









TEMA


Sistema de Gestión de Bases de Datos

   UNIVERSIDAD CARLOS III DE MADRID 

Sistemas Avanzados de Recuperación de Información (SARI)
2008-2009





Jorge Morato Lara
Sonia Sánchez-Cuadrado


   UNIVERSIDAD CARLOS III DE MADRID 



Índice


1. Tema - Recuperación de la Información en Internet
2. Tema -Técnicas de Representación y Organización en Internet
3. Tema - Modelos y Técnicas de Recuperación de Información
4. Tema - Sistemas de Gestión de Bases de Datos

   UNIVERSIDAD CARLOS III DE MADRID 




SQL

- ❑ Como **DDL** (Definición de datos) nos permite Crear y borrar tablas y relaciones (mediante CREATE, DROP y ALTER).
- ❑ Como **DML** (Manipulación de datos) están SELECT (selección registros), UPDATE (actualizar registros), DELETE (borrar registros) e INSERT (añadir registros). Sirve para que consultemos y modifiquemos los datos.
- ❑ Como Lenguaje de Control. GRANT (para dar privilegios), REVOKE (quitar privilegios), EXPLAIN y LOCK. Que sirven para controlar el acceso a las tablas.
- ❑ Guía de referencia en:
<https://aurora.vcu.edu/db2help/db2s0/frm3toc.htm>




UNIVERSIDAD CARLOS III DE MADRID




Características y Utilidades

- ❑ Estándar ISO y ANSI
- ❑ Es el lenguaje más universal existente para trabajar con BD
- ❑ Se puede insertar dentro del código de la mayoría de lenguajes de programación para así acceder a datos de BD (Visual C, .Net, ...) [Forma inmersa en un lenguaje anfitrión]
- ❑ Se puede emplear dentro de cualquier base de datos relacional actual (Oracle, Access, SQL Server...) [existen otros lenguajes como QBE, QUEL...]
- ❑ Es sencillo
- ❑ Muchas consultas no se pueden realizar en la ventana de diseño de consultas



UNIVERSIDAD CARLOS III DE MADRID




Consultas de definición de datos. Creación de tablas


El comando CREATE sirve para crear una tabla nueva.


CREATE TABLE nombre-tabla-nueva
(nombre-campo1 tipoDato otrasPropiedades, nombre-campo2 tipoDato otrasPropiedades, ...)

TipoDato: Integer, String, char, bit, date, real, etc
Otras propiedades: tamaño del campo (número caracteres entre paréntesis si es string), PRIMARY KEY, Not null, ...




UNIVERSIDAD CARLOS III DE MADRID






Tipos de Datos

Tipo de Datos	Longitud	Descripción
BINARY	1 byte	Para consultas sobre tabla adjunta de productos de bases de datos que definen un tipo de datos Binario.
BIT	1 byte	Valores Si/No ó True/False
BYTE	1 byte	Un valor entero entre 0 y 255.
COUNTER	4 bytes	Un número incrementado automáticamente (de tipo Long)
CURRENCY	8 bytes	Un entero escalable entre 922.337.203.685.477,5808 y 922.337.203.685.477,5807.
DATETIME	8 bytes	Un valor de fecha u hora entre los años 100 y 9999.
SINGLE	4 bytes	Un valor en punto flotante de precisión simple con un rango de - 3.402823*1038 a -1.401298*10-45 para valores negativos, 1.401298*10- 45 a 3.402823*1038 para valores positivos, y 0.
DOUBLE	8 bytes	Un valor en punto flotante de doble precisión con un rango de - 1.79769313486232*10308 a - 4.94065645841247*10-324 para valores negativos, 4.94065645841247*10-324 a 1.79769313486232*10308 para valores positivos, y 0.
SHORT	2 bytes	Un entero corto entre -32,768 y 32,767.
LONG	4 bytes	Un entero largo entre -2,147,483,648 y 2,147,483,647.
LONGTEXT	1 byte por carácter	De cero a un máximo de 1.2 gigabytes.
LONGBINAR Y	Según se necesite	De cero 1 gigabyte. Utilizado para objetos OLE.
TEXT	1 byte por carácter	De cero a 255 caracteres.



