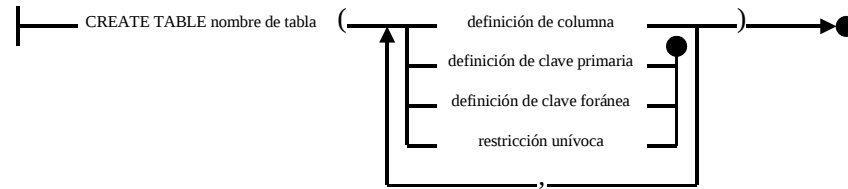


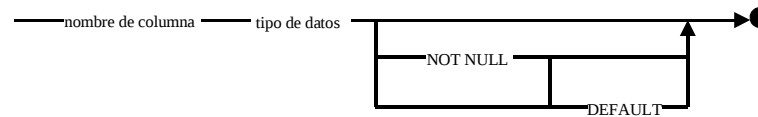
# Introducción

- La estructura de la base de datos se maneja mediante las sentencias denominadas “lenguaje de definición de datos”.
- El lenguaje de definición de datos permite:
  - Definir y crear nuevas tablas: **CREATE TABLE**
  - Eliminar tablas: **DROP TABLE**
  - Modificar las tablas: **ALTER TABLE**
  - Crear tablas virtuales (vistas): **CREATE VIEW**
  - Establecer controles de seguridad: **GRANT** y **REVOKE**

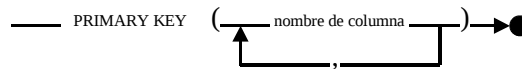
# Creación de tablas



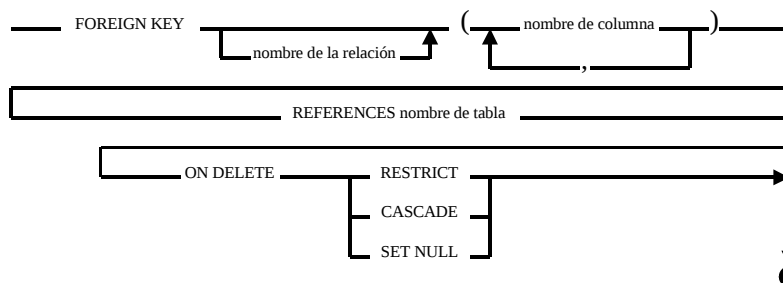
definición de columna:



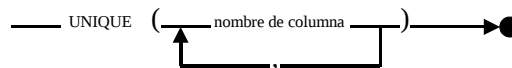
definición de clave primaria:



definición de clave foránea:



restricción unívoca:



## Definición de los campos (I)

- Se realiza mediante una lista separada por el símbolo coma.
- El orden de la definición determina el orden de los campos en la tabla.
- La definición de cada campo especifica:
  - El nombre.
  - El tipo de datos.
  - Si puede contener valores nulos (por defecto) o no (**NOT NULL**).
  - Opcionalmente, un valor del campo si en una inserción de filas omitimos el valor de este campo.

## Definición de los campos (II)

```
CREATE TABLE oficinas (  
    oficina INTEGER NOT NULL,  
    ciudad VARCHAR(15) NOT NULL,  
    region VARCHAR(10) NOT NULL,  
    objetivo INTEGER,  
    ventas INTEGER NOT NULL  
)
```

OFICINA	CIUDAD	REGION	OBJETIVO	VENTAS
---------	--------	--------	----------	--------

## Definición de los campos (III)

```
CREATE TABLE pedidos (  
    num_pedido INTEGER NOT NULL,  
    fecha_pedido DATE NOT NULL,  
    clie INTEGER NOT NULL,  
    rep INTEGER,  
    fab CHAR(3) NOT NULL,  
    producto CHAR(5) NOT NULL,  
    cant INTEGER NOT NULL,  
    importe INTEGER NOT NULL  
)
```

NUM_PEDIDO	FECHA_PEDIDO	CLIE	REP	FAB	PRODUCTO	CANT	IMPORTE
------------	--------------	------	-----	-----	----------	------	---------

