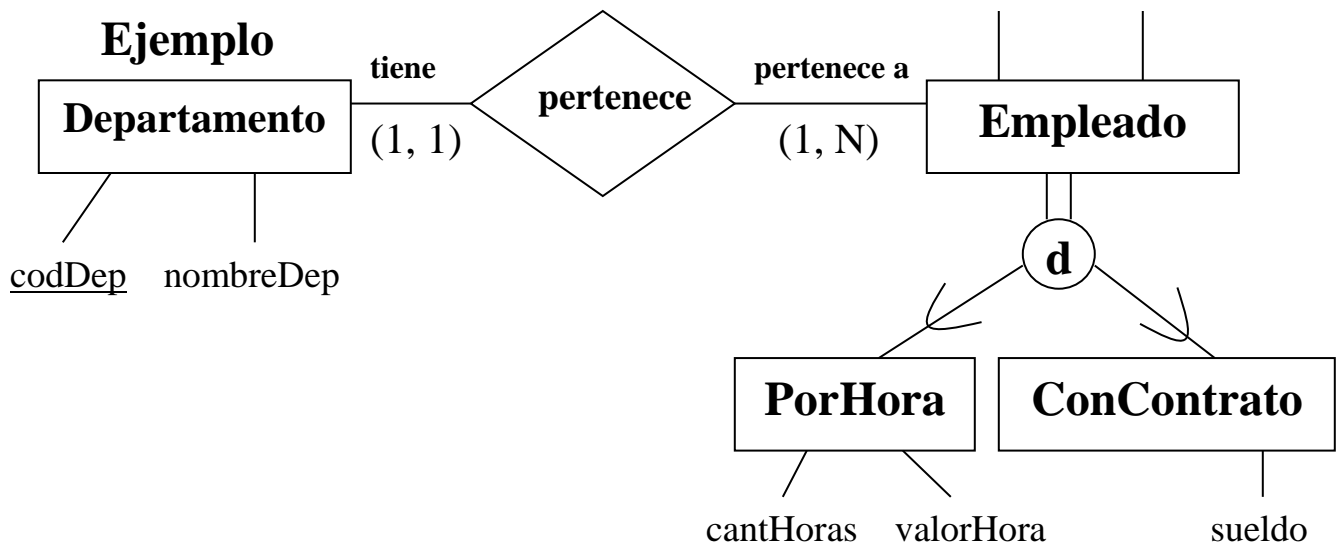
<u>VehicleId</u>	LicensePlateNo	Price	MaxSpeed	NoOfPassengers
------------------	----------------	-------	----------	----------------

**TRUCK**

<u>VehicleId</u>	LicensePlateNo	Price	NoOfAxies	Tonnage
------------------	----------------	-------	-----------	---------

**Modelo relacional para la especialización de VEHICLE**





**Departamento** (codDep, nombreDep)

**Empleado** (rutEmp, nombreEmp, codDep)

**PorHora** (rutEmp, cantHoras, valorHora)

**ConContrato** (rutEmp, sueldo)

ó

**Departamento** (codDep, nombreDep)

**PorHora** (rutEmp, nombreEmp, codDep, cantHoras, valorHora)

**ConContrato** (rutEmp, nombreEmp, codDep, sueldo)

ó

**Departamento** (codDep, nombreDep)

**Empleado** (rutEmp, nombreEmp, codDep, tipoEmp,  
cantHoras, valorHora, sueldo)

ó

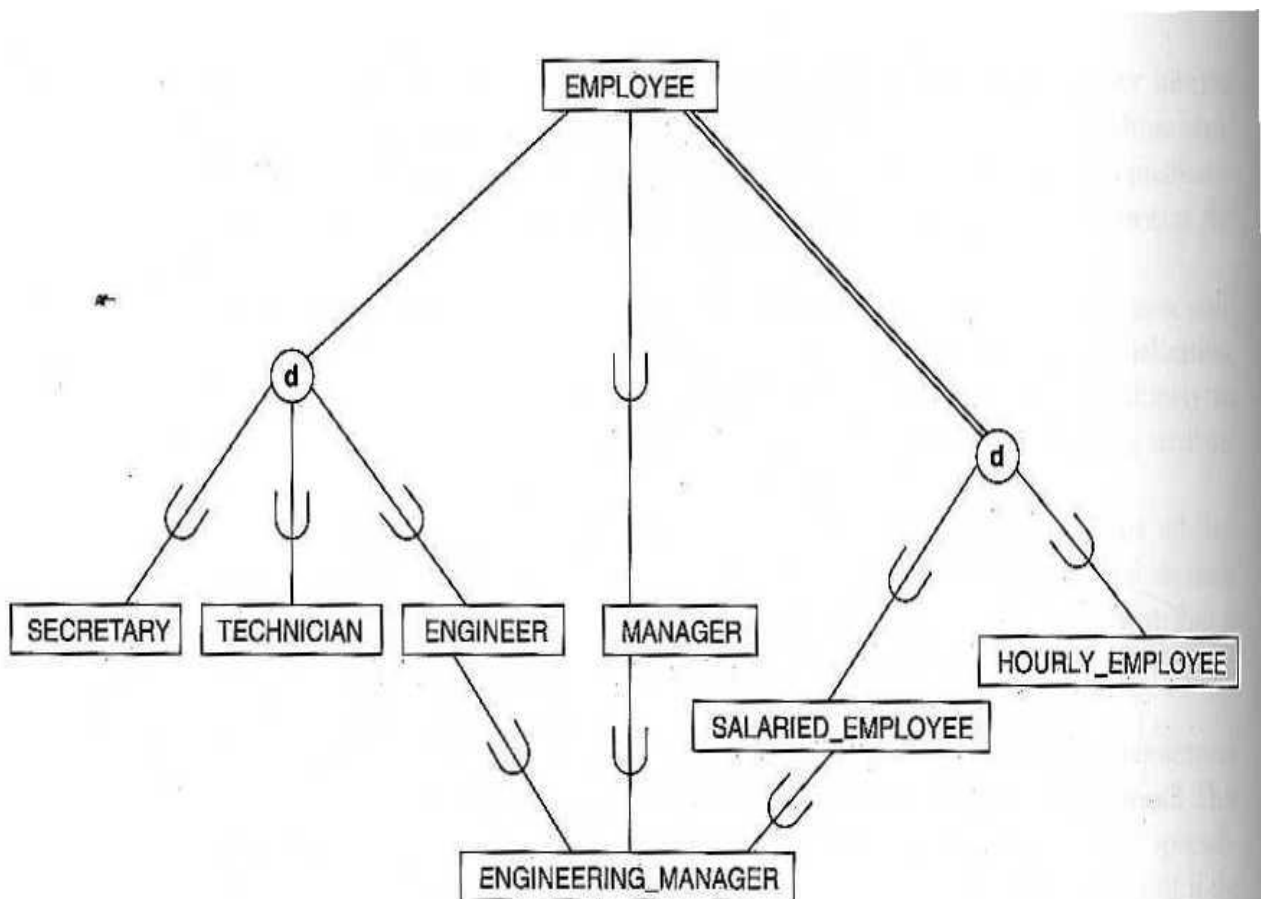
**Departamento** (codDep, nombreDep)

**Empleado** (rutEmp, nombreEmp, codDep, flagPorHora,  
flagConContrato, cantHoras, valorHora, sueldo)

## 9. Subclases compartidas

Se pueden aplicar cualquiera de las opciones discutidas en el paso 8, aunque usualmente la opción 8a es utilizada.

### Ejemplo 1:



**Especialización con la subclase compartida  
ENGINEERING\_MANAGER**

**SALARIED\_EMPLOYEE**

<u>SSN</u>	Fname	Minit	LName	BirthDate	Address	Salary
------------	-------	-------	-------	-----------	---------	--------

**HOURLY\_EMPLOYEE**

<u>SSN</u>	Fname	Minit	LName	BirthDate	Address	ValorHora
------------	-------	-------	-------	-----------	---------	-----------

**MANAGER**

<u>SSN</u>	Fname	Minit	LName	BirthDate	Address
------------	-------	-------	-------	-----------	---------

**ENGINEERING\_MANAGER**

<u>SSN</u>
------------

**SECRETARY**

<u>SSN</u>	TypingSpeed
------------	-------------

**TECHNICIAN**

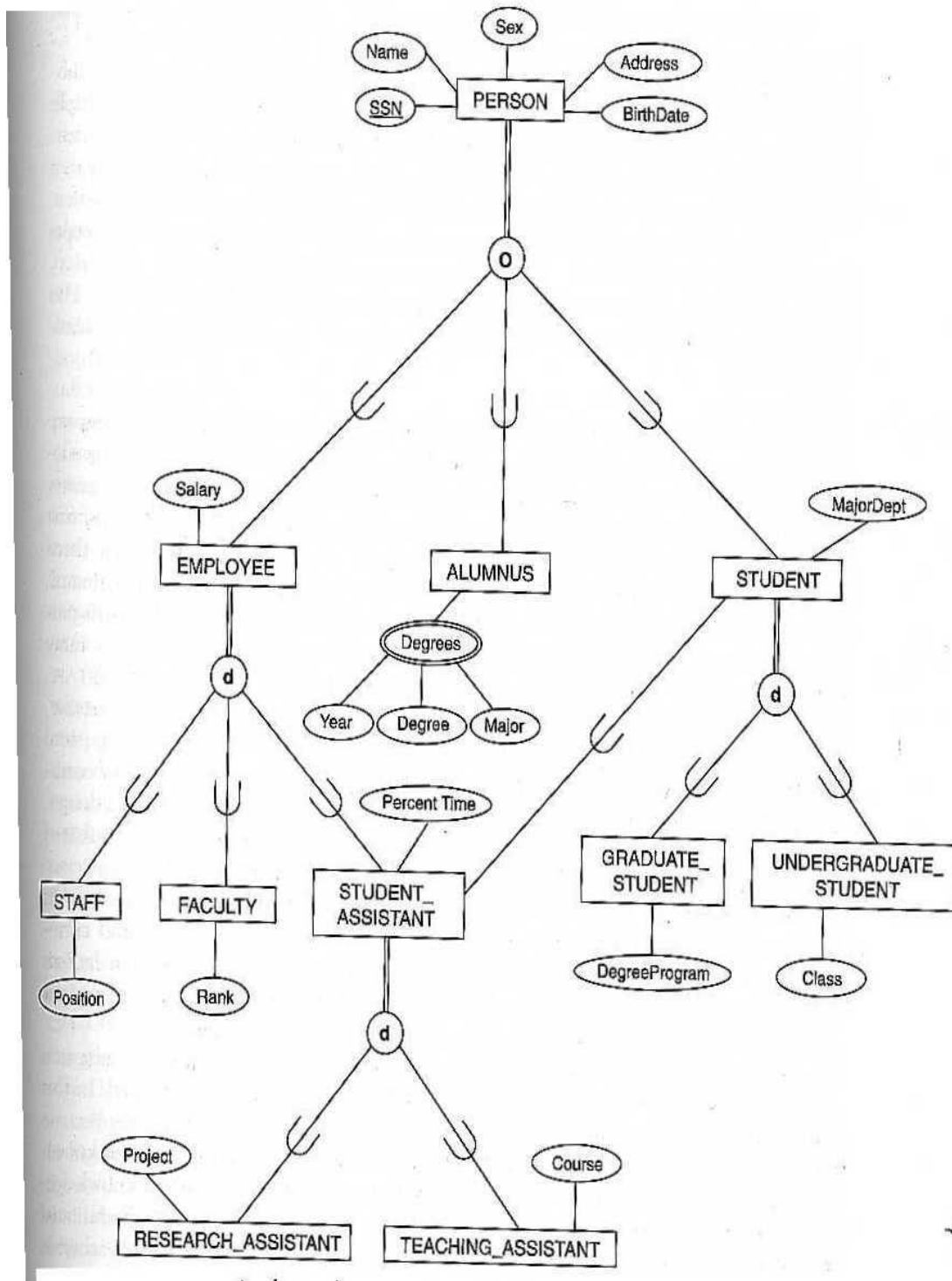
<u>SSN</u>	EngType
------------	---------