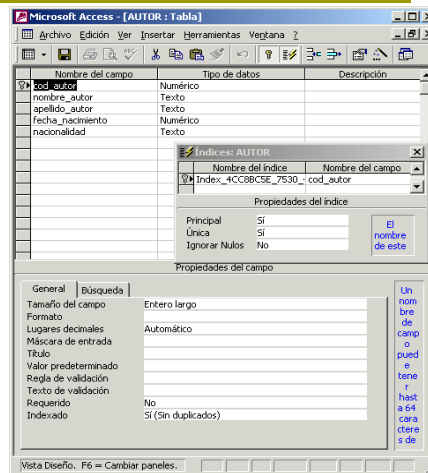
UNIVERSIDAD CARLOS III DE MADRID



Consultas de definición de datos. Creación de tablas



```
CREATE TABLE AUTOR (cod_autor  
integer PRIMARY KEY,  
nombre_autor text (70) not null,  
apellido_autor text (70),  
fecha_nacimiento integer,  
nacionalidad text (50))
```



UNIVERSIDAD CARLOS III DE MADRID



Crear una relación entre tablas con create



Las relaciones entre tablas suelen ser entre una primary key (clave principal) y una foreign key (clave ajena)

```
CREATE TABLE nombre-tabla  
(nombre-campo1 tipoDato propiedad, nombre-campo2  
tipoDato propiedad,...,  
CONSTRAINT nombre_clave  
FOREIGN KEY (campo_clave_ajena) REFERENCES tabla-  
a-relacionar (campo-de-la-tabla-a-relacionar))
```



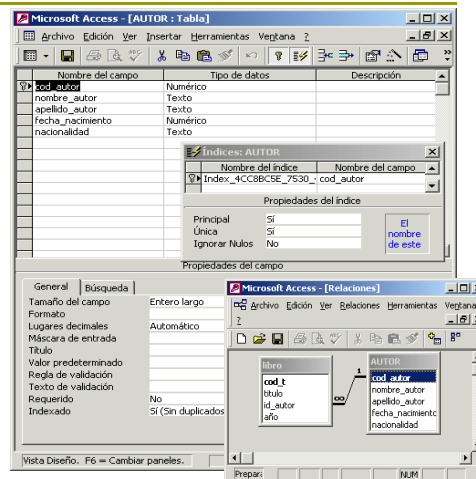
UNIVERSIDAD CARLOS III DE MADRID



Consultas de definición de datos. Creación de tablas



```
CREATE TABLE LIBRO
(cod_t integer primary
key, titulo text (70) not
null, id_autor integer,
año integer,
CONSTRAINT f FOREIGN
KEY (id_autor)
REFERENCES
autor(cod_autor))
```



UNIVERSIDAD CARLOS III DE MADRID



Crear una Tabla Nueva



- Los tipos de datos pueden ser: **text** (o **string**), **date**, si/no (**bit**), número (**integer**, **float**, **real**, etc), moneda (**currency**)...


A partir de otra tabla también se puede crear una nueva tabla:

- SELECT** campo1_origen[, campo2[, ...]] **INTO** nuevatabla [IN basedatosexterna] **FROM** tabla_origen



UNIVERSIDAD CARLOS III DE MADRID






DDL. Modificar la estructura de una tabla



- El COMANDO ALTER TABLE sirve para añadir, modificar, eliminar campos y claves de una tabla

`ALTER TABLE` **tabla-a-modificar**
`ADD/ALTER/DROP COLUMN` **campo**

ADD añade columna, **DROP** la elimina y **ALTER COLUMN** modifica su tipo de datos o tamaño