## Valores por omisión

- Es posible establecer un valor por defecto para cada campo.
- Formato diferente en ANSI/ISO SQL y SQL de IBM:

```
CREATE TABLE oficinas (  
    oficina INTEGER NOT NULL,  
    ciudad VARCHAR(15) NOT NULL,  
    region VARCHAR(10) NOT NULL DEFAULT 'Este',  
    objetivo INTEGER,  
    ventas INTEGER NOT NULL DEFAULT 0  
)
```

```
CREATE TABLE oficinas (  
    oficina INTEGER NOT NULL,  
    ciudad VARCHAR(15) NOT NULL,  
    region VARCHAR(10) NOT NULL WITH DEFAULT,  
    objetivo INTEGER,  
    ventas INTEGER NOT NULL WITH DEFAULT  
)
```

## Definición de clave primaria (I)

- Se realiza mediante **PRIMARY KEY**.
- Define un campo o agrupación de campos que tendrán valor único para cada elemento de la tabla.
- La definición de los campos, para todos los campos que forman la clave primaria, debe especificar que no pueden tomar valor nulo (**NOT NULL**).

## Definición de clave primaria (II)

```
CREATE TABLE pedidos (  
    num_pedido INTEGER NOT NULL,  
    fecha_pedido DATE NOT NULL,  
    clie INTEGER NOT NULL,  
    rep INTEGER,  
    fab CHAR(3) NOT NULL,  
    producto CHAR(5) NOT NULL,  
    cant INTEGER NOT NULL,  
    importe INTEGER NOT NULL,  
    PRIMARY KEY (num_pedido)  
)
```



## Definición de clave foránea (I)

- Se realiza mediante **FOREIGN KEY**.
- Define un campo o agrupación de campos que hacen referencia a datos de otra tabla.
- La clave foránea debe hacer referencia a una clave primaria de otra tabla.
- Los campos de la clave foránea deben ser del mismo tipo que los de la clave primaria a la que referencia.

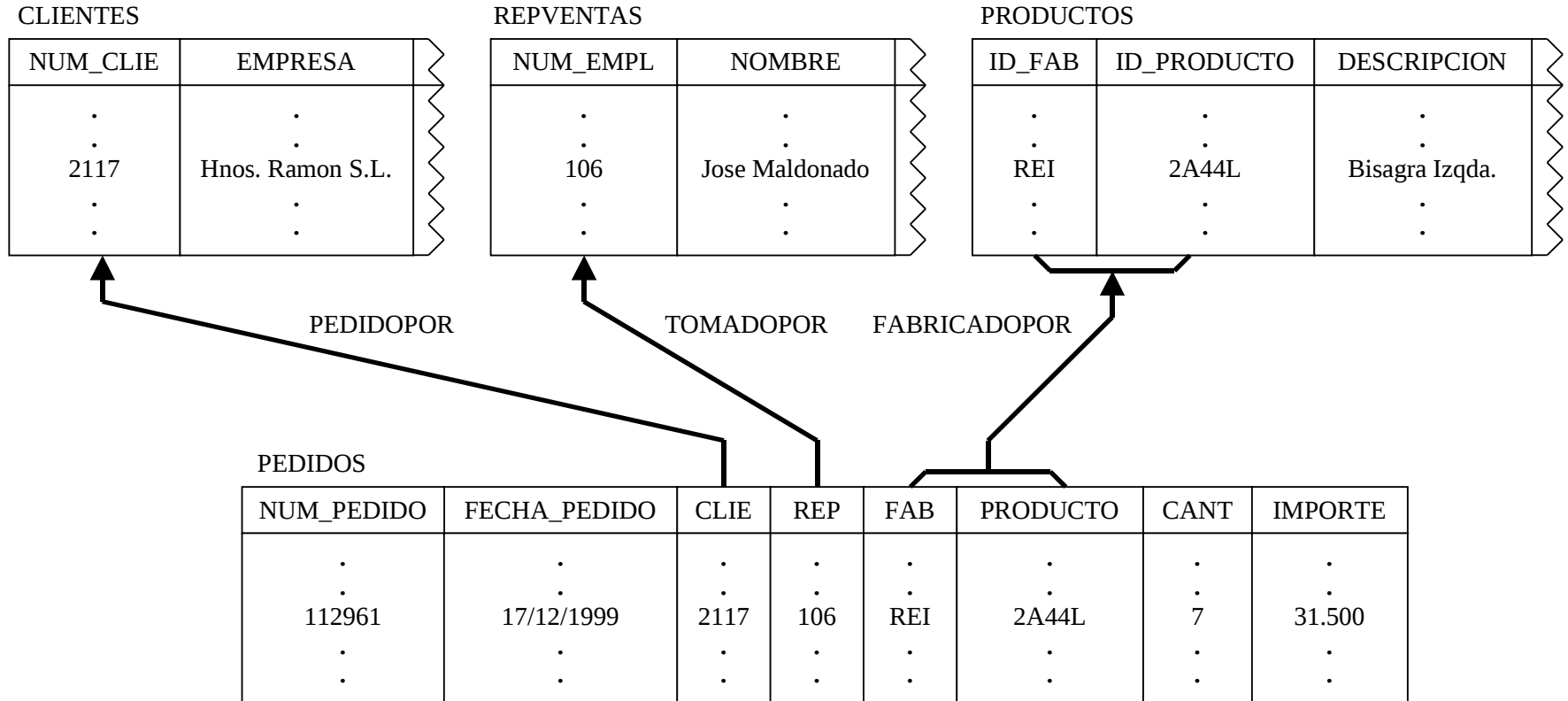
## Definición de clave foránea (II)