**ENGINEER**

<u>SSN</u>	Tgrade
------------	--------

**Transformación al modelo relacional de la subclase  
compartida ENGINEERING\_MANAGER**

## Ejemplo 2:



**Especialización para la base de datos COMPANY**

**PERSON**

<u>SSN</u>	Name	BirthDate	Sex	Address
------------	------	-----------	-----	---------

**EMPLOYEE**

<u>SSN</u>	Salary	EmployeeType	Position	Rank
PercentTime	RAFlag	TAFflag	Project	Course

**ALUMNUS**

<u>SSN</u>
------------

**ALUMNUS\_DEGREES**

<u>SSN</u>	<u>Year</u>	<u>Degree</u>	<u>Major</u>
------------	-------------	---------------	--------------

**STUDENT**

<u>SSN</u>	MajorDept	GradFlag	UndergradFlag
Degree Program	Class	StudAssist Flag	

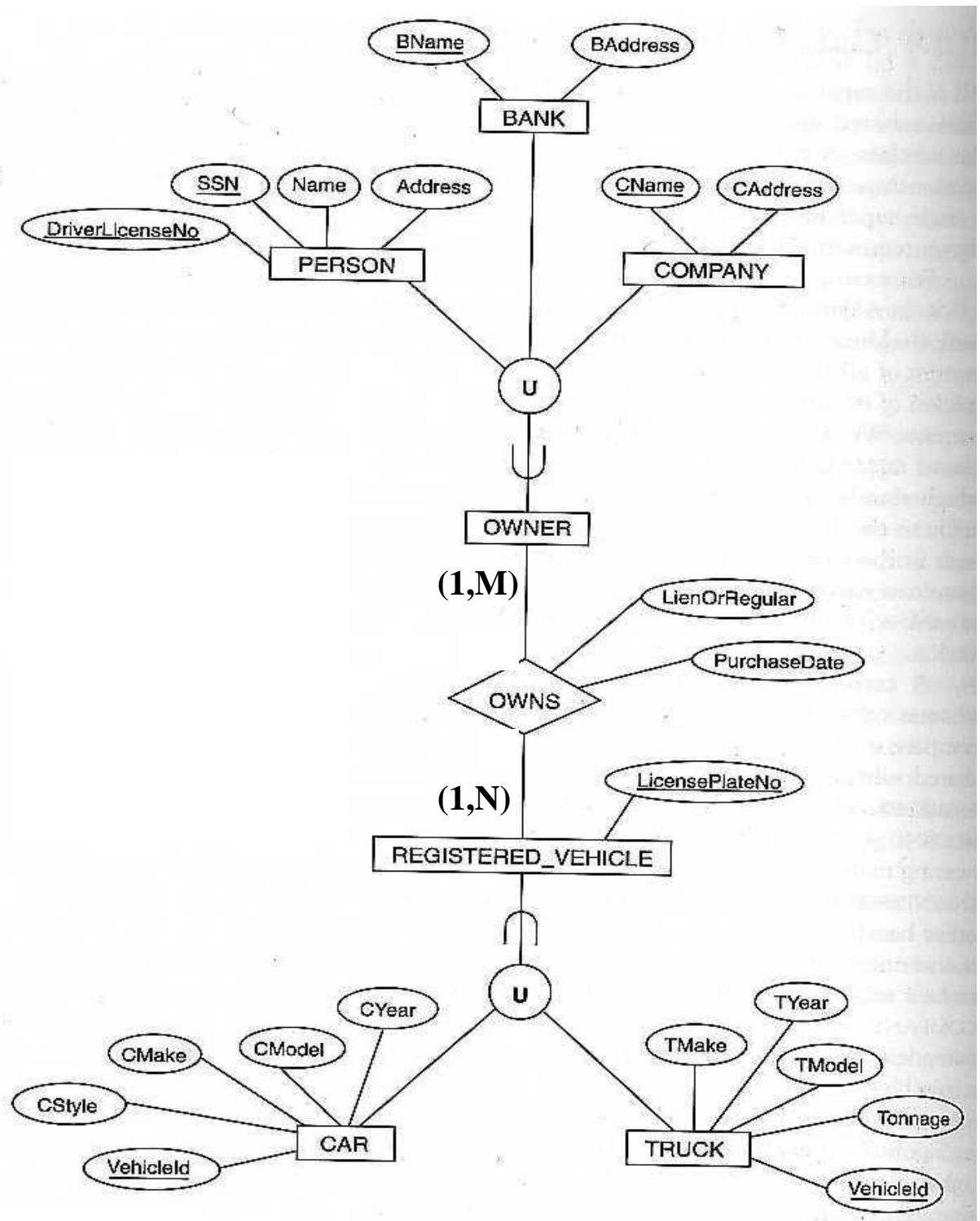
**Modelo relacional para la base de datos UNIVERSITY**

## 10. Categorías

En el caso de una categoría en que las superclases tienen diferentes claves, se acostumbra a especificar un nuevo atributo clave, denominado **surrogate key** (clave substituta), cuando se crea una relación que corresponde a la categoría.

Para una categoría en que las superclases tienen la misma clave, no se necesita una surrogate key.

## Ejemplo:



**Dos Categorías: OWNER y REGISTERED\_VEHICLE**

**PERSON**

<u>SSN</u>	DriverLicenseNo	Name	Address	OwnerId
------------	-----------------	------	---------	---------

**BANK**

<u>BName</u>	BAddress	OwnerId
--------------	----------	---------

**COMPANY**

<u>Cname</u>	CAddress	OwnerId
--------------	----------	---------

**OWNER**

<u>OwnerId</u>
----------------

**REGISTERED\_VEHICLE**

<u>VehicleId</u>	LicensePlateNumber
------------------	--------------------

**CAR**

<u>VehicledId</u>	CStyle	CMake	CModel	CYear
-------------------	--------	-------	--------	-------

**TRUCK**

<u>VehicledId</u>	TMake	TModel	Tonnage	TYear
-------------------	-------	--------	---------	-------

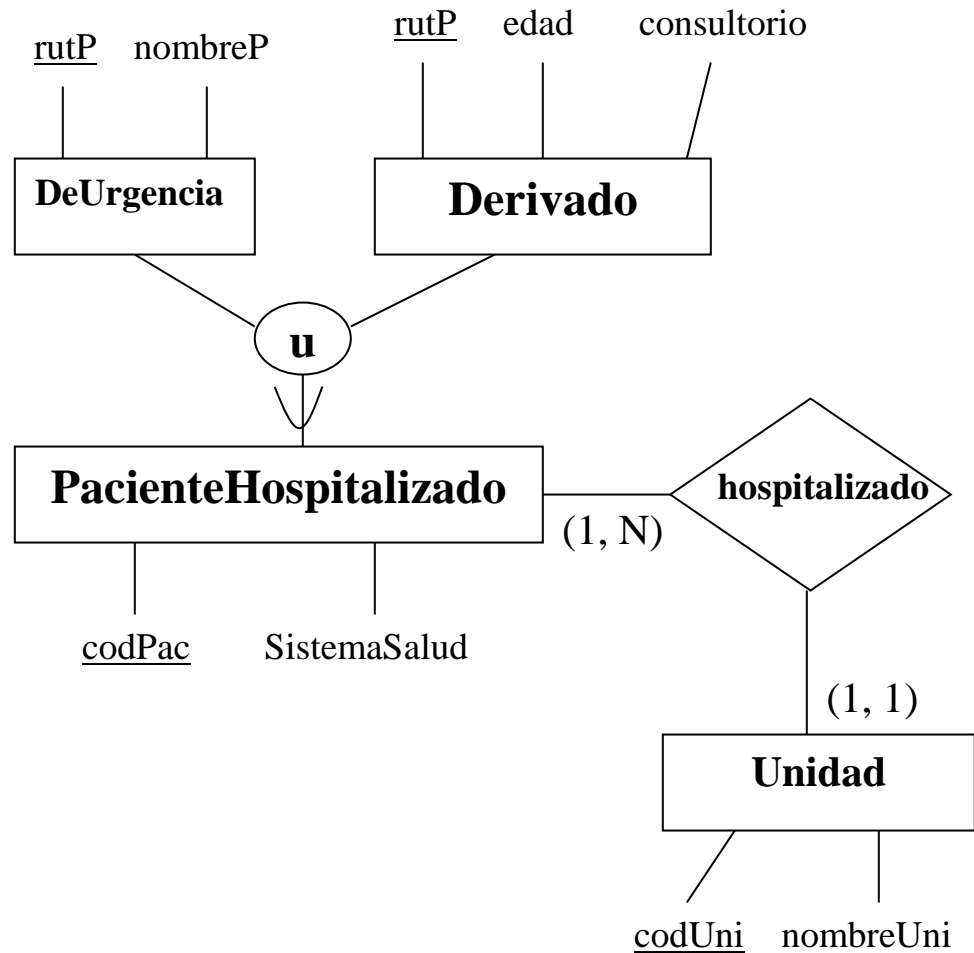
**OWNS**

<u>OwnerId</u>	<u>VehicledId</u>	PurchaseDate	LienOrRegular
----------------	-------------------	--------------	---------------