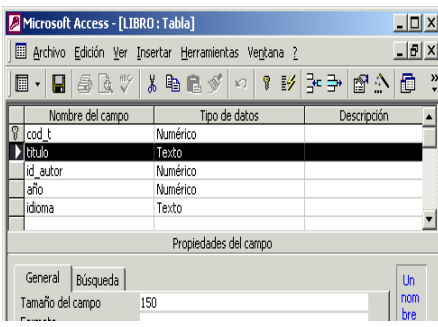


UNIVERSIDAD CARLOS III DE MADRID



DDL. Modificar la estructura de una tabla

- Añadir columna idioma:**
`ALTER TABLE libro`
`ADD COLUMN idioma text (15)`
- Añadir columna lugar:**
`ALTER TABLE libro`
`ADD COLUMN lugar text (15)`
- Modificar el tamaño de la columna titulo:**
`ALTER TABLE libro`
`ALTER COLUMN titulo text (150)`
- Eliminar la columna lugar:**
`ALTER TABLE libro`
`DROP COLUMN lugar`




Nombre del campo	Tipo de datos	Descripción
cod_t	Númérico	
titulo	Texto	
id_autor	Númérico	
año	Númérico	
idioma	Texto	


Propiedades del campo

General Búsqueda Un nombre

Tamaño del campo 150



UNIVERSIDAD CARLOS III DE MADRID



Modificar y Eliminar una tabla: Crear índices



MODIFICAR UNA TABLA

- **ALTER TABLE** tabla **ADD COLUMN** campo tipo [(tamaño)] [NOT NULL] [CONSTRAINT índice] | **CONSTRAINT** índice múltiples campos | **DROP COLUMN** campo | **CONSTRAINT** nombre índice

ELIMINAR UNA TABLA O INDICE

- **DROP TABLE** tabla | **INDEX** índice **ON** tabla

CREAR INDICES

- **CREATE** [**UNIQUE**] **INDEX** índice **ON** tabla(campo [ASC|DESC][, campo [ASC|DESC], ...]) [WITH { **PRIMARY** | **DISALLOW NULL** | **IGNORE NULL** }]



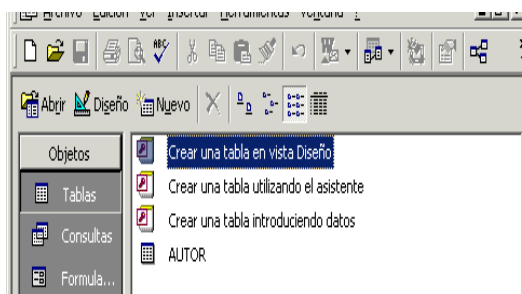
UNIVERSIDAD CARLOS III DE MADRID



DDL. Eliminar una tabla




- **DROP TABLE**
tabla-a-eliminar
- **DROP TABLE**
libro



UNIVERSIDAD CARLOS III DE MADRID








INSERT

- Sirve para anexar datos, esto es añadir una nueva fila con datos a determinada tabla

```
INSERT INTO Tabla-a-anexar
(campos-de-la-tabla-a-rellenar)
VALUES (valores con los que completar los campos);

INSERT INTO SALARIO
(CA_PERSONAL, SUELDO, MES, AÑO)
VALUES (30, 128000, "Diciembre", "2004");
```

 UNIVERSIDAD CARLOS III DE MADRID 

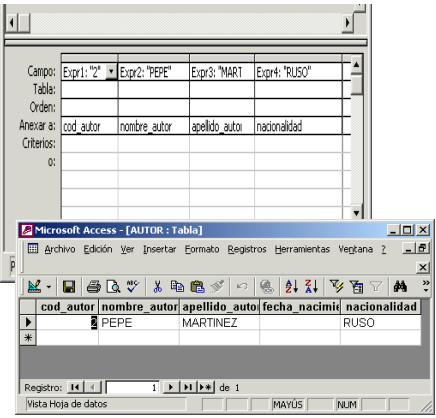


INSERT



INSERT INTO AUTOR


(cod_autor,nombre_autor,apellido_autor,nacionalidad)