- La definición de clave foránea se realiza especificando:
  - El campo o campos que forman la clave foránea.
  - La tabla que es referenciada por la clave foránea.
  - Un nombre opcional para la relación.
  - Una regla de supresión para la relación en caso de que se elimine la fila referenciada.
- La regla de supresión puede ser:
  - **RESTRICT**: Impide que se suprima la fila referenciada (valor por defecto).
  - **CASCADE**: Suprime la fila que referencia.
  - **SET NULL**: Pone valores nulos a los campos de la clave foránea.

## Definición de clave foránea (III)

```
CREATE TABLE pedidos (  
    num_pedido INTEGER NOT NULL,  
    fecha_pedido DATE NOT NULL,  
    clie INTEGER NOT NULL,  
    rep INTEGER,  
    fab CHAR(3) NOT NULL,  
    producto CHAR(5) NOT NULL,  
    cant INTEGER NOT NULL,  
    importe INTEGER NOT NULL,  
    PRIMARY KEY (num_pedido),  
    FOREIGN KEY pedidopor (clie) REFERENCES clientes ON DELETE  
    CASCADE,  
    FOREIGN KEY tomadopor (rep) REFERENCES repventas ON DELETE  
    SET NULL,  
    FOREIGN KEY fabricadopor (fab, producto) REFERENCES  
    productos ON DELETE RECTRICT  
)
```

# Definición de clave foránea (IV)



## Restricciones de unicidad (I)

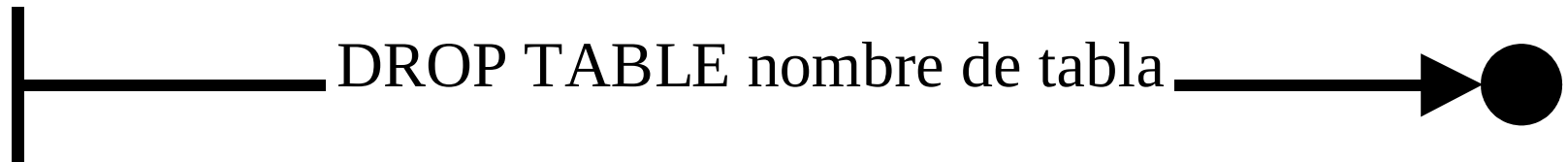
- Es posible definir campo o agrupaciones de campos que, sin ser claves primarias, también deban tener un valor único en cada elemento de la tabla.
- Se realiza mediante la palabra clave **UNIQUE**.

## Restricciones de unicidad (II)

```
CREATE TABLE oficinas (  
    oficina INTEGER NOT NULL,  
    ciudad VARCHAR(15) NOT NULL,  
    region VARCHAR(10) NOT NULL,  
    objetivo INTEGER,  
    ventas INTEGER NOT NULL,  
    PRIMARY KEY (oficina),  
    UNIQUE (ciudad)  
)
```

# Eliminación de tablas

Diagrama sintáctico:

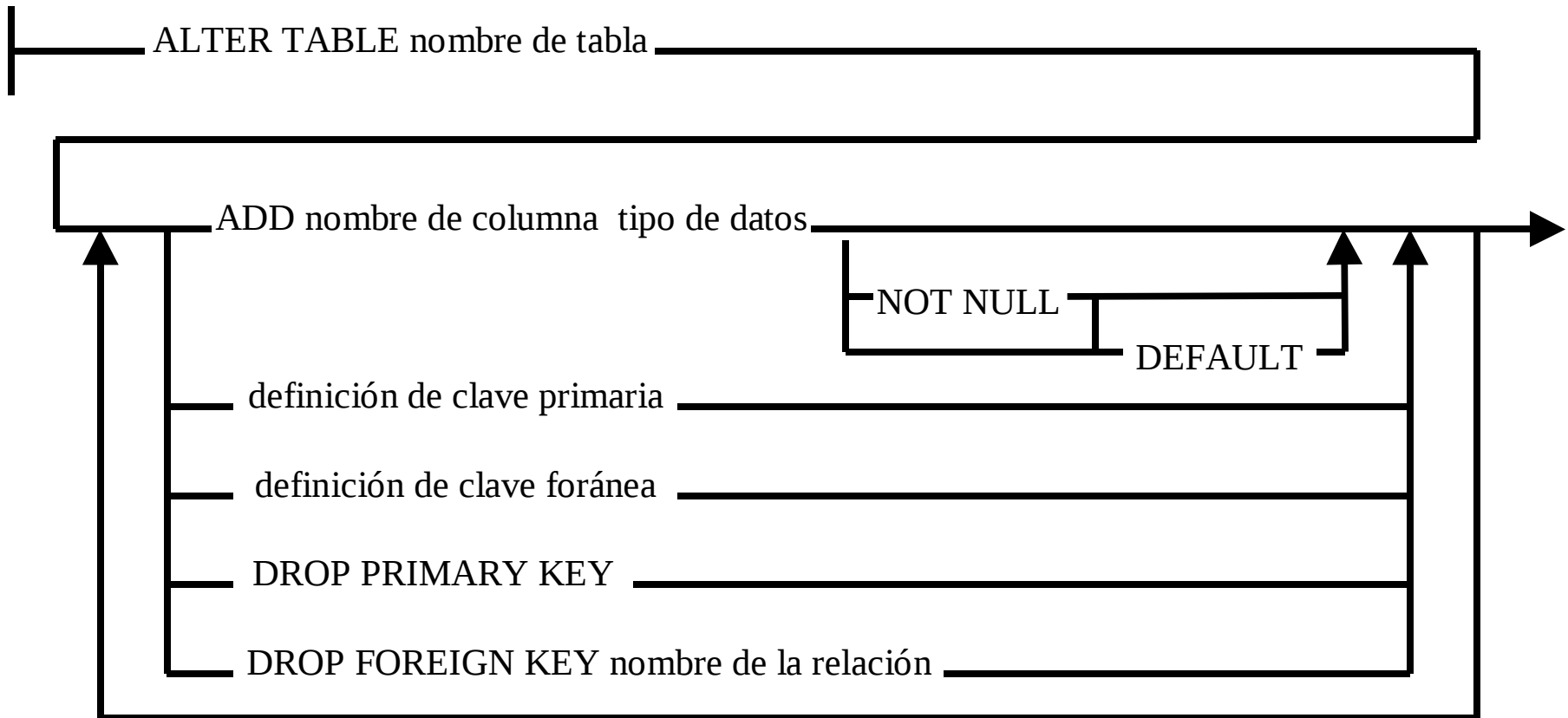


Ejemplo:

**DROP TABLE clientes**

# Modificación de tablas

## Diagrama sintáctico:





## Adición de un campo (I)

- El campo añadido se coloca a continuación del último campo existente.
- El campo añadido toma, para los elementos existentes en la tabla, los valores:
  - **NULL**, si no se indica que el campo no puede ser nulo.
  - El valor por defecto, si el campo no puede ser nulo. En este caso la definición debe llevar asociada la opción **DEFAULT**.

## Adición de un campo (II)

### Ejemplos:

```
ALTER TABLE clientes ADD telefono_contacto CHAR(9)
```

NUM_CLIE	EMPRESA	REP_CLIE	LIMITE_CREDITO	TELEFONO_CONTACTO
----------	---------	----------	----------------	-------------------

```
ALTER TABLE productos ADD cant_min INTEGER NOT NULL  
DEFAULT 0
```

ID_FAB	ID_PRODUCTO	DESCRIPCION	PRECIO	EXISTENCIAS	CANT_MIN
--------	-------------	-------------	--------	-------------	----------

## Supresión de un campo

- No existe instrucción para suprimir un campo.
- Para poder suprimir un campo hay que realizar las siguientes acciones:
  - Crear una tabla temporal sin el campo a eliminar.
  - Copiar todos los datos menos ese campo en la tabla temporal.
  - Eliminar la tabla original.
  - Volver a crear la tabla original sin el campo a eliminar.
  - Copiar los datos de la tabla temporal en la nueva tabla original.
  - Eliminar la tabla temporal.

