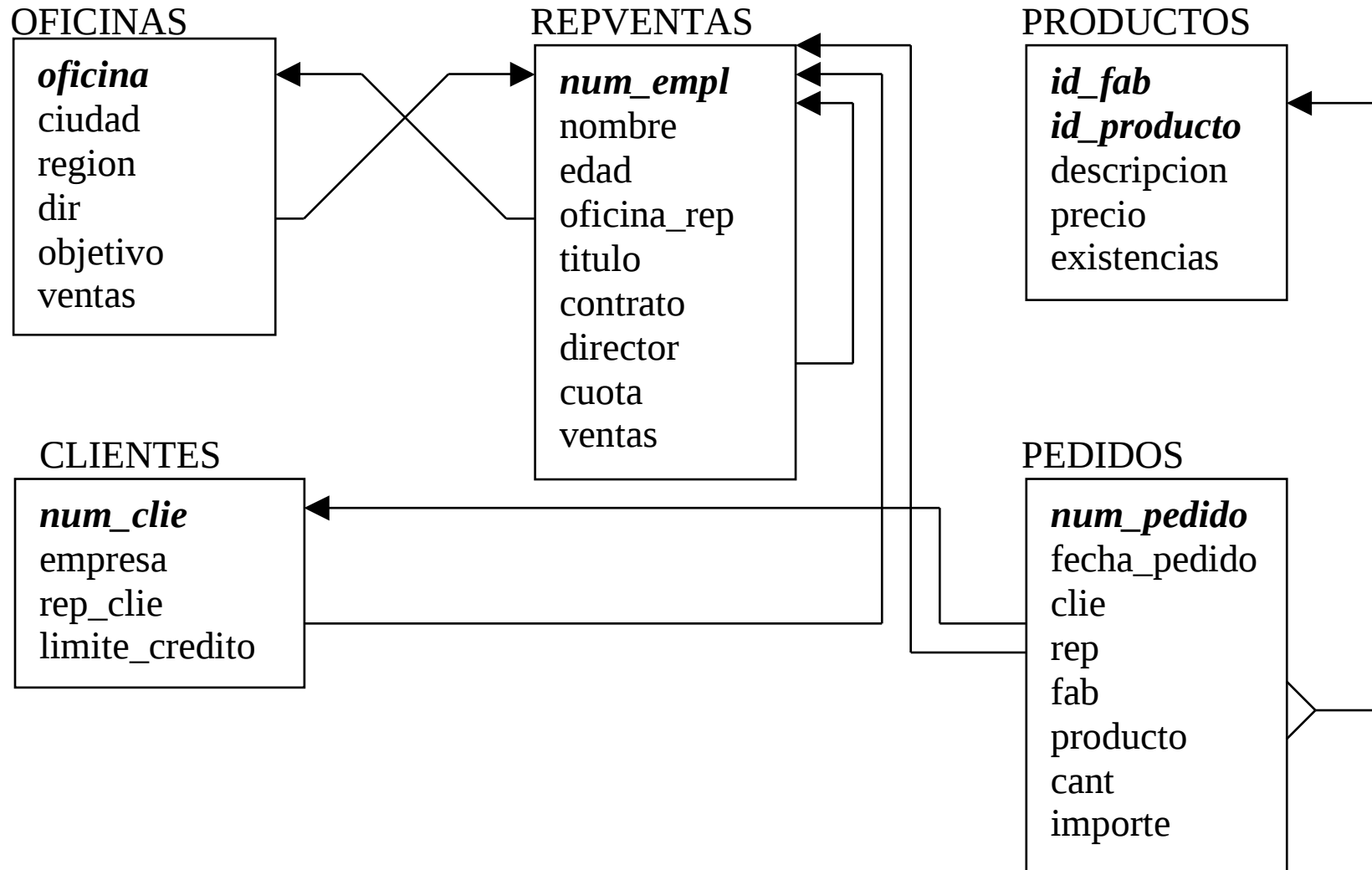


Esquema de la base de datos de ejemplo



Base de datos de ejemplo (I)

OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
22	Toledo	Centro	108	27.500	34.432
11	Valencia	Este	106	52.500	40.063
12	Barcelona	Este	104	70.000	29.328
13	Alicante	Este	105	30.000	39.327
21	Madrid	Centro	108	60.000	81.309

Tabla OFICINAS

Base de datos de ejemplo (II)

<i>NUM_CLIE</i>	<i>EMPRESA</i>	<i>REP_CLIE</i>	<i>LIMITE_CREDITO</i>
2111	EVBE S.A.	103	50.000
2102	Exclusivas del Este S.L.	101	65.000
2103	Pino S.L.	105	50.000
2123	Hnos. Martinez S.A.	102	40.000
2107	Distribuciones Sur S.A.	110	35.000
2115	AFS S.A.	101	20.000
2101	Exclusivas Soriano S.A.	106	65.000
2112	Lopez Asociados S.L.	108	50.000
2121	Hernandez & hijos S.L.	103	45.000
2114	Componentes Fernandez S.A.	102	20.000
2124	Domingo S.L.	107	40.000
2108	Zapater Importaciones S.A.	109	55.000
2117	Hnos. Ramon S.L.	106	35.000
2122	JPF S.L.	105	30.000
2120	Distribuciones Montiel S.L.	102	50.000
2106	Construcciones Leon S.A.	102	65.000
2119	Martinez & Garcia S.L.	109	25.000
2118	Exclusivas Norte S.A.	108	60.000
2113	Importaciones Martin S.L.	104	20.000
2109	Roda & Castedo S.L.	103	25.000
2105	MALB S.A.	101	45.000

Tabla CLIENTES

Base de datos de ejemplo (III)

<i>NUM_EMPL</i>	<i>NOMBRE</i>	<i>EDAD</i>	<i>OFICINA_REP</i>	<i>TITULO</i>	<i>CONTRATO</i>
106	Jose Maldonado	52	11	VP Ventas	14/06/1998
104	Carlos Martinez	33	12	Dir. Ventas	19/05/1997
105	Belen Aguirre	37	13	Dir. Ventas	12/02/1998
109	Maria Garcia	31	11	Rep. Ventas	12/10/1999
108	Lorenzo Fernandez	62	21	Dir. Ventas	12/10/1999
102	Soledad Martinez	48	21	Rep. Ventas	10/12/1996
101	Daniel Gutierrez	45	12	Rep. Ventas	20/10/1996
110	Antonio Valle	41	NULL	Rep. Ventas	13/01/2000
103	Pedro Cruz	29	12	Rep. Ventas	01/03/1997
107	Natalia Martin	49	22	Rep. Ventas	14/11/1998
<i>DIRECTOR</i>	<i>CUOTA</i>	<i>VENTAS</i>			
NULL	25.000	32.958			
106	17.500	0			
104	30.000	39.327			
106	27.500	7.105			
106	30.000	58.533			
108	30.000	22.776			
104	27.500	26.628			
101	NULL	23.123			
104	25.000	2.700			
108	27.500	34.432			