VALUES (2,"PEPE", "MARTINEZ", "RUSO")



cod_autor	nombre_autor	apellido_autor	fecha_nacimiento	nacionalidad
2	PEPE	MARTINEZ		RUSO



 UNIVERSIDAD CARLOS III DE MADRID 




UPDATE (Actualizaciones)

- Cambia el valor de una o varias celdas por un nuevo valor
`UPDATE TABLA-A-ACTUALIZAR`
`SET CAMPO-A-ACTUALIZAR="VALOR-NUEVO"`
`WHERE CAMPO-A-ACTUALIZAR="VALOR ANTIGUO"`

- Ejemplo para actualizar la tabla Salario y poner en el campo CATEGORIA el valor jefe siempre que aparezca la palabra consejero
`UPDATE SALARIO`
`SET CATEGORIA = "jefe"`
`WHERE CATEGORIA="consejero"`

 UNIVERSIDAD CARLOS III DE MADRID 



UPDATE

```
UPDATE autor
SET autor.nacionalidad =
"escocesa"
WHERE
autor.nacionalidad="ruso"
```



Microsoft Access - [AUTOR: Tabla]


cod_autor	nombre_autor	apellidos_autor	fecha_nacimiento	nacionalidad
	PEPE	MARTINEZ		escocesa
*				

autor

- * cod_autor
- nombre_autor
- apellidos_autor
- fecha_nacimiento


Campo: nacionalidad
Tabla: autor
Actualizar a: "escocesa"
Criterios:

 UNIVERSIDAD CARLOS III DE MADRID 





Resumen Actualización

- **UPDATE** tabla **SET** campo=loquesea **WHERE** criterio;
Si hay varias tablas:
- **UPDATE** tabla1 **INNER JOIN** tabla2 **ON**
tabla1.campo1=tabla2.campo2 **SET** campo=loquesea
WHERE criterio



UNIVERSIDAD CARLOS III DE MADRID




DELETE (Borrar)


Sirve para eliminar los registros que cumplan alguna condición


```
DELETE CAMPO-CUYO-VALOR-SE-QUIERE-BORRAR  
FROM TABLA-QUE-CONTIENE-EL-CAMPO  
WHERE CAMPO-CUYO-VALOR-SE-ELIMINA-SI-TIENE-CIERTO-  
VALOR=VALOR
```

DELETE * **FROM** autor **WHERE** nacionalidad='escocesa'



UNIVERSIDAD CARLOS III DE MADRID





Base de Datos de ejemplo


COD	NOMBRE	DNI	FECHA	SALARIO	SEX
10	Hernandez, Cris	34636321	651001		F
20	Tapia, Miguel	55789642	731010		M
30	García, Ana	20389571	750405	38250	F

Tabla PERSONAL


PERSONAL	SALARIO
COD	CA_PERSONAL
NOMBRE	SUELDO
DNI	MES
FECHA	AÑO
SALARIO	CATEGORIA


CA_PER	SUELDO	MES	AÑO	CATEGORIA
10	150253	Agosto	2004	Administrativo
10	120000	Septiembre	2004	Administrativo
10	120000	Ocutbre	2004	Administrativo
20	450000	Septiembre	2004	Consejero
20	450000	Octubre	2004	Consejero
30	120000	Julio	2004	Administrativo
30	150253	Agosto	2004	Administrativo
30	650000	Septiembre	2004	Consejero

Tabla SALARIO



UNIVERSIDAD CARLOS III DE MADRID





SELECT

```

SELECT campo1,campo2
FROM tabla
WHERE condición
GROUP BY campo
ORDER BY campo
HAVING condición
    
```