## Modificación de claves primaria y ... (I)

- Añadir una clave primaria o foránea es similar a la acción que se realiza en su creación.
- Para eliminar una clave foránea debe haberse definido la relación con un nombre.
- Eliminar una clave primaria requiere eliminar previamente las claves foráneas que le hacen referencia.

# Modificación de claves primaria y ... (II)

## Ejemplos:

```
ALTER TABLE oficinas FOREIGN KEY enregion (region)  
REFERENCES regiones
```

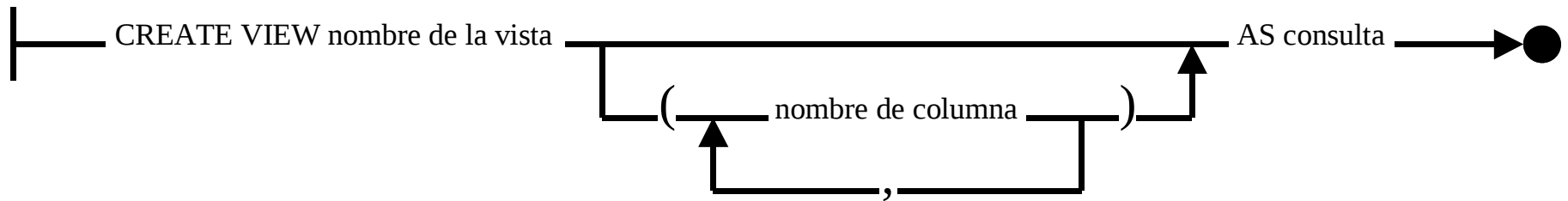
```
ALTER TABLE repventas DROP FOREIGN KEY operaen  
ALTER TABLE oficinas DROP PRIMARY KEY
```

## Vistas (I)

- Las vistas crean “tablas virtuales” que no existen físicamente en la base de datos.
- Cada vez que se arranca la BD deben crearse las vistas.
- Las vistas proporcionan beneficios evidentes como:
  - Seguridad.
  - Simplicidad de consulta.
- Sin embargo, producen problemas de:
  - Rendimiento.
  - Actualización

## Vistas (II)

### Diagrama sintáctico:



# Vistas horizontales

Permiten restringir el acceso a un conjunto de filas.

```
CREATE VIEW oficinaeste AS SELECT * FROM oficinas  
WHERE region='Este'
```

OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
11	Valencia	Este	106	52.500	40.063
12	Barcelona	Este	104	70.000	29.328
13	Alicante	Este	105	30.000	39.327

```
CREATE VIEW pedidossoledad AS SELECT * FROM pedidos  
WHERE clie IN (SELECT num_clie FROM clientes WHERE  
rep_clie=102)
```

NUM_PEDIDO	FECHA_PEDIDO	CLIE	REP	FAB	PRODUCTO	CANT	IMPORTE
113024	20/01/2000	2114	108	QSA	XK47	20	7.100
112979	12/10/1999	2114	102	ACI	4100Z	6	15.000
113048	10/02/2000	2120	102	IMM	779C	2	3.750
112993	04/01/2000	2106	102	REI	2A45C	24	1.896
113065	27/02/2000	2106	102	QSA	XK47	6	2.130



# Vistas verticales

Permiten restringir el acceso a un conjunto de campos.

```
CREATE VIEW inforep AS SELECT num_empl, nombre,  
oficina_rep FROM repventas
```

NUM_EMPL	NOMBRE	OFICINA_REP
106	Jose Maldonado	11
104	Carlos Martinez	12
105	Belen Aguirre	13
109	Maria Garcia	11
108	Lorenzo Fernandez	21
102	Soledad Martinez	21
101	Daniel Gutierrez	12
110	Antonio Valle	NULL
103	Pedro Cruz	12
107	Natalia Martin	22

## Vistas con subconjuntos fila/columna

- Una vista no tiene porque obtener subconjuntos de filas o columnas.
- Es posible obtener vistas con subconjuntos que contengan tan solo algunas filas y columnas.

```
CREATE VIEW cliebelen AS SELECT  
num_clie, empresa, limite_credito FROM clientes WHERE  
rep_clie=105
```