Tabla REPVENTAS

Base de datos de ejemplo (IV)

<i>ID_FAB</i>	<i>ID_PRODUCTO</i>	<i>DESCRIPCION</i>	<i>PRECIO</i>	<i>EXISTENCIAS</i>
REI	2A45C	V Stago Trinquete	79	210
ACI	4100Y	Extractor	2.750	25
QSA	XK47	Reductor	355	38
BIC	41672	Plate	180	0
IMM	779C	Riostra 2-Tm	1.875	9
ACI	41003	Artículo Tipo 3	107	207
ACI	41004	Artículo Tipo 4	117	139
BIC	41003	Manivela	652	3
IMM	887P	Perno Riostra	250	24
QSA	XK48	Reductor	134	203
REI	2A44L	Bisagra Izqda.	4.500	12
FEA	112	Cubierta	148	115
IMM	887H	Soporte Riostra	54	223
BIC	41089	Reten	225	78
ACI	41001	Artículo Tipo 1	55	277
IMM	775C	Riostra 1-Tm	1.425	5
ACI	4100Z	Montador	2.500	28
QSA	XK48A	Reductor	117	37
ACI	41002	Artículo Tipo 2	76	167
REI	2A44R	Bisagra Dcha.	4.500	12
IMM	773C	Riostra 1/2-Tm	975	28
ACI	4100X	Ajustador	25	37
FEA	114	Bancada Motor	243	15
IMM	887X	Retenedor Riostra	475	32
REI	2A44G	Pasador Bisagra	350	14

Tabla PRODUCTOS

Base de datos de ejemplo (V)

<i>NUM_PEDIDO</i>	<i>FECHA_PEDIDO</i>	<i>CLIE</i>	<i>REP</i>	<i>FAB</i>	<i>PRODUCTO</i>	<i>CANT</i>	<i>IMPORTE</i>
112961	17/12/1999	2117	106	REI	2A44L	7	31.500
113012	11/01/2000	2111	105	ACI	41003	35	3.745
112989	03/01/2000	2101	106	FEA	114	6	1.458
113051	10/02/2000	2118	108	QSA	XK47	4	1.420
112968	12/10/1999	2102	101	ACI	41004	34	3.978
110036	30/01/2000	2107	110	ACI	4100Z	9	22.500
113045	02/02/2000	2112	108	REI	2A44R	10	45.000
112963	17/12/1999	2103	105	ACI	41004	28	3.276
113013	14/01/2000	2118	108	BIC	41003	1	652
113058	23/02/2000	2108	109	FEA	112	10	1.480
112997	08/01/2000	2124	107	BIC	41003	1	652
112983	27/12/1999	2103	105	ACI	41004	6	702
113024	20/01/2000	2114	108	QSA	XK47	20	7.100
113062	24/02/2000	2124	107	FEA	114	10	2.430
112979	12/10/1999	2114	102	ACI	4100Z	6	15.000
113027	22/02/2000	2103	105	ACI	41002	54	4.104
113007	08/01/2000	2112	108	IMM	773C	3	2.825
113069	02/03/2000	2109	107	IMM	775C	22	31.350
113034	29/01/2000	2107	110	REI	2A45C	8	632
112992	04/11/1999	2118	108	ACI	41002	10	760
112975	12/10/1999	2111	103	REI	2A44G	6	2.100
113055	15/02/2000	2108	101	ACI	4100X	6	150
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
113003	25/01/2000	2108	109	IMM	779C	3	5.625
113049	10/02/2000	2118	108	QSA	XK47	2	776
112987	31/12/1999	2103	105	ACI	4100Y	11	27.500
113057	18/02/2000	2111	103	ACI	4100X	24	600
113042	02/02/2000	2113	101	REI	2A44R	5	22.500

Tabla PEDIDOS

Tipos de datos (I)

- Numéricos.
 - Enteros.
 - De 2 bytes [-32768, 32767], **SMALLINT**
 - De 4 bytes [-2147483648, 2147483647], **INTEGER**
 - Decimales.
 - Números de **p** dígitos de los que **s** son decimales, **DECIMAL(p,s)**
 - Coma flotante.
 - Números reales de 4 bytes [-3.402823466e38, -1.175494351e-38], 0, [1.175494351e-38, 3.402823466e38], **FLOAT** ó **REAL**

Tipos de datos (II)

- Alfanuméricos (cadenas de caracteres).
 - De longitud fija.
 - Tamaño N con $1 \leq N \leq 254$.
 - Deben tener siempre N caracteres, **CHAR(N)**
 - De longitud variable.
 - Tamaño N con $1 \leq N \leq 32767$.
 - Pueden tener entre 0 y N caracteres, **VARCHAR(N)**

Tipos de datos (III)

- Datos de tiempo.
 - Fecha.
 - Día del calendario Gregoriano.
 - Valor entre 01/01/0001 y 31/12/9999, **DATE**
 - Hora.
 - Hora con precisión de segundos.
 - Valor entre 00/00/00 y 23/59/59 ó 00/00/01 y 24/00/00, **TIME**
 - Instante.
 - Instante temporal (fecha, hora y microsegundo).
 - Rango de fecha, hora y 6 decimales para microsegundo, **TIMESTAMP**