**Modelo relacional para las categorías OWNER y  
REGISTERED\_VEHICLE**



## Ejemplo



Es parcial, por que no todos los derivados y de urgencia quedan hospitalizados

**DeUrgencia** (rutP, nombreP)

**Derivado** (rutP, edad, consultorio)

**PacienteHospitalizado** (codPac, sistemaSalud, rutP, codUni)

**Unidad** (codUni, nombreUni)

*En vez que codPac sea la clave primaria, podría haber sido rutP*

## 11. Agregación

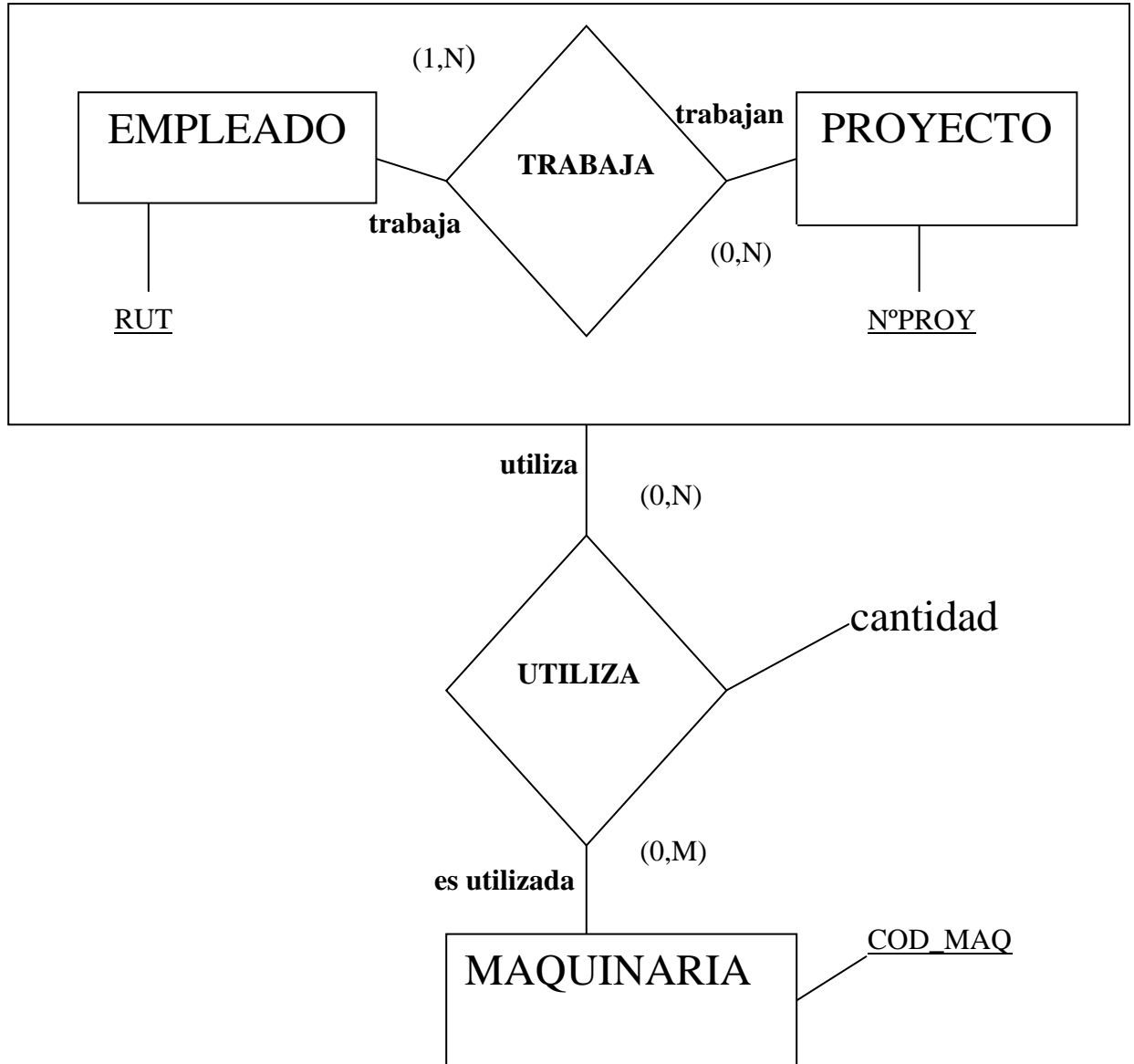
En el caso de una agregación, la transformación al modelo relacional implica crear las tablas de todas las entidades y relaciones, según corresponda.

Para el relacionamiento que involucra a la agregación, si es muchos a muchos, se crea una relación que contiene la llave primaria de cada una de las entidades simples (entidades no involucradas en la agregación, pero si asociadas con el relacionamiento en cuestión), la llave primaria del relacionamiento implicado en la agregación y los correspondientes atributos descriptivos del relacionamiento, si existieran.

La llave primaria de esta relación es la concatenación de las llaves primarias incluidas.

**Ejemplo:**

## TRABAJO



**D-E-R mostrando la agregación TRABAJO**

**MAQUINARIA**COD\_MAQ**EMPLEADO**RUT**PROYECTO**N°PROY**TRABAJA**

<u>RUT</u>	<u>N°PROY</u>

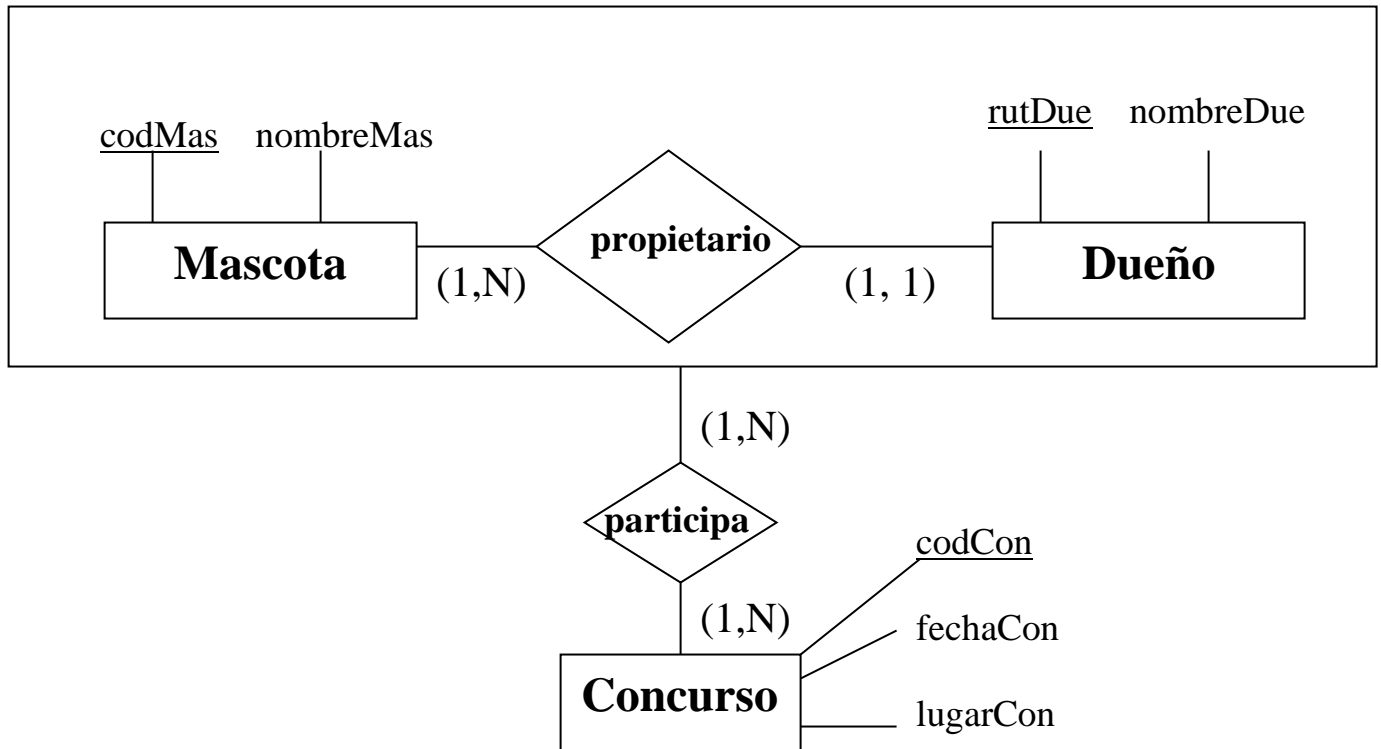
**UTILIZA**

<u>COD_MAQ</u>	<u>RUT</u>	<u>N°PROY</u>	CANTIDAD

**Modelo Relacional de la agregación TRABAJO**

## Ejemplo

### Participante



**Mascota** (codMas, nombreMas, rutDue)

**Dueño** (rutDue, nombreDue)

**Concurso** (codCon, fechaCon, lugarCon)

**participa** (codCon, codMas)

## **3.2 Lenguajes relacionales formales**

- 1. Álgebra relacional: ¡Procedural!.**
- 2. Cálculo relacional: ¡No procedural!, ¡Descriptivo!.**

**Las operaciones sobre los datos corresponden a operaciones sobre las relaciones.**

**Ambos lenguajes son equivalentes en poder expresivo.**

### **3.2.1 Álgebra relacional**

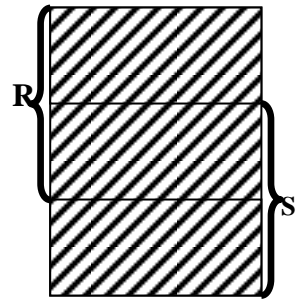
- Es un lenguaje procedural, porque cuando se escribe una expresión en álgebra relacional, se proporciona una secuencia de operaciones que genera la respuesta a la consulta.
- Dispone de un conjunto de operadores de alto nivel, que operan sobre relaciones.
- Cada uno de estos operadores, toma una o dos relaciones como entrada, y produce una nueva relación como salida (**propiedad de clausura**).