Obligatorio. Pon los campos que quieres ver como resultado de la consulta separados por comas, si todos →*


Pon las tablas donde están los campos, si varias pon comas

Condición(-es) que deben cumplir los registros que visualices. Si coexisten AND y OR,... usa paréntesis. Si es campo no numérico pon contenido entre comillas. Si pones comodines pon LIKE


Agrupar por un campo o campos

Ordenar por un campo o campos ASC ascendente o DESC inverso

Condición tras agrupar



UNIVERSIDAD CARLOS III DE MADRID



Funciones de agregado y valores de WHERE



FUNCIONES DE AGREGADO

AVG (media), **COUNT** (contar), **SUM** (sumar), **MAX** (el máximo), **MIN** (el mínimo)

CAMPO WHERE

- **LIKE** "texto*"
- **LIKE** "texto?"
- =numero (tb >=, <=, <>(distinto))
- ="texto"
- **BETWEEN A AND B**
- **IS NOT NULL/ IS NULL**
- Normalmente si queremos negar una situación se usa NOT tras el nombre del campo y luego la condición
- Para combinar varias condiciones en el WHERE se usan paréntesis y operadores booleanos (AND, OR)



UNIVERSIDAD CARLOS III DE MADRID



Ejemplos SELECT



- Selecciona todos los campos y todos los registros de la tabla personal
SELECT * FROM PERSONAL
- Selecciona los campos cod, nombre y fecha y todos los registros
SELECT cod,nombre,fecha FROM PERSONAL
- Selecciona nombre y DNI de las empleadas de la empresa
SELECT nombre,DNI FROM PERSONAL WHERE SEXO="F"
- Selecciona los empleados con el campo salario de la tabla PERSONAL vacío
SELECT nombre,DNI FROM PERSONAL WHERE SALARIO IS NULL
- Selecciona todos los empleados cuyo apellido comience por T
SELECT * FROM PERSONAL WHERE NOMBRE LIKE "T*"



UNIVERSIDAD CARLOS III DE MADRID



Ejemplos SELECT



- Selecciona los empleados cuyo nombre contenga una “e” o que sean mujeres
`SELECT * FROM PERSONAL WHERE NOMBRE LIKE “M*” OR SEXO=“F”`
- Selecciona de la tabla salario los sueldos mayor que 200000 y distintos de 450000 que no pertenezcan al mes de Agosto
`SELECT sueldo FROM SALARIO WHERE SUELDO > 200000 AND SUELDO <> 450000`
- Selecciona de la tabla salario los sueldos entre 100000 y 150000 que no pertenezcan al mes de Agosto
`SELECT sueldo FROM SALARIO WHERE SUELDO BETWEEN 100000 AND 150000 AND MES NOT LIKE “AGOSTO”`
- Selecciona los empleados con el campo salario de la tabla PERSONAL no este vacío
`SELECT nombre,DNI FROM PERSONAL WHERE SALARIO IS NOT NULL`



UNIVERSIDAD CARLOS III DE MADRID



Ejemplos SELECT



- Ordena a los empleados por nombre ascendente
`SELECT * FROM PERSONAL ORDER BY NOMBRE ASC`
- Selecciona los empleados cuyo nombre contenga la palabra garcía y que sean mujeres o cuyo DNI sea 55789642 ordena por numero de DNI descendente
`SELECT * FROM PERSONAL WHERE ((NOMBRE LIKE “*GARCÍA*” AND SEX=“F”) OR DNI=55789642) ORDER BY DNI DESC`
- Mostrar en una sola fila y sin duplicados los meses distintos que aparecen en la tabla salarios
`SELECT DISTINCT mes FROM SALARIO`



UNIVERSIDAD CARLOS III DE MADRID



SELECT Funciones y Agrupamientos

SELECT campo1
AVG(campo2)
FROM tabla

WHERE condición



GROUP BY campo
ORDER BY campo
HAVING condición

Función de agrupamiento, para calcular la media (AVG), suma (SUM), contar (COUNT), valor mínimo (MIN), máximo (MAX)...