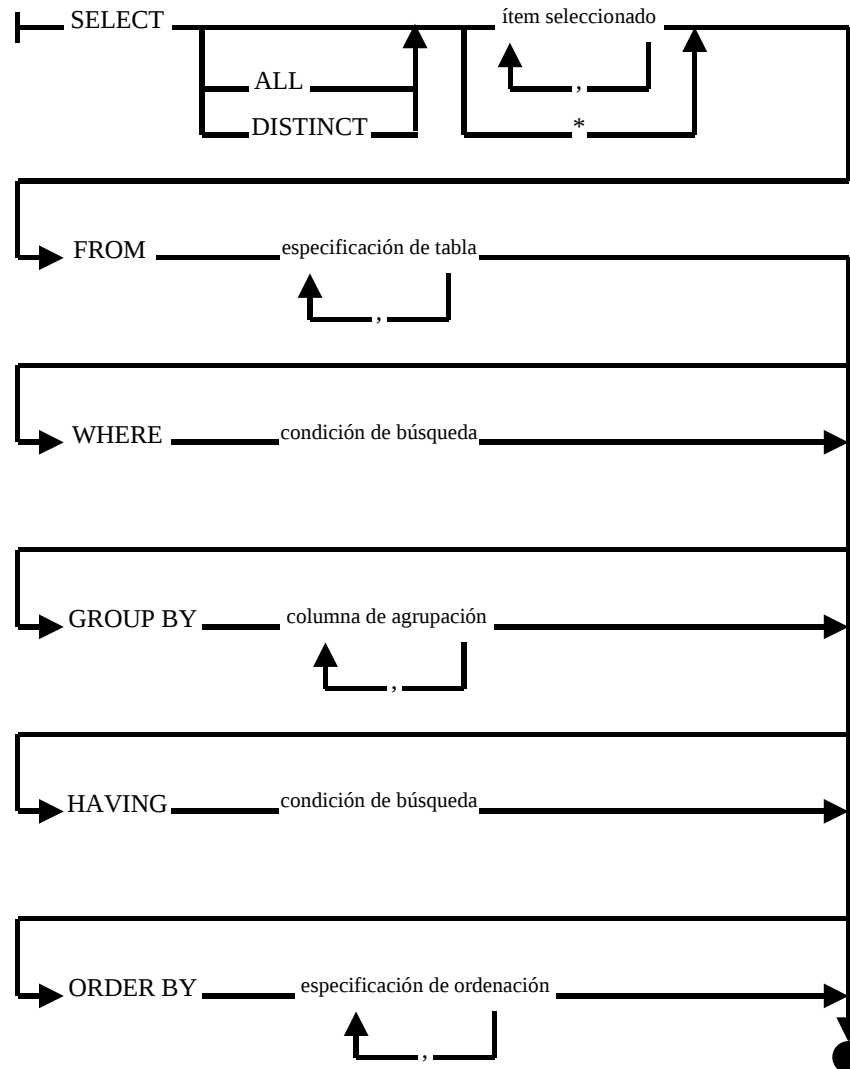
Constantes

- Numéricas.
 - Enteros: -5, 12
 - Coma flotante: 2.16, -3.05, 4.2e-3
- Alfanuméricas.
 - Caracteres limitados por ': 'hola', '34', 'pepe3'
- De tiempo.
 - Debe tenerse en cuenta la representación de los datos dd/mm/aaaa, mm/dd/aaaa, etc.
 - 03/10/2004, 10/29/58

Valor nulo

- El valor nulo se representa, para todo tipo de datos, mediante la palabra **NULL**
- Indica un dato desconocido o no aplicable.
- El valor nulo es distinto de:
 - El valor cero para datos numéricos.
 - La cadena vacía para datos alfanuméricos.
 - El primer valor de fecha, hora, etc., para datos de tiempo.

Diagrama sintáctico de la sentencia SELECT



Consultas sencillas (I)

- Utilización de una sola tabla en la consulta.
- Ninguna condición de selección de filas.

<i>OFICINA</i>	<i>CIUDAD</i>	<i>REGION</i>	<i>DIR</i>	<i>OBJETIVO</i>	<i>VENTAS</i>
22	Toledo	Centro	108	27.500	34.432
11	Valencia	Este	106	52.500	40.063
12	Barcelona	Este	104	70.000	29.328
13	Alicante	Este	105	30.000	39.327
21	Madrid	Centro	108	60.000	81.309

SELECT ciudad, region, ventas FROM oficinas

<i>CIUDAD</i>	<i>REGION</i>	<i>VENTAS</i>
Toledo	Centro	34.432
Valencia	Este	40.063
Barcelona	Este	29.328
Alicante	Este	39.327
Madrid	Centro	81.309

Consultas sencillas (II)

SELECT ciudad, region, (ventas-objetivo) FROM oficinas

<i>CIUDAD</i>	<i>REGION</i>	<i>VENTAS-OBJETIVO</i>
Toledo	Centro	6.932
Valencia	Este	-12.437
Barcelona	Este	-40.672
Alicante	Este	9.327
Madrid	Centro	21.309

SELECT ciudad, 'tiene ventas de', ventas FROM oficinas

<i>CIUDAD</i>	<i>'tiene ventas de'</i>	<i>VENTAS</i>
Toledo	tiene ventas de	34.432
Valencia	tiene ventas de	40.063
Barcelona	tiene ventas de	29.328
Alicante	tiene ventas de	39.327
Madrid	tiene ventas de	81.309

Consultas sencillas (III)

SELECT * FROM oficinas

<i>OFICINA</i>	<i>CIUDAD</i>	<i>REGION</i>	<i>DIR</i>	<i>OBJETIVO</i>	<i>VENTAS</i>
22	Toledo	Centro	108	27.500	34.432
11	Valencia	Este	106	52.500	40.063
12	Barcelona	Este	104	70.000	29.328
13	Alicante	Este	105	30.000	39.327
21	Madrid	Centro	108	60.000	81.309

**SELECT nombre, DAY(contrato), MONTH(contrato),
YEAR(contrato) FROM repventas**

<i>NOMBRE</i>	<i>DAY(contrato)</i>	<i>MONTH(contrato)</i>	<i>YEAR(contrato)</i>
Jose Maldonado	14	6	1998
Carlos Martinez	19	5	1997
Belen Aguirre	12	2	1998
Maria Garcia	12	10	1999
Lorenzo Fernandez	12	10	1999
Soledad Martinez	10	12	1996
Daniel Gutierrez	20	10	1996
Antonio Valle	13	1	2000
Pedro Cruz	01	3	1997
Natalia Martin	14	11	1998

Eliminación de filas duplicadas

- En una consulta aparecen, por defecto, todas las filas resultantes.
- Para eliminar las filas de respuesta que tienen todos sus datos iguales debe utilizarse la palabra **DISTINCT**

```
SELECT DISTINCT region FROM oficinas
```

<i>REGION</i>
Este
Centro

Selección de filas

- Permiten seleccionar un conjunto de filas en la respuesta a una consulta.
- Cinco tipos de test de selección de filas:
 - Test de comparación.
 - Test de rango.
 - Test de pertenencia a conjunto.
 - Test de correspondencia con patrón.
 - Test de valor nulo.

Test de comparación (I)

- Compara dos valores.
- Seis condiciones de comparación: =, <>, <, <=, >, >=

```
SELECT nombre FROM repventas WHERE contrato <=  
'01/01/1998'
```

<i>NOMBRE</i>
Carlos Martinez
Soledad Martinez
Daniel Gutierrez
Pedro Cruz

```
SELECT ciudad, ventas, objetivo FROM oficinas WHERE  
ventas <= (0.75*objetivo)
```

<i>CIUDAD</i>	<i>VENTAS</i>	<i>OBJETIVO</i>
Barcelona	29.328	70.000

Test de comparación (II)

- Valor nulo en la condición de comparación.