### 3.2.1.1 Operaciones fundamentales

- |                        |             |
|------------------------|-------------|
| a) UNION               | $R \cup S$  |
| b) DIFERENCIA          | $R - S$     |
| c) PRODUCTO CARTESIANO | $R * S$     |
| d) PROYECCION          | $\pi(R)$    |
| e) SELECCIÓN           | $\sigma(R)$ |

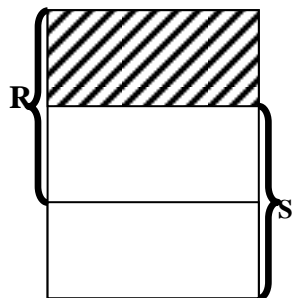
#### a) Unión de R y S

- $(R \cup S)$
- $t \in R \vee t \in S$
- Aplicable si R y S tienen la misma aridad

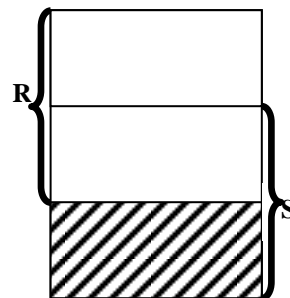


#### b) Diferencia de R y S

- $(R - S)$
- $t \in R \wedge t \notin S$



diferencia  
R MENOS S



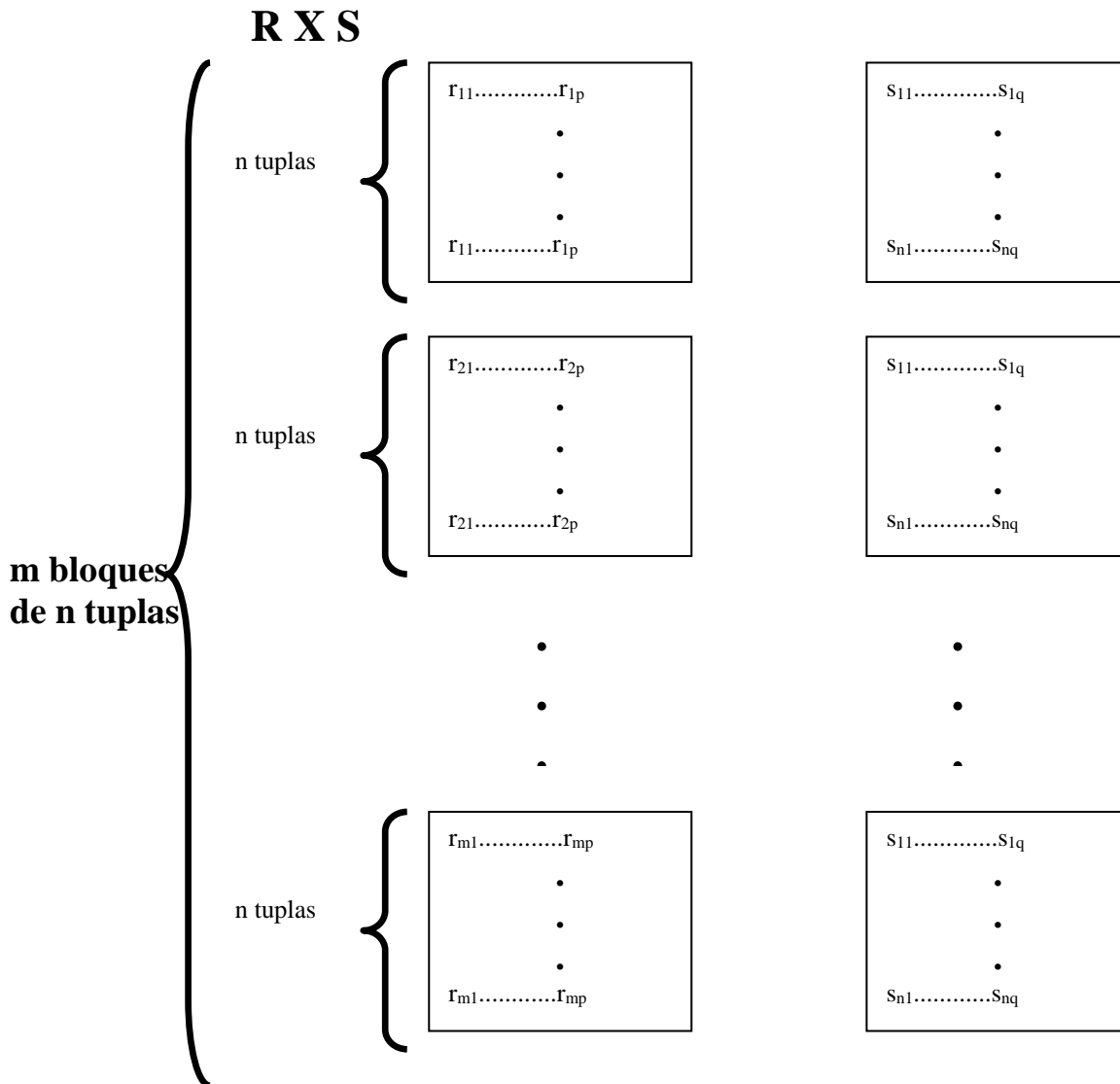
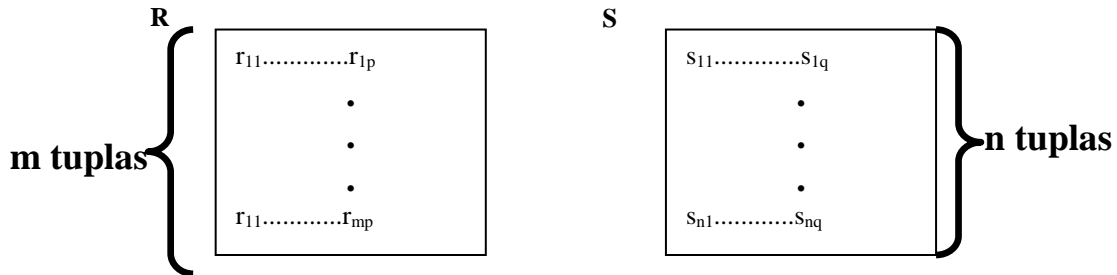
diferencia  
S MENOS R

- Aplicable si R y S tienen la misma aridad.

c) **Producto cartesiano de R y S**

- **( R x S )**
- Aridad es  $(K1 + K2)$ , si aridad de R es  $K1$  y aridad de S es  $K2$ .

d)

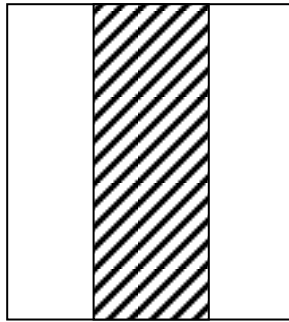


**Producto Cartesiano R X S**



## Proyección $\pi$

- $\Pi_{i_1 i_2 \dots i_m} (R)$
- Conjunto de m-tuplas  $a_1, a_2, \dots, a_m$  tales que  $\exists b_1, b_2, \dots, b_k \in R$  para el cual  $a_j = b_{i_j}$



Proyección

### Ejemplo 1:

$R (A, B, C, D)$

$\Pi_{3,1}(R)$  contiene las tuplas con dos componentes, correspondientes a los valores de los atributos C y A

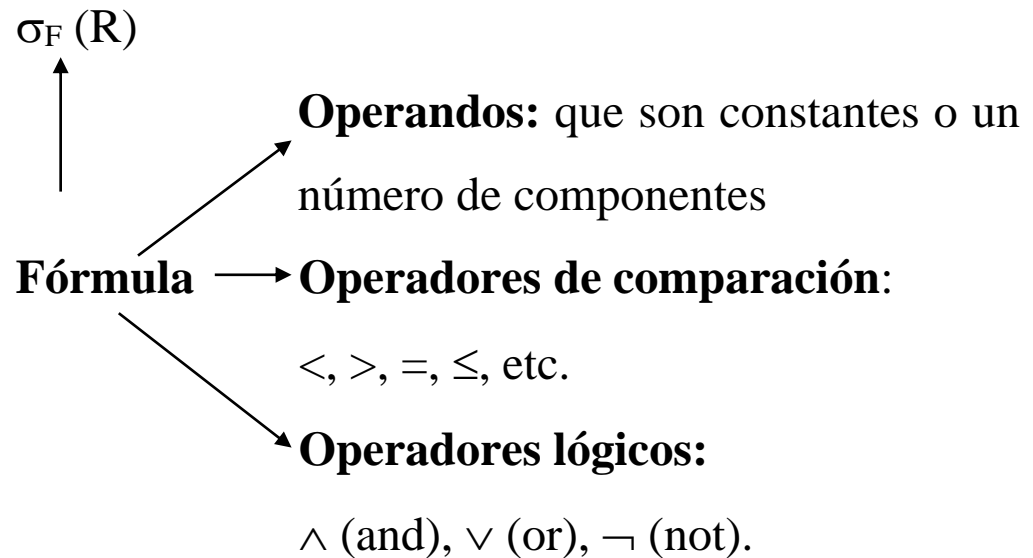
$$\Pi_{3,1}(R) \equiv \Pi_{C,A} (R)$$

**Ejemplo 2:**

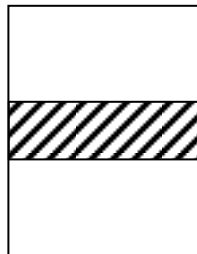
<b>R</b>			<b>S</b>	
<b>Nombre</b>	<b>Edad</b>	<b>Sueldo</b>	<b>Nombre</b>	<b>Depto</b>
Pérez	25	120.000	Pérez	Ventas
Fuentes	36	200.000	López	Finanzas
López	45	180.000	Fuentes	Producción
Soto	30	250.000	Soto	Ventas

$\pi_{3,1}(\mathbf{R}) \longrightarrow$	<b>Sueldo</b>	<b>Nombre</b>
	120.000	Pérez
	200.000	Fuentes
	180.000	López
	250.000	Soto

$\pi_{\text{Depto}}(\mathbf{S}) \longrightarrow$	<b>Depto</b>
	Ventas
	Finanzas
	Producción

e) **Selección  $\sigma$** 

Valor de atributo  $i \Rightarrow \$i$



Selección

**Ejemplo 3:**  $\sigma_{\$1 = \text{'manzana'}}(R) \rightarrow (\text{manzana}, \text{roja})$

$\sigma_{\$1 > \$2}(S) \rightarrow$  Tuplas de  $S$  en que el valor de la primera componente es mayor que el de la segunda componente