Opcional. Condición antes de agrupar

Agrupar por un campo o campos, cuando se ha puesto una función de agrupamiento en el SELECT todos los campos sin función deben estar agrupados. Si varios se separan por comas

Opcional. Condición tras agrupar



UNIVERSIDAD CARLOS III DE MADRID

EJEMPLOS

- Suma de lo pagado en el mes de Agosto

```
SELECT Sum(SUELDO) FROM SALARIO WHERE MES="agosto"
```
- Media de lo pagado a los empleados cada mes. El campo calculado deberá llamarse media

```
SELECT Avg(SALARIO.SUELDO) AS Media, SALARIO.MES  
FROM SALARIO  
GROUP BY SALARIO.MES
```

Función promedio

El nombre del campo se puede poner con su tabla

Mediante "AS" se puede cambiar el nombre a un campo

Recordar agrupar los campos que no tienen función en el SELECT



UNIVERSIDAD CARLOS III DE MADRID

EJEMPLOS



- Contar el número de salarios que se pagaron en octubre del 2004

```
SELECT AÑO, MES, Count(MES) AS Cuenta
FROM SALARIO
WHERE AÑO=2004 AND MES="octubre"
GROUP BY AÑO, MES
```

- Que sueldo cobraron los empleados que ganaron más de 800000 en el 2004

```
SELECT Sum(SUELDO) AS Suma, AÑO, CA_PERSONAL
FROM SALARIO
WHERE AÑO=2004
GROUP BY AÑO, CA_PERSONAL
HAVING SUM(SUELDO)>=800000
```



UNIVERSIDAD CARLOS III DE MADRID



Notas: Evitar campos ambiguos



- Si en vez del nombre del campo pones el nombre de la tabla un punto y nombre del campo quedará menos ambiguo

```
SELECT Avg(SALARIO.SUELDO) AS Media, SALARIO.MES
FROM SALARIO
GROUP BY SALARIO.MES
```

El nombre del campo
autor.titulo en vez
de **titulo**



UNIVERSIDAD CARLOS III DE MADRID





CONSULTA DE UNIÓN

- Requisitos: deben de tener igual estructura las dos tablas
- La consulta unión sirven para ver en un único campo los datos de igual tipo de varias tablas. En Access se realiza en diseño de consultas, menú consulta, opción específica de SQL y Unión. O directamente escribiéndola en la pantalla de SQL. No se puede hacer en modo gráfico.
- `SELECT [CAMPO1], [CAMPO2] FROM [TABLA1] UNION SELECT [CAMPO1], [CAMPO2] FROM [TABLA2];`
- La TABLA1 y la TABLA2 deben de tener el mismo número de campos.
- El resultado es que devuelve en una misma columna el resultado de las dos tablas.
- Los registros duplicados se eliminan. Si no se quiere que se eliminen se escribe UNION ALL

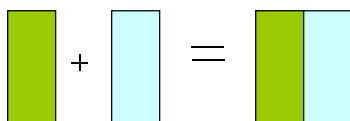


UNIVERSIDAD CARLOS III DE MADRID

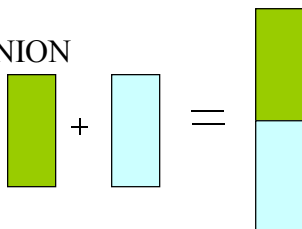


JOIN vs UNION

JOIN



UNION


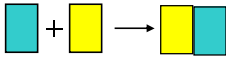
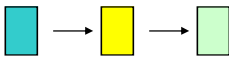