<i>NOMBRE</i>	<i>CUOTA</i>	<i>VENTAS</i>
Jose Maldonado	25.000	32.958
Carlos Martinez	17.500	0
Belen Aguirre	30.000	39.327
Maria Garcia	27.500	7.105
Lorenzo Fernandez	30.000	58.533
Soledad Martinez	30.000	22.776
Daniel Gutierrez	27.500	26.628
Antonio Valle	NULL	23.123
Pedro Cruz	25.000	2.700
Natalia Martin	27.500	34.432

SELECT nombre FROM repventas WHERE ventas <= cuota

<i>NOMBRE</i>
Carlos Martinez
Maria Garcia
Soledad Martinez
Daniel Gutierrez
Pedro Cruz

Test de rango

- Comprueba si:
 - El valor esta en un rango **BETWEEN**
 - El valor esta fuera de un rango **NOT BETWEEN**

**SELECT num_pedido, importe FROM pedidos WHERE importe
BETWEEN 20000 AND 27500**

<i>NUM_PEDIDO</i>	<i>IMPORTE</i>
110036	22.500
112987	27.500
113042	22.500

**SELECT nombre, ventas, cuota FROM repventas WHERE
ventas NOT BETWEEN (0.75*cuota) AND (1.5*cuota)**

<i>NOMBRE</i>	<i>VENTAS</i>	<i>CUOTA</i>
Carlos Martinez	0	17.500
Maria Garcia	7.105	27.500
Lorenzo Fernadez	58.533	30.000
Pedro Cruz	2.700	25.000

Test de pertenencia a conjunto

- Examina si un valor se encuentra en un conjunto de valores **IN**

```
SELECT nombre, cuota, ventas FROM repventas WHERE  
oficina_rep IN (11, 13, 22)
```

<i>NOMBRE</i>	<i>CUOTA</i>	<i>VENTAS</i>
Jose Maldonado	25.000	32.958
Belen Aguirre	30.000	39.327
Maria Garcia	27.500	7.105
Natalia Martin	27.500	34.432

Test de correspondencia con patrón

- Comprueba si una cadena alfanumérica se corresponde con un determinado patrón **LIKE**
- El patrón se especifica mediante los caracteres:
 - % Indica 0 o más caracteres.
 - _ Indica 1 y solo 1 carácter.
 - Si el la cadena contiene los caracteres % ó _, se deben preceder del carácter \$ para indicar que no son patrones, sino un caracteres literales.

```
SELECT num_empl, nombre FROM repventas WHERE nombre  
      LIKE '% Martinez'
```

<i>NUM_EMPL</i>	<i>NOMBRE</i>
104	Carlos Martinez
102	Soledad Martinez

Test de valor nulo

- Las comparaciones = NULL o <> NULL no son validas.
- Para comprobar si un valor es:
 - Nulo **IS NULL**
 - No nulo **IS NOT NULL**

SELECT nombre FROM repventas WHERE oficina_rep IS NULL

<i>NOMBRE</i>
Antonio Valle

SELECT nombre FROM repventas WHERE oficina_rep IS NOT NULL

Devuelve el resto de elementos de repventas.

Condiciones de búsqueda compuestas

- Se construyen a partir de los test anteriores utilizando **AND**, **OR** ó **NOT**
- Se usa la lógica trivaluada para devolver su resultado (TRUE, FALSE ó NULL).

```
SELECT nombre, cuota, ventas FROM repventas WHERE  
ventas<cuota AND ventas<20000
```

<i>NOMBRE</i>	<i>CUOTA</i>	<i>VENTAS</i>
Carlos Martinez	17.500	0
Maria Garcia	27.500	7.105
Pedro Cruz	25.000	2.700

Ordenación de los resultados (I)

- Se realiza con **ORDER BY**
- La ordenación se especifica mediante:
 - El nombre de los campos
 - Su posición en los datos de salida.

```
SELECT ciudad, region, ventas FROM oficinas ORDER BY  
region, ciudad
```

<i>CIUDAD</i>	<i>REGION</i>	<i>VENTAS</i>
Madrid	Centro	81.309
Toledo	Centro	34.432
Alicante	Este	39.327
Barcelona	Este	29.328
Valencia	Este	40.063

Ordenación de los resultados (II)

- La palabra **DESC** permite invertir la ordenación y realizarla de mayor a menor.

```
SELECT ciudad, region, (ventas-objetivo) FROM oficinas  
ORDER BY 3 DESC
```

<i>CIUDAD</i>	<i>REGION</i>	<i>VENTAS-OBJETIVO</i>
Madrid	Centro	21.309
Alicante	E ste	9.327
Toledo	Centro	6.932
Valencia	E ste	-12.437
Barcelona	E ste	-40.672

Combinación de los resultados (I)

- Posibilidad de unir los resultados de varias consultas en una sola respuesta, **UNION**
- Las filas duplicadas solo aparecen una vez, excepto que se utilice **UNION ALL**
- Restricciones:
 - Las consultas deben contener el mismo número de columnas.
 - Los tipos de datos de cada columna deben ser iguales.
 - Ninguna de las consultas puede estar ordenada, pero puede ordenarse la respuesta final.

Combinación de los resultados (II)

SELECT id_fab, id_producto FROM productos WHERE precio > 2000

<i>ID_FAB</i>	<i>ID_PRODUCTO</i>
ACI	4100Y
REI	2A44L
ACI	4100Z
REI	2A44R

SELECT DISTINCT fab, producto FROM pedidos WHERE importe > 30000

<i>FAB</i>	<i>PRODUCTO</i>
IMM	775C
REI	2A44L
REI	2A44R

**SELECT id_fab, id_producto FROM productos WHERE precio > 2000 UNION
SELECT DISTINCT fab, producto FROM pedidos WHERE importe > 30000**

<i>ID_FAB</i>	<i>ID_PRODUCTO</i>
ACI	4100Y
ACI	4100Z
IMM	775C
REI	2A44L
REI	2A44R

Composiciones (I)

- Es necesario utilizar datos de más de una tabla para responder la consulta.
- Al especificar más de una tabla:
 - Se realiza el producto cartesiano generalizado de las tablas especificadas.
 - Se ejecutan las condiciones impuestas a la consulta sobre la tabla obtenida.

SELECT ... FROM tablaA, tablaB ...