$\sigma_{B = 200 \wedge c > 1000}(R)$

**Ejemplo 4:** Dadas las relaciones R y S:

<b>R</b>			<b>S</b>	
<u>Nombre</u>	<u>Edad</u>	<u>Sueldo</u>	<u>Nombre</u>	<u>Depto</u>
Pérez	25	120.000	Pérez	Ventas
Fuentes	36	200.000	López	Finanzas
López	45	180.000	Fuentes	Producción

**i) R x S**

<b>R.Nombre</b>	<b>Edad</b>	<b>Sueldo</b>	<b>S.Nombre</b>	<b>Depto</b>
Pérez	25	120.000	Pérez	Ventas
Pérez	25	120.000	López	Finanzas
Pérez	25	120.000	Fuentes	Producción
Fuentes	36	200.000	Pérez	Ventas
Fuentes	36	200.000	López	Finanzas
Fuentes	36	200.000	Fuentes	Producción
López	45	180.000	Pérez	Ventas
López	45	180.000	López	Finanzas
López	45	180.000	Fuentes	Producción

ii. Obtener la información asociada a Pérez.

$$\sigma_{\text{Nombre} = \text{'Pérez'}}(\mathbf{R})$$

Nombre	Edad	Sueldo
Pérez	25	120.000

iii. Obtener la información de todos los empleados cuyas edades sobrepasan los 30 años.

$$\sigma_{\text{Edad} > 30}(\mathbf{R}) \longrightarrow$$

Nombre	Edad	Sueldo
Fuentes	36	200.000
López	45	180.000

iv. Obtener la información de todos los empleados cuyas edades sobrepasan los 30 años y tienen un salario menor que \$200.000.

$$\sigma_{\text{Edad} > 30 \wedge \text{Sueldo} < 200.000}(\mathbf{R})$$

Nombre	Edad	Sueldo
López	45	180.000

### 3.2.1.2 Operaciones adicionales

a) INTERSECCION  $\mathbf{R \cap S}$

b) CUOCIENTE  $\mathbf{R \div S}$

c) JOIN:

i.  $\theta$  JOIN  $\mathbf{R \bowtie_{i \theta j} S}$

ii. EQUIJOIN  $\mathbf{R \bowtie_{i=j} S}$

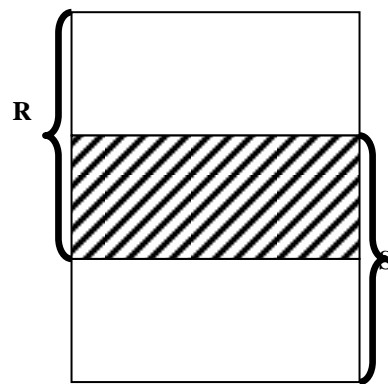
iii. JOIN NATURAL  $\mathbf{R \bowtie S}$

iv. SEMIJOIN  $\mathbf{R \ltimes S}$

v. OUTER JOIN  $\mathbf{R \Join S}$

a) **Intersección de R y S**

- $(R \cap S)$
- $t \in R \wedge t \in S$
- Es el conjunto de tuplas en R y en S
- $R \cap S \equiv R - (R - S)$



**intersección  
R INTERSECCION S**

b) **Cuociente entre R y S**

- $R \div S$
- Sean R y S relaciones de aridad r y s respectivamente, donde  $r > s$  y  $S \neq \emptyset$
- $R \div S$  es el conjunto de las  $(r - s)$  tuplas t, tal que **para todas** las s tuplas u en S, la tupla tu está en R.

**Ejemplo 5:**

Relación R				Relación S		Relación $R \div S$		
a	b	c	d	c	d	a	b	(x x c d
a	b	e	f	e	f	e	d	x x e f en R)
b	c	e	f					
e	d	c	d					
e	d	e	f					
a	b	d	e					

**Ejemplo 6:**

R:	<u>Nombre</u>	<u>Curso</u>	S:	<u>Curso</u>
	Juan	B. Datos		B. Datos
	Alvaro	I. Artif.		Com. Datos
	Mariana	Com. Datos		I. Artif.
	Mariana	B. Datos		
	Alvaro	Com. Datos		
	Mariana	I. Artif.		

¿Qué personas han tomado **todos** los cursos?

R / S  
Mariana



- El cuociente se puede expresar en términos de los operadores básicos:

$$R \div S = \pi_{1,2,\dots,r-s}(R) - \pi_{1,2,\dots,r-s}((\pi_{1,2,\dots,r-s}(R) \times S) - R)$$

**Ejemplo 7:** Usando las relaciones del ejemplo 5

$$\pi_{1,2}(R) \longrightarrow \begin{array}{c|c} a & b \\ b & c \\ e & d \end{array}$$

$$\pi_{1,2}(R) \times S \longrightarrow \begin{array}{c|c|c|c} a & b & c & d \\ a & b & e & f \\ b & c & c & d \\ b & c & e & f \\ e & d & c & d \\ e & d & e & f \end{array}$$

$$(\pi_{1,2}(R) \times S) - R \longrightarrow \begin{array}{c|c|c|c} b & c & c & d \end{array}$$

$$\pi_{1,2}((\pi_{1,2}(R) \times S) - R) \longrightarrow \begin{array}{c|c} b & c \end{array}$$

$$\pi_{1,2}(R) - \pi_{1,2}((\pi_{1,2}(R) \times S) - R) \longrightarrow \begin{array}{c|c} a & b \\ e & d \end{array}$$

### c) Join

i.  **$\theta$  join** de R y S sobre las columnas i y j

- Se denota  $R \bowtie_{i \theta j} S$  donde  $\theta$  es un operador de comparación aritmético

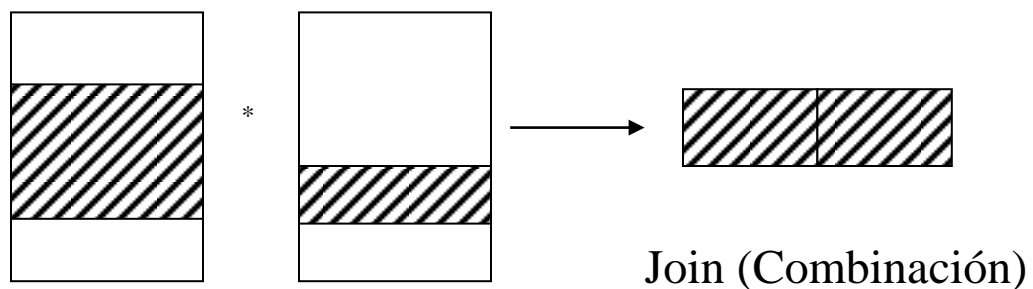
$$\theta \left\{ \begin{array}{l} < \\ = \\ > \\ \text{etc.} \end{array} \right.$$

- Es equivalente a:

$$\sigma_{i \theta j} (R \times S)$$

r es la aridad de R

- La i-ésima columna de R está en relación  $\theta$  con la j-ésima columna de S.



## ii. Equijoin

- Si  $\theta \equiv '='$ , la operación se denomina **Equijoin**.

### Ejemplo 8:

R			S	
<u>Nombre</u>	<u>Edad</u>	<u>Sueldo</u>	<u>Nombre</u>	<u>Depto</u>
Pérez	25	120.000	Pérez	Ventas
Fuentes	36	200.000	López	Finanzas
López	45	180.000	Fuentes	Producción

$$R \bowtie S \equiv \sigma_{\$1 = \$4} (R \times S)$$

<u>R.Nombre</u>	<u>Edad</u>	<u>Sueldo</u>	<u>S.Nombre</u>	<u>Depto</u>
Pérez	25	120.000	Pérez	Ventas
Fuentes	36	200.000	Fuentes	Producción
López	45	180.000	López	Finanzas


Ejemplo 9:

<b>A</b>	<b>B</b>	<b>C</b>
1	2	3
4	5	6
7	8	9

**R**

<b>D</b>	<b>E</b>
3	1
6	2

**S**

**R**  **S**  
B < D

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

**Ejemplo 10:**

**R:**

<b>A</b>	<b>B</b>	<b>C</b>
1	2	3
4	5	6
7	5	3
8	4	6

**S:**

<b>D</b>	<b>B</b>
a	2
b	5

$$\mathbf{R} \bowtie \mathbf{S}$$

$$2 = 2$$

<b>A</b>	<b>R.B</b>	<b>C</b>	<b>D</b>	<b>S.B</b>
1	2	3	a	2
4	5	6	b	5
7	5	3	b	5

### iii. Join Natural

- El join natural, escrito  $R \bowtie S$ , es aplicable solamente cuando las relaciones  $R$  y  $S$  tienen atributos con el mismo nombre.
- El cálculo de  $R \bowtie S$  procede de la siguiente forma:
  - i. Calcular  $R \times S$ .
  - ii. Para cada atributo  $A$ , en  $R$  y en  $S$ , se seleccionan las tuplas  $R \times S$  cuyos valores coinciden en las columnas  $R.A$  y  $S.A$ .
  - iii. Para cada atributo  $A$ , se elimina la columna  $S.A$ .
- Luego, si  $A_1, A_2, \dots, A_k$  son todos los nombres de atributos usados tanto en  $R$  y en  $S$

$$R \bowtie S = \pi_{i_1, i_2, \dots, i_m} \sigma_{R.A_1 = S.A_1 \wedge \dots \wedge R.A_k = S.A_k} (R \times S)$$

donde:

$i_1, i_2, \dots, i_m$  es la lista de todos los componentes de  $R \times S$ , en orden, excepto los componentes  $S.A_1, \dots, S.A_k$

**Ejemplo 11:**

<b>R</b>			<b>S</b>		
<b>A</b>	<b>B</b>	<b>C</b>	<b>B</b>	<b>C</b>	<b>D</b>
a	b	c	b	c	d
d	b	c	b	c	e
b	b	f	a	d	b
c	a	d			

**R**  $\bowtie$  **S**

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
a	b	c	d
a	b	c	e
d	b	c	d
d	b	c	e
c	a	d	b

$$R \bowtie S = \Pi_{A, R.B, R.C, D} \sigma_{R.B = S.B \wedge R.C = S.C} (R \times S)$$

iv. **Semijoin** ( $\bowtie$ )

- $R \ltimes S \equiv \pi_R(R \bowtie S)$
- Permite obtener las tuplas de R que participan en el join de R con S
- $R \ltimes S = R \bowtie \pi_{R \cap S}(S)$
- $R \ltimes S \neq S \ltimes R$

**Ejemplo 12:**

R			S			R $\bowtie$ S		
A	B	C	B	C	D	A	B	C
a	b	c	b	c	d	a	b	c
d	b	c	b	c	e	d	b	c
b	b	f	a	d	b	c	a	d
c	a	d						

$\pi_{R \cap S}(S)$			$\longrightarrow$		
B	C				
b	c				
a	d				

R $\ltimes$ $\pi_{R \cap S}(S)$			$\longrightarrow$		
A	B	C			
a	b	c			
d	b	c			
c	a	d			



**Ejemplo 13:**

<b>R:</b>	<b>A</b>	<b>B</b>	<b>C</b>
	b	c	d
	b	c	e
	a	d	b

<b>X</b>	<b>Y</b>	<b>Z</b>
b	c	d
b	c	e
a	d	b

<b>Z</b>	<b>V</b>	<b>W</b>
b	c	d
b	c	e
a	d	b

**R(X, Y, Z)    $\bowtie$    R(Z, V, W)**

**$\Downarrow$**

<b>X</b>	<b>Y</b>	<b>Z</b>	<b>V</b>	<b>W</b>
a	d	b	c	d
a	d	b	c	e

### v. Outer join

- $R \bowtie S$
- $R \Join S$

El outer join es un operador adicional del álgebra relacional. No puede ser expresado en término de los 5 operadores básicos.