NUM_CLIE	EMPRESA	LIMITE_CREDITO
2103	Pino S.L.	50.000
2122	JPF S.L.	30.000

## Vistas agrupadas

- Una vista puede contener consultas agrupadas (**GROUP BY**). Ejemplo:

```
CREATE VIEW ped_por_rep (quien, cuantos, total, inf,  
sup, medio) AS SELECT rep, COUNT(*), SUM(importe),  
MIN(importe), MAX(importe), AVG(importe) FROM  
pedidos GROUP BY rep
```

QUIEN	CUANTOS	TOTAL	INF	SUP	MEDIO
106	2	32.958	1.458	31.500	16.479,00
105	5	39.327	702	27.500	7.865,40
108	7	58.633	652	45.000	8.376,14
101	3	26.628	150	22.500	8.876,00
110	2	23.132	632	22.500	11.566,00
109	2	7.105	1.480	5.625	3.552,50
107	3	34.432	652	31.350	11.477,33
102	4	22.776	1.896	15.000	5.694,00
103	2	2.700	600	2.100	1.350,00

## Vistas compuestas (I)

- Permiten agrupar en una tabla el resultado de una consulta multitabla.
- Extraen sus datos de dos o más tablas.
- Simplifican posteriores consultas de esos datos, al estar agrupados en una sola “tabla”, la vista.

Ejemplo:

```
CREATE VIEW info_pedido (num_pedido, empresa,  
nombre_rep, importe) AS SELECT num_pedido, empresa,  
nombre, importe FROM pedidos, clientes, repventas  
WHERE clie=num_clie AND rep=num_emp1
```

# Vistas compuestas (II)

NUM PEDIDO	EMPRESA	NOMBRE REP	IMPORTE
112961	Hnos. Ramon S.L.	Jose Maldonado	31.500
113012	EVBE S.A.	Belen Aguirre	3.745
112989	Exclusivas Soriano S.A.	Jose Maldonado	1.458
113051	Exclusivas Norte S.A.	Lorenzo Fernandez	1.420
112968	Exclusivas del Este S.L.	Daniel Gutierrez	3.978
110036	Distribuciones Sur S.A.	Antonio Valle	22.500
113045	Lopez Asociados S.L.	Lorenzo Fernandez	45.000
112963	Pino S.L.	Belen Aguirre	3.276
113013	Exclusivas Norte S.A.	Lorenzo Fernandez	652
113058	Zapater Importaciones S.A.	Maria Garcia	1.480
112997	Domingo S.L.	Natalia Martin	652
112983	Pino S.L.	Belen Aguirre	702
113024	Componentes Fernandez S.A.	Lorenzo Fernandez	7.100
113062	Domingo S.L.	Natalia Martin	2.430
112979	Componentes Fernandez S.A.	Soledad Martinez	15.000
113027	Pino S.L.	Belen Aguirre	4.104
113007	Lopez Asociados S.L.	Lorenzo Fernandez	2.825
113069	Roda & Castedo S.L.	Natalia Martin	31.350
113034	Distribuciones Sur S.A.	Antonio Valle	632
112992	Exclusivas Norte S.A.	Lorenzo Fernandez	760
112975	EVBE S.A.	Pedro Cruz	2.100
113055	Zapater Importaciones S.A.	Daniel Gutierrez	150
113048	Distribuciones Montiel S.L.	Soledad Martinez	3.750
112993	Construcciones Leon S.A.	Soledad Martinez	1.896
113065	Construcciones Leon S.A.	Soledad Martinez	2.130
113003	Zapater Importaciones S.A.	Maria Garcia	5.625
113049	Exclusivas Norte S.A.	Lorenzo Fernandez	776
112987	Pino S.L.	Belen Aguirre	27.500
113057	EVBE S.A.	Pedro Cruz	600
113042	Importaciones Martin S.L.	Daniel Gutierrez	22.500