TABLA A		X	TABLA B		=	TABLA AxB			
Col A	Col B		Col C	Col D		Col A	Col B	Col C	Col D
A1	B1		C1	D1		A1	B1	C1	D1
A2	B2		C2	D2		A1	B1	C2	D2
			C3	D3		A1	B1	C3	D3
						A2	B2	C1	D1
						A2	B2	C2	D2
						A2	B2	C3	D3

Composiciones (II)

**SELECT num_pedido, importe, empresa, limite_credito FROM pedidos,
clientes WHERE clie=num_clie**

<i>NUM_PEDIDO</i>	<i>IMPORTE</i>	<i>EMPRESA</i>	<i>LIMITE_CREDITO</i>
112961	31.500	Hnos. Ramon S.L.	35.000
113012	3.745	EVBE S.A.	50.000
112989	1.458	Exclusivas Soriano S.A.	65.000
113051	1.420	Exclusivas Norte S.A.	60.000
112968	3.978	Exclusivas del Este S.L.	65.000
110036	22.500	Distribuciones Sur S.A.	35.000
113045	45.000	Lopez Asociados S.L.	50.000
112963	3.276	Pino S.L.	50.000
113013	652	Exclusivas Norte S.A.	60.000
113058	1.480	Zapater Importaciones S.A.	55.000
112997	652	Domingo S.L.	40.000
112983	702	Pino S.L.	50.000
113024	7.100	Componentes Fernandez S.A.	20.000
113062	2.430	Domingo S.L.	40.000
112979	15.000	Componentes Fernandez S.A.	20.000
113027	4.104	Pino S.L.	50.000
113007	2.825	Lopez Asociados S.L.	50.000
113069	31.350	Roda & Castedo S.L.	25.000
113034	632	Distribuciones Sur S.A.	35.000
112992	760	Exclusivas Norte S.A.	60.000
112975	2.100	EVBE S.A.	50.000
113055	150	Zapater Importaciones S.A.	55.000
113048	3.750	Distribuciones Montiel S.L.	50.000
112993	1.896	Construcciones Leon S.A.	65.000
113065	2.130	Construcciones Leon S.A.	65.000
113003	5.625	Zapater Importaciones S.A.	55.000
113049	776	Exclusivas Norte S.A.	60.000
112987	27.500	Pino S.L.	50.000
113057	600	EVBE S.A.	50.000
113042	22.500	Importaciones Martin S.L.	20.000

Consultas padre/hijo

- Relacionan tablas mediante claves primaria y foránea.

```
SELECT nombre, ciudad, region FROM repventas, oficinas  
WHERE oficina_rep = oficina
```

<i>NOMBRE</i>	<i>CIUDAD</i>	<i>REGION</i>
Jose Maldonado	Valencia	Este
Carlos Martinez	Barcelona	Este
Belen Aguirre	Alicante	Este
Maria Garcia	Valencia	Este
Lorenzo Fernandez	Madrid	Centro
Soledad Martinez	Madrid	Centro
Daniel Gutierrez	Barcelona	Este
Pedro Cruz	Barcelona	Este
Natalia Martin	Toledo	Centro

Criterios de selección de filas

- Las composiciones pueden incluir condiciones para seleccionar filas en la respuesta.

```
SELECT nombre, ciudad, region FROM repventas, oficinas  
WHERE oficina_rep = oficina AND objetivo >= 60000
```

<i>NOMBRE</i>	<i>CIUDAD</i>	<i>REGION</i>
Carlos Martinez	Barcelona	Este
Lorenzo Fernandez	Madrid	Centro
Soledad Martinez	Madrid	Centro
Daniel Gutierrez	Barcelona	Este
Pedro Cruz	Barcelona	Este

Múltiples columnas de emparejamiento (I)

- Dos tablas se relacionan por dos o más columnas.
- Generalmente en relaciones con claves primarias formadas por varios campos.

```
SELECT num_pedido, importe, descripcion FROM pedidos,  
       productos WHERE fab = id_fab AND producto =  
                        id_producto
```

Múltiples columnas de emparejamiento (II)

<i>NUM_PEDIDO</i>	<i>IMPORTE</i>	<i>DESCRIPCION</i>
112961	31.500	Bisagra Izqda.
113012	3.745	Articulo Tipo 3
112989	1.458	Bancada Motor
113051	1.420	Reductor
112968	3.978	Articulo Tipo 4
110036	22.500	Montador
113045	45.000	Bisagra Dcha.
112963	3.276	Articulo Tipo 4
113013	652	Manivela
113058	1.480	Cubierta
112997	652	Manivela
112983	702	Articulo Tipo 4
113024	7.100	Reductor
113062	2.430	Bancada Motor
112979	15.000	Montador
113027	4.104	Articulo Tipo 2
113007	2.825	Riostra 1/2-Tm
113069	31.350	Riostra 1-Tm
113034	632	V Stago Trinquete
112992	760	Articulo Tipo 2
112975	2.100	Pasador Bisagra
113055	150	Ajustador
113048	3.750	Riostra 2-Tm
112993	1.896	V Stago Trinquete
113065	2.130	Reductor
113003	5.625	Risotra 2-Tm
113049	776	Reductor
112987	27.500	Extractor
113057	600	Ajustador
113042	22.500	Bisagra Dcha.

Consultas de tres o más tablas (I)

- Es posible realizar consultas que impliquen cualquier número de tablas.
- Las tablas se combinan de forma similar a la composición de dos tablas.

```
SELECT num_pedido, importe, empresa, nombre FROM  
pedidos, clientes, repventas WHERE clie = num_clie  
AND rep = num_empl AND importe > 25000
```

<i>NUM_PEDIDO</i>	<i>IMPORTE</i>	<i>EMPRESA</i>	<i>NOMBRE</i>
112961	31.500	Hnos. Ramon S.L.	Jose Maldonado
113045	45.000	Lopez Asociados S.L.	Lorenzo Fernandez
113069	31.350	Roda & Castedo S.L.	Natalia Martin
112987	27.500	Pino S.L.	Belen Aguirre

Consultas de tres o más tablas (II)

```
SELECT num_pedido, importe, empresa, nombre, ciudad  
FROM pedidos, clientes, repventas, oficinas WHERE  
clie = num_clie AND rep_clie = num_empl AND  
oficina_rep = oficina AND importe > 25000
```

<i>NUM_PEDIDO</i>	<i>IMPORTE</i>	<i>EMPRESA</i>	<i>NOMBRE</i>	<i>CIUDAD</i>
112961	31.500	Hnos. Ramon S.L.	Jose Maldonado	Valencia
113045	45.000	Lopez Asociados S.L.	Lorenzo Fernandez	Madrid
113069	31.350	Roda & Castedo S.L.	Pedro Cruz	Barcelona
112987	27.500	Pino S.L.	Belen Aguirre	Alicante

Otras equicomposiciones (I)

- Las composiciones no tienen porque implicar relaciones padre/hijo.
- Es posible componer mediante:
 - Cualquiera dos columnas de tipo de datos igual.
 - Desigualdades o cualquier otro tipo de relación

Otras equicomposiciones (II)

SELECT num_pedido, importe, fecha_pedido, nombre FROM pedidos, repventas WHERE fecha_pedido = contrato

<i>NUM_PEDIDO</i>	<i>IMPORTE</i>	<i>FECHA_PEDIDO</i>	<i>NOMBRE</i>
112968	3.978	12/10/1999	Maria Garcia
112979	15.000	12/10/1999	Maria Garcia
112975	2.100	12/10/1999	Maria Garcia
112968	3.978	12/10/1999	Lorenzo Fernandez
112979	15.000	12/10/1999	Lorenzo Fernandez
112975	2.100	12/10/1999	Lorenzo Fernandez

SELECT nombre, cuota, ciudad, objetivo FROM repventas, oficinas WHERE oficina_rep = oficina AND cuota > (0.5*objetivo)

<i>NOMBRE</i>	<i>CUOTA</i>	<i>CIUDAD</i>	<i>OBJETIVO</i>
Belen Aguirre	30.000	Alicante	30.000
Maria Garcia	27.500	Valencia	52.500
Natalia Martin	27.500	Toledo	27.500

Nombres de columna cualificados (I)

- Las tablas de la base de datos pueden tener columnas con el mismo nombre:
 - Columna ventas de la tabla oficinas.
 - Columna ventas de la tabla repventas.
- Deseamos ejecutar la consulta:

```
SELECT nombre, ventas, ciudad FROM repventas, oficinas  
WHERE oficina_rep = oficina
```

Resultado:

Error: Columna 'ventas' se encuentra en más de una tabla

Solución: Nombres de columna cualificados.