### Ejemplo 14:

- **Alumno** (rut, nombreAl)
- **Asignatura** (rut, nombreAsig)

Se desea obtener el nombre del alumno, seguido del nombre de la asignatura.

Si existe un alumno, que no tiene asignaturas asociadas, igual se debe desplegar su nombre, seguido de un valor nulo para el atributo nombre de la asignatura.

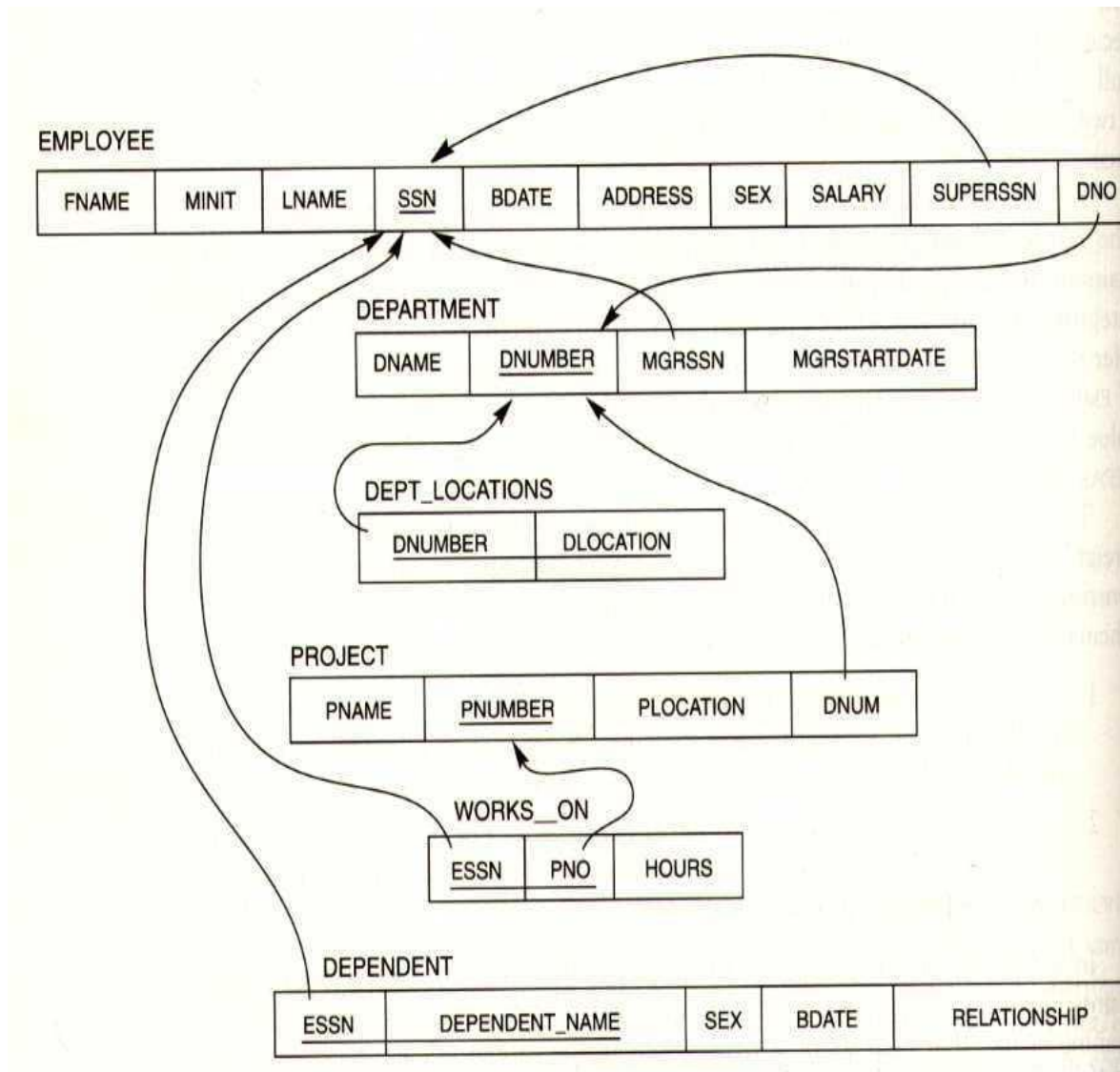
$$\pi_{\text{nombreAl}, \text{nombreAsig}} (\text{Alumno} \Join \text{Asignatura})$$

### Propiedad de Clausura:

Toda operación sobre una o más relaciones tiene como resultado una relación.

### 3.2.1.3 Ejercicios

1. Dado el modelo relacional para la empresa COMPANY:



**Esquema de la base de datos COMPANY**

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	09-JAN-55	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	08-DEC-45	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	19-JUL-58	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	20-JUN-31	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	15-SEP-52	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	31-JUL-62	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	29-MAR-59	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	10-NOV-27	450 Stone, Houston, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	22-MAY-78
	Administration	4	987654321	01-JAN-85
	Headquarters	1	888665555	19-JUN-71

WORKS_ON	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	05-APR-76	DAUGHTER
	333445555	Theodore	M	25-OCT-73	SON
	333445555	Joy	F	03-MAY-48	SPOUSE
	987654321	Abner	M	29-FEB-32	SPOUSE
	123456789	Michael	M	01-JAN-78	SON
	123456789	Alice	F	31-DEC-78	DAUGHTER
	123456789	Elizabeth	F	05-MAY-57	SPOUSE

**Instancia de la base de datos COMPANY**

- a) Recuperar el nombre y la dirección de los empleados que trabajan para el departamento “Research”.

$$\pi_{\text{Fname}, \left[ \begin{array}{l} \text{Minit}, \\ \text{Lname}, \\ \text{Address} \end{array} \right]} \left( (\sigma_{\text{Dname}='RESEARCH'} \text{DEPARTMENT}) \bowtie \text{EMPLOYEE} \right)$$

DNUMBER = DNO

ó

$$\pi_{\text{Fname}, \left[ \begin{array}{l} \text{Minit}, \\ \text{Lname}, \\ \text{Address} \end{array} \right]} \left( \sigma_{\text{Dname}='RESEARCH'} (\text{DEPARTMENT} \bowtie \text{EMPLOYEE}) \right)$$

DNUMBER = DNO

*Es más eficiente la primera solución*

- b) Encontrar el nombre de los empleados que trabajan en algún proyecto controlados por el departamento número 5.

$$\pi_{\text{Fname}, \left[ \begin{array}{l} \text{Minit}, \\ \text{Lname} \end{array} \right]} \left( ((\sigma_{\text{DNUM}=5} \text{PROJECT}) \bowtie \text{WORKS\_ON}) \bowtie \text{EMPLOYEE} \right)$$

PNUMBER = PNO

ESSN = SSN

- c) Para cada proyecto localizado en “STAFFORD”, indicar el número del proyecto, el número del departamento que lo controla y el nombre, fecha de nacimiento y dirección del jefe del departamento que controla el proyecto.

$$\pi_{\text{Pnumber, dnum, Minit, Lname, Bdate, Address}} \left( ((\sigma_{\text{LOCATION} = \text{'STAFFORD'}} \text{PROJECT}) \bowtie_{\text{DNUM} = \text{DNUMBER}} \text{DEPARTMENT}) \bowtie_{\text{MGRSSN} = \text{SSN}} \text{EMPLOYEE} \right)$$

- d) Para cada empleado, indicar el nombre del empleado y el nombre de su jefe inmediato. Considerar que aquellos empleados que no tienen un jefe, también deben aparecer en el resultado de la consulta.

Se debe crear un Alias para la tabla EMPLOYEE.

Crear una copia (EMPLOYEE1)

$$\pi_{\text{EMPLOYEE.LNAME, EMPLOYEE1.LNAME}} \left( \text{EMPLOYEE} \bowtie_{\text{EMPLOYEE.SUPERSSN} = \text{EMPLOYEE1.SSN}} \text{EMPLOYEE1} \right)$$

2. Se tiene el siguiente esquema:

**Profesor** (rut, nombre, apellido, dirección, categoría)

**Curso** (sigla, descripción, créditos, vacantes)

**Dicta** (rut, sigla, semestre, sala, hora)

Se desea obtener el Apellido y Categoría de los profesores que dictan cursos de más de 12 créditos en el 2º semestre. Además, se desea saber a que hora y en que sala son dictados dichos cursos.

$$\pi_{\substack{\text{Apellido,} \\ \text{Categoría,} \\ \text{Sala,} \\ \text{Hora}}} \{ [(\sigma_{\text{creditos} > 12} \text{Curso}) \bowtie_{\text{sigla}} (\sigma_{\text{Semestre} = 2} \text{Dicta})] \bowtie_{\text{rut}} \text{Profesor} \}$$

### Consideraciones finales:

- **Optimizador de consultas**
- Para la operación de proyección, los sistemas relacionales deben preocuparse de eliminar las posibles tuplas duplicadas que resultan de una proyección.