UNIVERSIDAD CARLOS III DE MADRID



CONSULTAS A PARTIR DE VARIAS TABLAS

- ❑ Cuando se unen varias tablas es mejor poner el nombre completo (tabla.campo) para evitar ambigüedades
- ❑ CON SUBCONSULTAS
 - Menos trabajo para el ordenador
 - Se trata de unir varios selects, inserts... seguidos
- ❑ CON JOIN
 - Necesario si se quiere mostrar campos de varias tablas simultáneamente
 - Se trata de unir varias tablas en una a partir de campos del mismo tipo (usualmente, aunque no siempre, con clave principal-clave ajena)





JOIN

- ❑ Mostrar en una consulta el nombre de los empleados junto con la remuneración en el 2004
- ❑ Existen dos formas equivalentes
 - CON INNER JOIN...ON

```
SELECT PERSONAL.NOMBRE, Sum(SALARIO.SUELDO) AS Remuneracion
FROM PERSONAL INNER JOIN SALARIO ON PERSONAL.COD =
SALARIO.CA_PERSONAL WHERE SALARIO.AÑO=2004 GROUP BY
PERSONAL.NOMBRE;
```
 - PONIENDO LA INFORMACIÓN DE JOIN EN EL WHERE

```
SELECT PERSONAL.NOMBRE, Sum(SALARIO.SUELDO) AS Remuneracion
FROM PERSONAL,SALARIO WHERE PERSONAL.COD = SALARIO.CA_PERSONAL
AND SALARIO.AÑO=2004
GROUP BY PERSONAL.NOMBRE;
```






JOIN: DELETE



Por ejemplo para eliminar en la tabla salarios los registros relacionados con Ana García será:

```
DELETE SALARIO.*  
FROM PERSONAL  
INNER JOIN SALARIO  
ON PERSONAL.COD = SALARIO.CA_PERSONAL  
WHERE PERSONAL.NOMBRE="García, Ana"
```

O en la tabla autor:




UNIVERSIDAD CARLOS III DE MADRID




SUBCONSULTAS


- ❑ Mostrar en una consulta el nombre de los empleados que cobraron un sueldo en algún mes del 2004 superior a 200000 y que son mujeres
 - **Primero** tendré que ver en la tabla SALARIO que sueldos fueron en el 2004 superiores a 200000 y retener el valor del CA_PERSONAL. Si lo hago "a mano" puedo comprobar que los empleados con el CA_PERSONAL igual a 20 y a 30 cobraron
 - **Segundo** tendré que ver en la tabla PERSONAL que nombres tienen los empleados cuyo campo COD tiene los números 20 y 30 y cuyo campo sexo es igual a "f"
- 2 La forma de hacer la subconsulta es invirtiendo el orden anterior, primero pondremos el paso segundo y después el primero, de la siguiente forma:



UNIVERSIDAD CARLOS III DE MADRID



SUBCONSULTA



Mostrar en una consulta el nombre de los empleados que cobraron un sueldo en el 2004 superior a 200000 y que son mujeres



2 `SELECT NOMBRE, SEX`
`FROM PERSONAL`
`WHERE COD IN`

1 `SELECT CA_PERSONAL`
`FROM SALARIO`
`WHERE SUELDO>200000)`
`AND SEX="F"`


Esto indica que los registros resultado de la sentencia entre paréntesis se transfieren al WHERE superior

} Subselect

- * Aquí no es necesario el nombre completo de los campos pues no hay ambigüedad
- * La forma de pasar valores de una sentencia a otra es con:
 ...WHERE campo IN (...)