## Vistas compuestas (III)

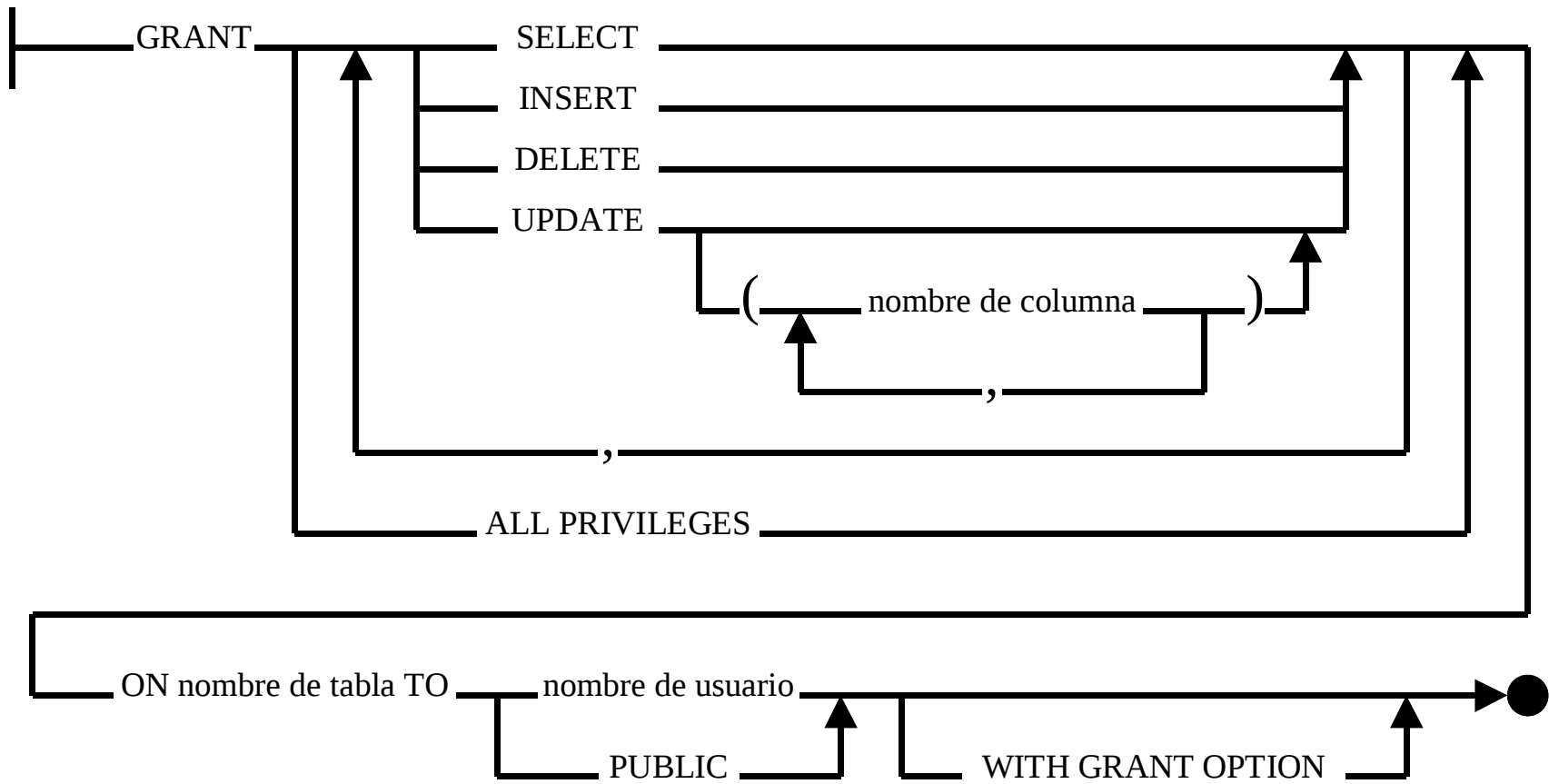
Una consulta sobre la “tabla” info\_pedido creada por la vista es tan sencilla como:

```
SELECT empresa, importe, nombre_rep FROM info_pedido  
WHERE importe>20000 ORDER BY importe DESC
```

EMPRESA	IMPORTE	NOMBRE_REP
Lopez Asociados S.L.	45.000	Lorenzo Fernandez
Hnos. Ramon S.L.	31.500	Jose Maldonado
Roda & Castedo S.L.	31.350	Natalia Martin
Pino S.L.	27.500	Belen Aguirre
Distribuciones Sur S.A.	22.500	Antonio Valle
Importaciones Martin S.L.	22.500	Daniel Gutierrez

# Concesión de privilegios (I)

Se realiza mediante la sentencia **GRANT**.