### 3.2.2 Cálculo Relacional

- El cálculo relacional es un lenguaje no procedural (declarativo o descriptivo), en que se da una descripción formal de la información deseada, sin especificar cómo obtenerla.
- Existen dos formas de cálculo relacional:
  - Una en que las variables representan **tuplas**.
  - Otra, en que las variables representan **valores de dominios**.
- Estas variantes se denominan:
  - **Cálculo relacional de tuplas.**
  - **Cálculo relacional de dominios.**
- Basado en una rama de la lógica matemática llamada el **cálculo de predicados**.



### 3.2.2.1 Cálculo relacional de tuplas (CRT)

#### 3.2.2.1.1 Especificación formal del CRT

- El cálculo relacional de tuplas está basado en un número específico de **variables de tuplas**.
- Cada variable de tupla, usualmente está en el rango de una relación de la base de datos particular.

**$\{t \mid \text{COND}(t)\}$**

- $t$  es una variable de tupla
- $\text{COND}(t)$  es una expresión condicional que involucra a  $t$ .
- El resultado de tal consulta es el conjunto de todas las tuplas  $t$  que satisfacen a  $\text{COND}(t)$ .

**Ejemplo:** Encontrar todos los empleados cuyo salario está sobre \$50.000

$\{t \mid \text{EMPLOYEE}(t) \text{ and } t.\text{SALARY} > 50.000\}$

$\{t.\text{FNAME}, t.\text{LNAME} \mid \text{EMPLOYEE}(t) \text{ and } t.\text{SALARY} > 50.000\}$

```

SELECT    T.FNAME, T.LNAME
FROM      EMPLOYEE T
WHERE      T.SALARY > 50.000
  
```

Informalmente, se necesita especificar en el CRT:

1. Para cada variable de tupla  $t$ , **la relación  $R$  correspondiente al rango de  $t$ . ( $R(t)$ ).**
2. Una **condición** para seleccionar combinaciones particulares de tuplas.
3. **El conjunto de atributos a recuperar** llamados, los atributos de respuesta.

### **Ejemplo:**

Recuperar la fecha de nacimiento y la dirección del empleado (o empleados) cuyo nombre sea ‘John B. Smith’.

$$\{t.BDATE, t.ADDRESS \mid \text{EMPLOYEE}(t) \text{ and } \\ t.FNAME = \text{'John'} \text{ and } t.MINIT = \text{'B'} \\ \text{and } t.LNAME = \text{'Smith'}\}$$

Una expresión general para el CRT es la siguiente:

$$\{t_1.A_1, t_1.A_2, \dots, t_n.A_n \mid \text{COND}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m})\}$$

donde:

- $t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m}$  son variables de tuplas.
- Cada  $A_i$  es un atributo en alguna de las relaciones asociadas a los rangos de los  $t_i$ .
- COND es la condición o fórmula del CRT.

Una fórmula está hecha de **átomos** del cálculo de predicados, que pueden ser uno de los siguientes:

1. Un átomo de la forma  **$R(t_i)$**
2. Un átomo de la forma  **$t_i.A \text{ op } t_j.B$**
3. Un átomo de la forma  **$t_i.A \text{ op } c \text{ ó } c \text{ op } t_j.B$**

Una **fórmula** (condición) está hecha de uno o más átomos conectados vía operadores lógicos **and**, **or**, y, **not** y se define recursivamente como sigue:

1. Cada átomo es una fórmula.
2. Si  $F_1$  y  $F_2$  son fórmulas, entonces lo son:
  - $(F_1 \text{ and } F_2)$
  - $(F_1 \text{ or } F_2)$
  - **not**  $(F_1)$
  - **not**  $(F_2)$ .

Además, dos símbolos especiales llamados cuantificadores pueden aparecer en fórmulas; son:

- el **cuantificador universal**  $(\forall)$        $\forall t (f(t))$
- el **cuantificador existencial**  $(\exists)$        $\exists t (f(t))$

Informalmente, una variable de tupla  $t$  está:

- **Ligada** si está cuantificada, o sea, que aparece en una cláusula  $(\exists t)$  ó  $(\forall t)$ .
- De otro modo, es **libre**.

Formalmente, definimos una variable de tupla en una fórmula como libre ó ligada, de acuerdo con las siguientes reglas:

- Una ocurrencia de una variable de tupla en una fórmula  $F$  que es un átomo, está libre en  $F$ .
- Una ocurrencia de una variable de tupla  $t$  está libre ó ligada en una fórmula del tipo: **( $F_1$  and  $F_2$ )**, **( $F_1$  or  $F_2$ )**, **not ( $F_1$ )** y **not ( $F_2$ )**, dependiendo de si está libre o ligada en  $F_1$  ó  $F_2$ .
- Todas las ocurrencias libres de una variable de tupla  $t$  en  $F$  son ligadas en una fórmula  $F'$  de la forma  $F' = (\exists t)(F)$  ó  $F' = (\forall t)(F)$ . La variable de tupla está ligada al cuantificador especificado en  $F'$ .

**Por ejemplo:**

$F_1 : d.NAME = \text{'Research'}$

$F_2 : (\exists t)(d.DNUMBER = t.DNO)$

*t: ligada*

*d: libre*

3. Si  $F$  es una fórmula, también lo es  $(\exists t)(F)$ , donde  $t$  es una variable de tupla. La fórmula  $(\exists t)(F)$  es:

- TRUE si la fórmula  $F$  se evalúa TRUE para alguna (al menos una) tupla asignada a ocurrencias libres de  $t$  en  $F$ .
- De otro modo  $(\exists t)(F)$  es FALSE.

4. Si  $F$  es una fórmula, también lo es  $(\forall t)(F)$ , donde  $t$  es una variable de tupla. La fórmula  $(\forall t)(F)$  es:

- TRUE si la fórmula  $F$  se evalúa TRUE para **cada tupla** (en el universo) asignada a ocurrencias libres de  $t$  en  $F$ .
- De otro modo  $(\forall t)(F)$  es FALSE.

## Transformación de los cuantificadores universal y existencial

- $(\forall x) (P(x)) \equiv (\exists x) (\text{not } (P(x)))$
- $(\exists x) (P(x)) \equiv \text{not } (\forall x) (\text{not}(P(x)))$
- $(\forall x) (P(x)) \text{ and } Q(x) \equiv (\exists x) (\text{not } (P(x)) \text{ or } \text{not } (Q(x)))$
- $(\forall x) (P(x) \text{ or } Q(x)) \equiv (\exists x) (\text{not } (P(x)) \text{ and } \text{not } (Q(x)))$
- $(\exists x) (P(x)) \text{ or } Q(x) \equiv \text{not } (\forall x) (\text{not } (P(x)) \text{ and } \text{not } Q(x))$
- $(\exists x) (P(x)) \text{ and } Q(x) \equiv \text{not } (\forall x) (\text{not } P(x)) \text{ or } \text{not } (Q(x))$

## Resumiendo:

### 1. Variables de tupla

$$\{t \mid \text{COND}(t)\}$$

$$\{t \mid \text{EMPLOYEE}(t) \text{ AND } t.\text{SALARY} > 50.000\}$$

$$\{t.\text{FNAME}, t.\text{LNAME} \mid \text{EMPLOYEE}(t) \text{ AND } t.\text{SALARY} > 50.000\}$$

### 2. Especificación formal

- $\{t_1.A_1, t_2.A_2, \dots, t_n.A_n \mid \text{COND}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m})\}$

donde: COND, es una condición o fórmula

- Una fórmula compuesta de átomos:

a)  $R(t_i)$

b)  $t_i.A \text{ op } t_j.B$

c)  $t_i.A \text{ op } C \quad C \text{ op } t_j.B$

- Una fórmula (o condición) se construye con átomos conectados por: AND, OR, NOT

- También, pueden usarse:  $\forall, \exists$

- Si F es una fórmula, también lo es:

$$\exists t(F)$$

$$\forall t(F)$$

$$\exists t(F(t))$$

$$\forall t(F(t))$$



### 3.2.2.1.2 Ejercicios

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	09-JAN-55	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	08-DEC-45	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	19-JUL-58	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	20-JUN-31	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	15-SEP-52	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	31-JUL-62	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	29-MAR-59	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	10-NOV-27	450 Stone, Houston, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	22-MAY-78
	Administration	4	987654321	01-JAN-85
	Headquarters	1	888665555	19-JUN-71

WORKS_ON	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	05-APR-76	DAUGHTER
	333445555	Theodore	M	25-OCT-73	SON
	333445555	Joy	F	03-MAY-48	SPOUSE
	987654321	Abner	M	29-FEB-32	SPOUSE
	123456789	Michael	M	01-JAN-78	SON
	123456789	Alice	F	31-DEC-78	DAUGHTER
	123456789	Elizabeth	F	05-MAY-57	SPOUSE

**Esquema de la base de datos COMPANY**

1. Recuperar el nombre y la dirección de los empleados que trabajan para el departamento 'RESEARCH'.

{t.FNAME, t.LNAME, t.ADDRESS | EMPLOYEE(t) AND  
 (( $\exists$  d) (DEPARTMENT(d) AND  
 d.DNAME = 'RESEARCH' AND  
 d.DNUMBER = t.DNO) ) }

2. Para cada proyecto localizado en 'STAFFORD', indicar el número del proyecto, el número del departamento que lo controla, el nombre, fecha de nacimiento y dirección del jefe del departamento que controla al proyecto.

{p.PNUMBER, p.DNUM, m.LNAME, m.BDATE, m.ADDRESS |  
 PROJECT(p) AND EMPLOYEE(m) AND  
 p.LOCATION = 'STAFFORD' AND  
 (( $\exists$  d) (DEPARTMENT(d) AND  
 p.DNUM = d.DNUMBER AND  
 d.MGRSSN = m.SSN) ) }

3. Para cada empleado, recuperar su nombre y el nombre de su jefe inmediato.

$$\{e.FNAME, e.LNAME, s.FNAME, s.LNAME \mid$$

$$EMPLOYEE(e) \text{ AND}$$

$$EMPLOYEE(s) \text{ AND}$$

$$e.SUPERSSN = s.SSN\}$$

4. Encontrar el nombre de los empleados que trabajan en algún proyecto controlado por el departamento número 5.

$$\{e.LNAME, e.FNAME \mid EMPLOYEE(e) \text{ AND}$$

$$((\exists x)(\exists w) (PROJECT(x) \text{ AND}$$

$$WORKS\_ON(w) \text{ AND}$$

$$x.DNUM = 5 \text{ AND}$$

$$w.ESSN = e.SSN \text{ AND}$$

$$x.PNUMBER = w.PNO)) \}$$

5. Listar los números de proyectos que tienen a un empleado con nombre 'SMITH', ya sea como un integrante del proyecto o como jefe del departamento que controla el proyecto.