Nombres de columna cualificados (II)

- Un nombre de columna cualificado se forma mediante:
 - El nombre de la tabla.
 - El carácter . (punto).
 - El nombre de la columna.

SELECT nombre, repventas.ventas, ciudad FROM repventas, oficinas WHERE oficina_rep = oficina

<i>NOMBRE</i>	<i>VENTAS</i>	<i>CIUDAD</i>
Jose Maldonado	32.958	Valencia
Carlos Martinez	0	Barcelona
Belen Aguirre	39.327	Alicante
Maria Garcia	7.105	Valencia
Lorenzo Fernandez	58.533	Madrid
Soledad Martinez	22.776	Madrid
Daniel Gutierrez	26.628	Barcelona
Pedro Cruz	2.700	Barcelona
Natalia Martin	34.432	Toledo

Autocomposiciones (I)

- Necesidad de componer una tabla consigo misma.
- En nuestro ejemplo, queremos obtener el nombre del empleado y el de su director.

```
SELECT nombre, nombre FROM repventas WHERE director =  
num_emp1
```

- ¡ Ninguna respuesta !.
- No podemos poner dos veces la tabla repventas en FROM. ¿A que tabla pertenece nombre?.
- Necesidad de poder renombrar las tablas.

Autocomposiciones (II)

- Renombrar tablas mediante:
 - <nombre de tabla> AS <nombre nuevo>
 - AS es opcional y puede no ponerse.

**SELECT repventas.nombre, dirs.nombre FROM repventas,
repventas dirs WHERE repventas.director =
dirs.num_empl**

<i>NOMBRE</i>	<i>NOMBRE</i>
Carlos Martinez	Jose Maldonado
Belen Aguirre	Carlos Martinez
Maria Garcia	Jose Maldonado
Lorenzo Fernandez	Jose Maldonado
Soledad Martinez	Lorenzo Fernandez
Daniel Gutierrez	Carlos Martinez
Antonio Valle	Daniel Gutierrez
Pedro Cruz	Carlos Martinez
Natalia Martin	Lorenzo Fernandez

Consultas sumarias (I)

- Realizan cálculos sobre los resultados de una consulta.
- Operan sobre los valores de columnas.
- Cinco operaciones:
 - Calculo de la suma de una columna: **SUM**
 - Calculo de la media de una columna: **AVG**
 - Calculo del máximo de una columna: **MAX**
 - Calculo del mínimo de una columna: **MIN**
 - Calculo del número de filas de una columna: **COUNT**

Consultas sumarias (II)

- SUM y AVG operan solo sobre tipos de datos numéricos, devolviendo un dato de tipo numérico.
- MAX y MIN operan sobre cualquier tipo de dato, devolviendo un dato del tipo sobre el que operan.
- COUNT opera sobre cualquier tipo de dato, devolviendo un dato de tipo entero: El número de filas de la columna.

Consultas sumarias (III)

SELECT SUM(cuota), SUM(ventas) FROM repventas

<i>SUM(CUOTA)</i>	<i>SUM(VENTAS)</i>
240.000	247.582

**SELECT SUM(importe) FROM pedidos, repventas WHERE
nombre = 'Belen Aguirre' AND rep = num_empl**

<i>SUM(IMPORTE)</i>
39.327

SELECT AVG(precio) FROM productos WHERE id_fab = 'ACI'

<i>AVG(PRECIO)</i>
804,29

SELECT AVG(importe) FROM pedidos WHERE clie = 2103

<i>AVG(IMPORTE)</i>
8.895,50

Consultas sumarias (IV)

SELECT MIN(cuota), MAX(cuota) FROM repventas

<i>MIN(CUOTA)</i>	<i>MAX(CUOTA)</i>
17.500	30.000

SELECT MIN(fecha_pedido) FROM pedidos

<i>MIN(FECHA_PEDIDO)</i>
12/10/1999

SELECT MAX(100*ventas/cuota) FROM repventas

<i>MAX(100*VENTAS/CUOTA)</i>
195

SELECT MIN(nombre), MAX(nombre) FROM repventas

<i>MIN(NOMBRE)</i>	<i>MAX(NOMBRE)</i>
Antonio Valle	Soledad Martinez

Consultas sumarias (V)

SELECT COUNT(num_clie) FROM clientes

<i>COUNT(NUM_CLIE)</i>
21

SELECT COUNT(nombre) FROM repventas WHERE ventas > cuota

<i>COUNT(NOMBRE)</i>
4

SELECT COUNT(importe) FROM pedidos WHERE importe > 25000

<i>COUNT(IMPORTE)</i>
4

SELECT COUNT(*) FROM pedidos WHERE importe > 25000

<i>COUNT(*)</i>
4

Consultas sumarias y valores nulos

- Si un valor de la consulta sumaria es nulo, dicho valor es ignorado por la consulta sumaria.

```
SELECT COUNT(*), COUNT(ventas), COUNT(cuota) FROM  
repventas
```

<i>COUNT(*)</i>	<i>COUNT(VENTAS)</i>	<i>COUNT(CUOTA)</i>
10	10	9

```
SELECT SUM(ventas), SUM(cuota), SUM(ventas)-  
SUM(cuota), SUM(ventas-cuota) FROM repventas
```

<i>SUM(VENTAS)</i>	<i>SUM(CUOTA)</i>	<i>SUM(VENTAS)-SUM(CUOTA)</i>	<i>SUM(VENTAS-CUOTA)</i>
247.582	240.000	7.582	-15.541

Eliminación de filas duplicadas

- Es posible eliminar filas duplicadas mediante la palabra **DISTINCT**
- Solo puede usarse con las funciones COUNT, SUM y AVG.

```
SELECT COUNT(DISTINCT titulo) FROM repventas
```

<i>COUNT(DISTINCT TITULO)</i>
3

```
SELECT COUNT(DISTINCT oficina_rep) FROM repventas  
WHERE ventas > cuota
```

<i>COUNT(DISTINCT OFICINA_REP)</i>
4

Consultas sumarias agrupadas (I)

- Permite obtener consultas sumarias sobre conjuntos de datos y no sobre el total de los datos.
- Para ello se utiliza la sentencia **GROUP BY**
- **GROUP BY:**
 - Divide los datos en conjuntos en función de los campos de agrupación que se indiquen.
 - Calcula la consulta sumaria de cada uno de los conjuntos obtenidos.