## Concesión de privilegios (II)

- Un usuario solo puede conceder un privilegio si:
  - Posee el privilegio.
  - Esta autorizado a pasar ese privilegio a otro usuario.
- El propietario de una tabla siempre tiene todos los privilegios sobre la tabla.
- Ejemplos:

**GRANT SELECT, INSERT, UPDATE, DELETE ON pedidos TO  
usuarioA**

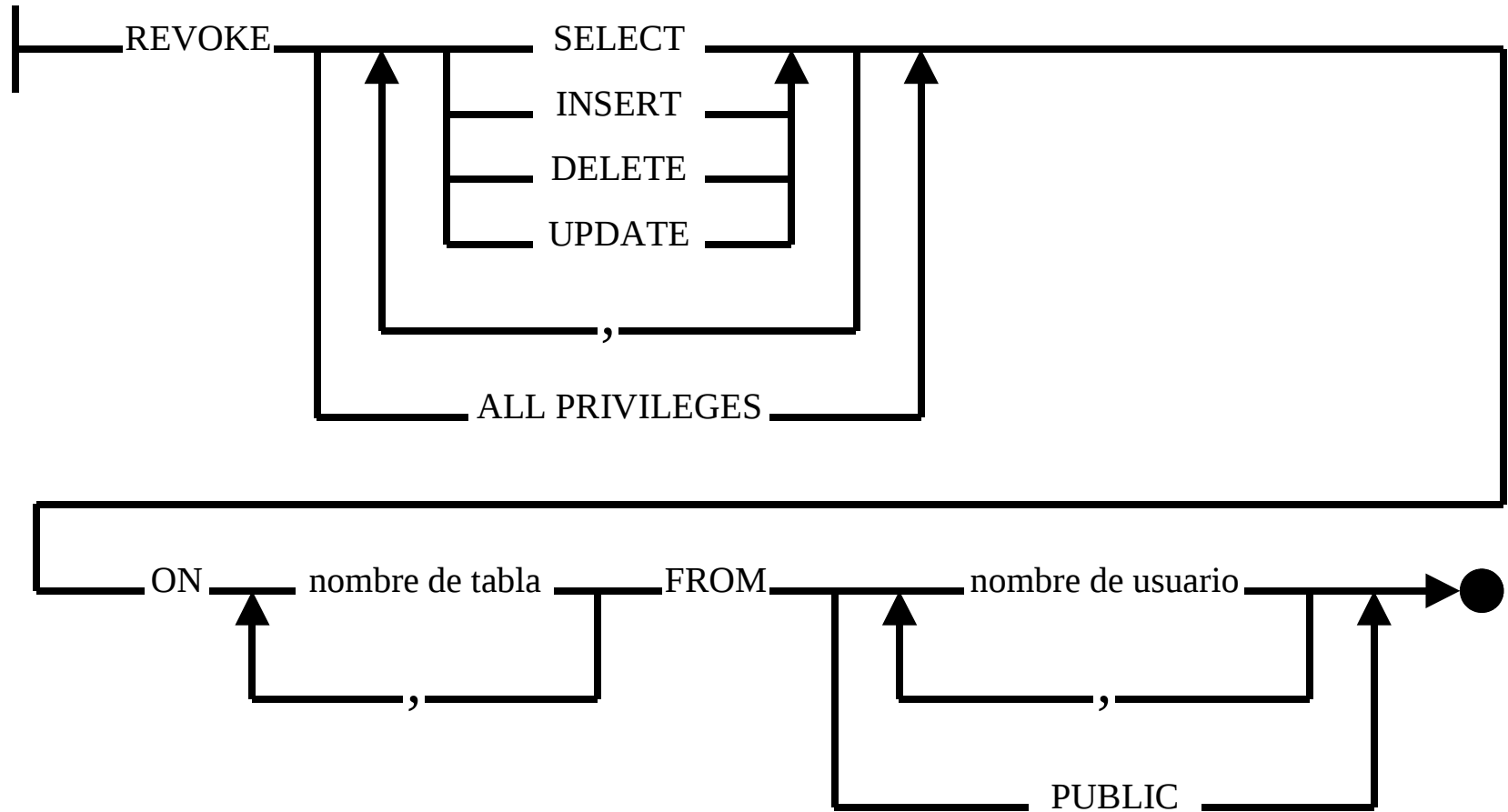
**GRANT SELECT ON clientes TO PUBLIC**

**GRANT SELECT ON oficinas TO usuarioA WITH GRANT OPTION**



# Revocación de privilegios (I)

Se realiza mediante la sentencia **REVOKE**.



## Revocación de privilegios (II)

- Un usuario solo puede revocar los permisos que él haya concedido.
- Ejemplos:

```
REVOKE SELECT, UPDATE ON pedidos FROM usuarioA
```

```
REVOKE ALL PRIVILEGES ON oficinas FROM usuarioA
```

```
REVOKE ALL PRIVILEGES ON oficinas FROM PUBLIC
```