{p.PNUMBER | PROJECT(p) AND

( ( (∃e)( ∃w) (EMPLOYEE(e) AND  
WORKS\_ON(w) AND  
w.PNO = p.PNUMBER AND  
e.LNAME = 'SMITH' AND  
e.SSN = w.ESSN)

*Empleado  
Smith como  
integrante  
del proyecto*

OR

( (∃m)( ∃d) (EMPLOYEE(m) AND  
DEPARTMENT(d) AND  
p.DNUM = d.DNUMBER AND  
d.MGRSSN = m.SSN AND  
m.LNAME = 'SMITH')

*Empleado  
Smith como  
jefe del  
departamento  
que controla el  
proyecto*

)

}

6. Encontrar los nombres de los empleados que trabajan en todos los proyectos controlados por el departamento número 5.

$$\{e.LNAME, e.FNAME \mid EMPLOYEE(e) \text{ AND} \\ ((\forall x) (PROJECT(x) \text{ AND} \\ x.DNUM = 5 \text{ AND} \\ ((\exists w) (WORKS\_ON(w) \text{ AND} \\ w.ESSN = e.SSN \text{ AND} \\ x.PNUMBER = w.PNO) \\ ) \\ ) \\ ) \\ }$$

7. Encontrar los nombres de los empleados que no tienen dependientes.

$$\{e.FNAME, e.LNAME \mid EMPLOYEE(e) \text{ AND} \\ (\text{NOT } (\exists d) (\text{DEPENDENT}(d) \text{ AND} \\ e.SSN = d.ESSN) \\ ) \\ \}$$

$$(\forall x)(P(x) \text{ or } Q(x)) \equiv (\cancel{\exists x} (\text{not } (P(x)) \text{ and not } Q(x)))$$

$$\{e.FNAME, e.LNAME \mid EMPLOYEE(e) \text{ AND} \\ (\forall d) (\text{not } \text{DEPENDENT}(d) \text{ or not } (e.SSN = d.ESSN)) \\ \}$$

8. Listar los nombres de los jefes de departamento que tienen al menos un dependiente

$$\{e.FNAME, e.LNAME \mid EMPLOYEE(e) \text{ AND} \\ ((\exists d) (\exists p) (\text{DEPARTMENT}(d) \text{ AND } \text{DEPENDENT}(p) \text{ AND} \\ e.SSN = d.MGRSSN \text{ AND } p.ESSN = e.SSN)) \\ \}$$

## Expresiones Seguras

- Una expresión **segura** en cálculo relacional es aquella que garantiza un número finito de tuplas como resultado.
- De otro modo, la expresión se llama **insegura**.

**Ejemplo:**  $\{t \mid \text{NOT}(\text{EMPLOYEE}(t))\}$  es insegura.

- **Dominio** es el conjunto de todos los valores que, o aparecen como valores constantes en la expresión, o existen en cualquier tupla de las relaciones referenciadas en la expresión.

**P:** fórmula

**Dom(P):** Conjunto de todos los valores a los que se refiere con P.

- Una expresión se dice **segura**, si todos los valores en su resultado se encuentran en el dominio de la expresión.

### 3.2.2.2 Cálculo relacional de dominios (CRD)

#### 3.2.2.2.1 Especificación formal del CRD

- Existe un lenguaje de consulta comercial, **QBE**.
- En el cálculo de dominio, las variables toman sus valores desde el dominio de los atributos.
- Para formar una relación de grado  $n$  correspondiente al resultado de una consulta, debemos tener  $n$  de estas variables de dominios, una para cada atributo.
- Una expresión de cálculo de dominios es de la forma:
 
$$\{x_1, x_2, \dots, x_n \mid \text{COND}(x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+m})\}$$
 donde:
  - $x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+m}$  son variables de dominio que toman valores desde el dominio (de los atributos)
  - COND es una condición o fórmula del cálculo relacional de dominios.



Una fórmula está hecha de átomos. Los átomos de una fórmula pueden ser uno de los siguientes:

1. Un átomo de la forma  $R(x_1, x_2, \dots, x_j)$ , donde:
  - $R$  es el nombre de la relación de grado  $j$ .
  - Cada  $x_i$ ,  $1 \leq i \leq j$ , es una variable de dominio.

$$\star \{ x_1 x_2 \dots x_n \mid R(x_1 x_2 x_3) \textbf{ and} \dots \}$$

$$\{ x_1, x_2, \dots, x_n \mid R(x_1, x_2, x_3) \textbf{ and} \dots \}$$

2. Un átomo de la forma  $x_i \textbf{ op } x_j$ , donde:
  - $\textbf{op}$  es un operador de comparación en el conjunto  $\{=, \neq, <, \leq, <, \geq\}$
  - $x_i$  y  $x_j$  son variables de dominio.
3. Un átomo de la forma  $x_i \textbf{ op } c \textbf{ ó } c \textbf{ op } x_j$ , donde:
  - $\textbf{op}$  es un operador de comparación en el conjunto  $\{=, \neq, <, \leq, <, \geq\}$
  - $x_i$  y  $x_j$  son variables de dominio
  - $c$  es un valor constante.

**En resumen:**

$$\{x_1, x_2, \dots, x_n \mid \text{COND}(x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+m})\}$$

La fórmula tiene:

- **Átomos:**

1.  $R(x_1, x_2, \dots, x_j)$

Lista de valores  $\langle x_1, x_2, \dots, x_j \rangle$

$$\{x_1 x_2 \dots x_n \mid R(x_1 x_2 x_3) \text{ and } \dots\}$$

$$\{x_1, x_2, \dots, x_n \mid R(x_1 x_2 x_3) \text{ and } \dots\}$$

2.  $x_i \text{ op } x_j$

3.  $x_i \text{ op } c \qquad c \text{ op } x_j$

- **Cuantificadores ( $\forall$  y  $\exists$ )**

## Ejercicios

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	09-JAN-55	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	08-DEC-45	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	19-JUL-58	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	20-JUN-31	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	15-SEP-52	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	31-JUL-62	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	29-MAR-59	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	10-NOV-27	450 Stone, Houston, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	22-MAY-78
	Administration	4	987654321	01-JAN-85
	Headquarters	1	888665555	19-JUN-71

WORKS_ON	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	05-APR-76	DAUGHTER
	333445555	Theodore	M	25-OCT-73	SON
	333445555	Joy	F	03-MAY-48	SPOUSE
	987654321	Abner	M	29-FEB-32	SPOUSE
	123456789	Michael	M	01-JAN-78	SON
	123456789	Alice	F	31-DEC-78	DAUGHTER
	123456789	Elizabeth	F	05-MAY-57	SPOUSE

## Esquema de la base de datos COMPANY

1. Recuperar la fecha de nacimiento y la dirección de los empleados cuyo nombre sea 'John B. Smith'.

$$\{u \ v \mid (\exists q) (\exists r) (\exists s) (\text{EMPLOYEE}(q \ r \ s \ t \ u \ v \ w \ x \ y \ z) \text{ and } q = \text{'John'} \text{ and } r = \text{'B'} \text{ and } s = \text{'Smith'})\}$$

$$\{u \ v \mid \text{EMPLOYEE}(\text{'John'}, \text{'B'}, \text{'Smith'}, t, u, v, w, x, y, z)\}$$

2. Recuperar el nombre y la dirección de todos los empleados que trabajan para el departamento 'Research'.

$$\{q \ s \ v \mid (\exists z) (\text{EMPLOYEE}(q \ r \ s \ t \ u \ v \ w \ x \ y \ z) \text{ and } (\exists l) (\exists m) (\text{DEPARTMENT}(l \ m \ n \ o) \text{ and } l = \text{'research'} \text{ and } m = z) ) \}$$

3. Para cada proyecto localizado en 'Stafford', indicar el número del proyecto, el número del departamento que lo controla y el nombre, fecha de nacimiento y dirección del jefe del departamento.

$$\{i \ k \ s \ u \ v \mid (\exists j) (\text{PROJECT}(h \ i \ j \ k) \text{ and} \\
(\exists t) (\text{EMPLOYEE}(q \ r \ s \ t \ u \ v \ w \ x \ y \ z) \text{ and} \\
(\exists m) (\exists n) (\text{DEPARTMENT}(l \ m \ n \ o) \text{ and} \\
k = m \text{ and } n = t \text{ and } j = \text{'STAFFORD'}) \\
) \\
) \\
\}$$

4. Encontrar los nombre de los empleados que no tienen dependientes

$$\{q \ s \mid (\exists t) (\text{EMPLOYEE}(q \ r \ s \ t \ u \ v \ w \ x \ y \ z) \text{ and} \\
(\text{not } (\exists a) (\text{DEPENDENT}(a \ b \ c \ d \ e) \text{ and } t = a)) \\
\}$$

### 3.2.3 Consideraciones fundamentales

- En un aspecto importante, el álgebra relacional y el cálculo relacional son idénticos.
- Se ha demostrado que cualquier recuperación de información que puede ser especificada en el álgebra relacional, también lo puede ser en el cálculo relacional, y viceversa.
- Concepto de **lenguaje relacionalmente completo**.
- Un lenguaje de consulta relacional L se llama **relacionalmente completo**, si se puede expresar en L cualquier consulta que se puede expresar en el cálculo relacional.
- La mayoría de los lenguajes de consulta relacionales, son relacionalmente completos, pero tienen **más poder expresivo** que el álgebra relacional o el cálculo relacional, debido a la incorporación de:
  - Comandos de inserción, eliminación y modificación.
  - Capacidad aritmética. Ejemplo:  $A < B \pm 3$
  - Comandos de asignación e impresión.
  - Funciones de agregación: promedio, suma, menor o mayor.

- Los lenguajes relacionales comerciales están basados en:
  - Álgebra relacional.
  - Cálculo relacional de dominios.
  - Cálculo relacional de tuplas.
- Algunos lenguajes relacionales comerciales:
  - Lenguaje **QUEL** (Query Language)
  - Lenguaje **QBE** (Query – By – Example)
  - Lenguaje **SQL** (Structured Query Language)