Consultas sumarias agrupadas (II)

SELECT rep, AVG(importe) FROM pedidos GROUP BY rep

<i>REP</i>	<i>AVG(IMPORTE)</i>
101	8.876,00
102	5.694,00
103	1.350,00
105	7.865,40
106	16.479,00
107	11.477,33
108	8.361,86
109	3.552,50
110	11.566,00

SELECT oficina_rep, MIN(cuota), MAX(cuota) FROM repventas GROUP BY oficina_rep

<i>OFICINA_REP</i>	<i>MIN(CUOTA)</i>	<i>MAX(CUOTA)</i>
NULL	NULL	NULL
11	25.000	27.500
12	17.500	27.500
13	30.000	30.000
21	30.000	30.000
22	27.500	27.500

Consultas sumarias agrupadas (III)

- SQL ignora si una clave es primaria al agrupar:

```
SELECT num_empl, nombre, SUM(importe) FROM pedidos,  
       repventas WHERE rep = num_empl GROUP BY num_empl
```

Resultado:

Error: La columna 'nombre' no es una expresión GROUP BY

Solución: Agrupar siempre por todos los elementos.

```
SELECT num_empl, nombre, SUM(importe) FROM pedidos,  
       repventas WHERE rep = num_empl GROUP BY num_empl, nombre
```

<i>NUM_EMPL</i>	<i>NOMBRE</i>	<i>SUM(IMPORTE)</i>
101	Daniel Gutierrez	26.628
102	Soledad Martinez	22.776
103	Pedro Cruz	2.700
105	Belen Aguirre	39.327
106	Jose Maldonado	32.958
107	Natalia Martin	34.432
108	Lorenzo Fernandez	58.533
109	Maria Garcia	7.105
110	Antonio Valle	23.132

Selección en conjuntos agrupados (I)

- En una consulta sumaria agrupada es posible seleccionar los conjuntos de datos a mostrar.
- La selección se realiza sobre los conjuntos de datos formados por GROUP BY con **HAVING**
- HAVING examina los conjuntos de datos y elimina todos aquellos que no cumplen las condiciones indicadas.

Selección en conjuntos agrupados (II)

```
SELECT rep, AVG(importe) FROM pedidos GROUP BY rep  
HAVING SUM(importe) > 30000
```

<i>REP</i>	<i>AVG(IMPORTE)</i>
105	7.865,40
106	16.479,00
107	11.477,33
108	8.361,86

```
SELECT ciudad, SUM(cuota), SUM(repventas.ventas) FROM  
oficinas, repventas WHERE oficina = oficina_rep  
GROUP BY ciudad HAVING COUNT(*) >= 2
```

<i>CIUDAD</i>	<i>SUM(CUOTA)</i>	<i>SUM(REPVENTAS.VENTAS)</i>
Barcelona	70.000	29.328
Madrid	60.000	81.309
Valencia	52.500	40.063

Selección en conjuntos agrupados (III)

```
SELECT descripcion, precio, existencias, SUM(cant)
FROM productos, pedidos WHERE fab = id_fab AND
producto = id_producto GROUP BY id_fab, id_producto,
descripcion, precio, existencias HAVING SUM(cant) >
(0.75*existencias) ORDER BY existencias DESC
```

<i>DESCRIPCION</i>	<i>PRECIO</i>	<i>EXISTENCIAS</i>	<i>SUM(CANT)</i>
Reductor	355	38	32
Ajustador	25	37	30
Bancada Motor	243	15	16
Bisagra Dcha.	4.500	12	15
Riostra 1-Tm	1.425	5	22

Noción intuitiva de subconsulta

Pregunta: Oficinas donde las ventas son inferiores al 50% de la suma de las cuotas de los vendedores.

Oficinas con ventas inferiores a un valor:

```
SELECT ciudad FROM oficinas WHERE ventas < ???
```

50% de las ventas de los vendedores de una oficina:

```
SELECT 0.5*SUM(cuota) FROM repventas WHERE oficina_rep = XX
```

Respuesta:

```
SELECT ciudad FROM oficinas WHERE ventas < (SELECT  
0.5*SUM(cuota) FROM repventas WHERE oficina_rep = oficina)
```

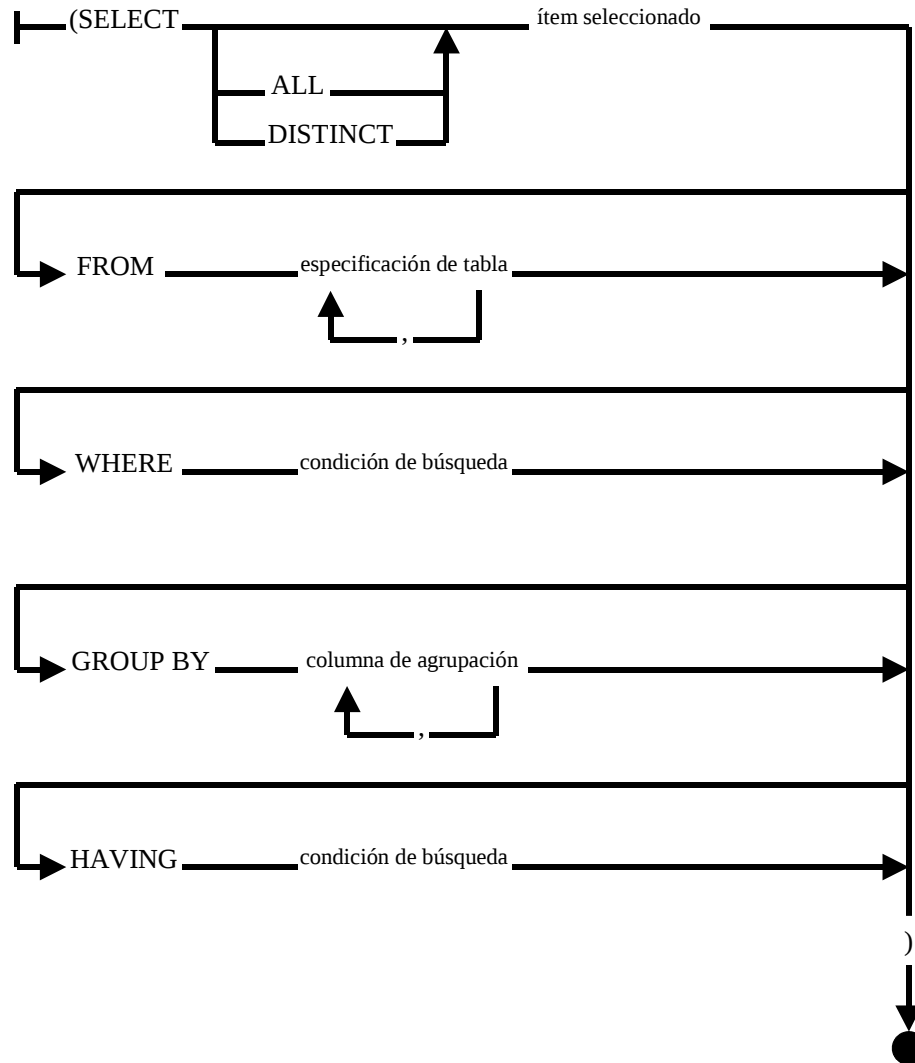

Subconsultas (I)