UNIVERSIDAD CARLOS III DE MADRID



Ejemplo de subconsulta



NOMBRE		DIRECCION-NOMBRE			DIRECCION	
1ER_APEL	CP_NOM	CA_NOM	CA_DIREC	CA_DIREC	CALLE	
Martínez	1	1	1	1	3 C/Pez, 7	
Gómez	2	2	2	2	2 Av. Murcia, 11	
López	3	2	3	3	2 C/Caniche, 2	

Q: SELECCIONAR LOS APELLIDOS DE LAS PERSONAS QUE VIVAN EN LA CALLE CANICHE

Sin subconsultas habría que hacer tres selects:

1º CONSULTAR EN LA TABLA DIRECCIÓN LA CA_DIRECCIÓN CUANDO CALLE ES C/CANICHE

`SELECT CA_DIREC FROM DIRECCION WHERE CALLE LIKE '*caniche*'`

RESULTADO=2




UNIVERSIDAD CARLOS III DE MADRID



Ejemplo de subconsulta



2º CONSULTAR EN LA TABLA DIRECCIÓN-NOMBRE LA CA_NOMBRE CUANDO CA_DIRECCION ES 2

```
SELECT CA_NOM FROM DIRECCION-NOMBRE WHERE CA_DIREC=2  
RESULTADO =1
```

3º CONSULTAR EN LA TABLA NOMBRE EL APELLIDO CUANDO CP_NOMBRE ES 1

```
SELECT 1ER_APEL FROM NOMBRE WHERE CP_NOM=1  
RESULTADO = MARTINEZ
```

EN UNA SOLA CONSULTA CON SUBCONSULTAS

```
SELECT 1ER_APEL FROM NOMBRE WHERE CP_NOM IN (SELECT  
CA_NOM FROM DIRECCION-NOMBRE WHERE CA_DIREC IN  
(SELECT CA_DIREC FROM DIRECCION WHERE CALLE  
LIKE '*caniche*'))
```



UNIVERSIDAD CARLOS III DE MADRID



SUBCONSULTA vs JOIN




- ❑ Muchas consultas se pueden realizar indistintamente con JOIN y SUBCONSULTAS
- ❑ JOIN es como hace la unión de tablas ACCESS por defecto
- ❑ Existe un caso en el que no se puede emplear subconsultas en vez de JOIN y es cuando nos piden que como resultado mostremos el contenido de varios campos procedentes de distintas tablas



UNIVERSIDAD CARLOS III DE MADRID









NOTAS: CASILLAS EN BLANCO


Puede haber dos razones para que una casilla este en blanco.

	Causa	Por defecto Access	Forma de rellenar la celda	Búsqueda
NULO	Se desconoce el valor	Permitido	No se inserta nada	Where <campo> is null
CADENA LONGITUD CERO	No existe el valor	No permitido	Se insertan dos comillas ""	Where <campo> = ""

UNIVERSIDAD CARLOS III DE MADRID









SELECT


```

SELECT [ALL | DISTINCT] [nombre = expresión | expresión | AS nombre]
FROM [nombre-de-tabla | nombre-de-vista | nombre-de-correlación]
WHERE [condición-de-búsqueda]
GROUP BY [constante-entera | nombre-de-columna]
HAVING [condición-de-búsqueda]
ORDER BY [constante-entera | nombre-de-columna] [ASC | DESC]
FOR UPDATE OF [nombre-de-columna]
UNION [ALL] comando-select
ORDER BY [constante-entera] [ASC | DESC]
    
```

UNIVERSIDAD CARLOS III DE MADRID





UPDATE

→ UPDATE nombre-de-tabla nombre-de-correlación
nombre-de-vista nombre-de-correlación




→ SET nombre-de-columna= expresión
NULL

→ WHERE condición-de-búsqueda
CURRENT OF nombre-de-cursor

→ [CHECK EXISTS]

* UPDATE = Operación de modificación. Modifica el valor de una o más columnas de una tabla

* CLAUSULA WHERE , WHERE CURRENT OF nombre de cursor





DELETE

→ DELETE FROM nombre-de-tabla nombre-de-correlación
nombre-de-vista nombre-de-correlación

→ WHERE condición-de-búsqueda
CURRENT OF nombre-de-cursor

* DELETE = Operación de modificación. Elimina una o más filas de una tabla

* CLAUSULA WHERE , WHERE CURRENT OF nombre de cursor





INSERT