- Son consultas SQL que se utilizan como condiciones en las cláusulas WHERE o HAVING.
- Permiten descomponer una consulta complicada en consultas más sencillas.
- Existen consultas SQL que no pueden expresarse sin utilizar subconsultas.
- Una subconsulta solo puede devolver una única columna.

Subconsultas (II)

- La subconsulta puede utilizar los campos de:
 - Las tablas definidas en la subconsulta.
 - Las tablas definidas en la consulta.
- Si una tabla se encuentra definida en la subconsulta y en la consulta, se utilizan los campos de la subconsulta y no los de la consulta.

Diagrama sintáctico de una subconsulta



Tests en subconsultas

- Cuatro tipos de tests:
 - Test de comparación.
 - Test de pertenencia a conjunto.
 - Test de existencia.
 - Test de comparación cuantificada.

Test de comparación (I)

- Compara el resultado de una subconsulta, que debe ser un solo valor, con un valor de la consulta.
- Seis condiciones de comparación: =, <>, <, <=, >, >=

```
SELECT nombre FROM repventas WHERE cuota >= (SELECT  
objetivo FROM oficinas WHERE ciudad = 'Alicante')
```

<i>NOMBRE</i>
Belen Aguirre
Lorenzo Fernandez
Soledad Martinez

Test de comparación (II)

```
SELECT empresa FROM clientes WHERE rep_clie = (SELECT  
num_emp1 FROM repventas WHERE nombre = 'Belen  
Aguirre')
```

<i>EMPRESA</i>
Pino S.L.
JPF S.L.

```
SELECT descripcion, existencias FROM productos WHERE  
id_fab = 'ACI' AND existencias > (SELECT existencias  
FROM productos WHERE id_fab = 'ACI' AND id_producto  
= '41004')
```

<i>DESCRIPCION</i>	<i>EXISTENCIAS</i>
Articulo Tipo 1	277
Articulo Tipo 2	167
Articulo Tipo 3	207

Test de pertenencia a conjunto (I)

- Comprueba si un valor de una consulta se encuentra en una columna de valores de una subconsulta.

```
SELECT nombre FROM repventas WHERE oficina_rep IN  
  (SELECT oficina FROM oficinas WHERE ventas >  
    objetivo)
```

<i>NOMBRE</i>
Belen Aguirre
Lorenzo Fernandez
Soledad Martinez
Natalia Martin

Test de pertenencia a conjunto (II)

```
SELECT nombre FROM repventas WHERE num_empl NOT IN  
(SELECT rep_clie FROM clientes WHERE limite_credito  
    < 50000)
```

<i>NOMBRE</i>
Lorenzo Fernandez

```
SELECT empresa FROM clientes WHERE num_clie IN (SELECT  
    DISTINCT clie FROM pedidos WHERE fab = 'ACI' AND  
    producto LIKE '4100%' AND fecha_pedido BETWEEN  
        '01/01/2000' AND '30/06/2000')
```

<i>EMPRESA</i>
EVBE S.A.
Pino S.L.
Distribuciones Sur S.A.
Zapater Importaciones S.A.

Test de existencia (I)

- Comprueba si una subconsulta produce filas de resultados.
- Dos posibilidades:
 - Existe al menos una fila **EXISTS**
 - No existe ninguna fila **NOT EXISTS**
- Puede utilizarse * (todas las columnas) en lugar del nombre de una columna.

Test de existencia (II)

```
SELECT DISTINCT descripcion FROM productos WHERE  
EXISTS (SELECT num_pedido FROM pedidos WHERE fab =  
id_fab AND producto = id_producto AND importe >=  
25000)
```

<i>DESCRIPCION</i>
Bisagra Dcha.
Bisagra Izqda.
Extractor
Riostra 1-Tm

```
SELECT empresa FROM clientes WHERE rep_clie = (SELECT  
num_empl FROM repventas WHERE nombre = 'Soledad  
Martinez') AND NOT EXISTS (SELECT * FROM pedidos  
WHERE clie = num_clie AND importe > 3000)
```

<i>EMPRESA</i>
Hnos. Martinez S.A.
Construcciones Leon S.A.

Test cuantificados (I)

- Utilizan los operadores de comparación =, <>, <, <=, >, >=
- Comparan un valor de la consulta con los resultados de la subconsulta de forma que se cumpla el operador de comparación para:
 - Algún valor de la subconsulta: ANY
 - Todos los valores de la subconsulta: ALL

Test cuantificados (II)

SELECT nombre FROM repventas WHERE cuota <ANY (SELECT importe FROM pedidos WHERE rep = num_emp1)

<i>NOMBRE</i>
Jose Maldonado
Lorenzo Fernandez
Natalia Martin

SELECT ciudad, objetivo FROM oficinas WHERE objetivo <ALL (SELECT ventas FROM repventas WHERE oficina_rep = oficina)

<i>CIUDAD</i>	<i>OBJETIVO</i>
Toledo	27.500
Alicante	30.000

Subconsultas en la cláusula HAVING

- Una subconsulta puede aparecer en una cláusula HAVING.
- Utiliza los operadores de comparación.

```
SELECT nombre, AVG(importe) FROM repventas, pedidos  
WHERE num_empl = rep AND fab = 'ACI' GROUP BY nombre  
HAVING AVG(importe) > (SELECT AVG(importe) FROM  
pedidos)
```

<i>NOMBRE</i>	<i>AVG(IMPORTE)</i>
Antonio Valle	22.500,00
Soledad Martinez	15.000,00

Subconsultas anidadas

- Una subconsulta puede contener subconsultas en su interior.

```
SELECT empresa FROM clientes WHERE rep_clie IN (SELECT  
num_empl FROM repventas WHERE oficina_rep IN (SELECT  
oficina FROM oficinas WHERE region = 'Centro'))
```

<i>EMPRESA</i>
Hnos. Martinez S.A.
Lopez Asociados S.L.
Componentes Fernandez S.A.
Domingo S.L.
Distribuciones Montiel S.L.
Construcciones Leon S.A.
Exclusivas Norte S.A.