

Resumen-AR-y-SQL.pdf



Lisenk



Fundamentos de Bases de Datos



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





**KEEP
CALM
AND
ESTUDIA
UN POQUITO**



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



Resumen AR

Operador	Notación
Selección	σ
Proyección	π
Unión	\cup
Intersección	\cap
Diferencia	$-$
Producto cartesiano	\times
Θ -reunión	\bowtie_{Θ}
División	\div

Operadores monarios

- **Selección:** *Selecciona un conjunto de filas de una tabla dada (corte horizontal)*
- **Proyección:** *Elige las columnas que necesitamos de una tabla dada (corte vertical). Además, elimina las tuplas redundantes de una proyección.*

Producto cartesiano

Permite conectar tablas

A	B		D
a_1	b_1	\times	d_1
a_2	b_2	\times	d_2
a_3	b_3	\times	d_3

Cuando aparezcan atributos con el mismo nombre de distintas tablas, se cualificarán los atributos con el nombre. De la tabla a cuyo esquema pertenecen.

La O-Reunión

Operación que verdaderamente permite pegar dos tablas y restablecer las conexiones semánticas existentes entre ellas.

1. **Determinar las tablas que intervienen en la consulta (selección previa si es necesario)**
2. **Determinar los atributos comunes en ambas tablas que establecen la conexión semántica entre ellas.**
3. **Formular la selección y el producto cartesiano.**
4. **Proyectar para obtener los atributos adecuados.**

Obtener para cada profesor, su NRP, su nombre y el nombre del departamento al que pertenece.

El proceso de resolución es:

1. Las tablas implicadas son profesores y departamentos.
2. Los atributos comunes a estas tablas, que establecen la conexión de pertenencia, son los códigos de departamento (COD_DEP).
3. La tabla completa sería:

$$\sigma_{\text{departamentos.cod_dep}=\text{profesores.cod_dep}}(\text{departamentos} \times \text{profesores})$$

4. Proyectando sobre los atributos deseados, obtenemos la solución a la consulta:

$$\pi_{\text{NRP,nom_prof,nom_dep}}(\sigma_{\text{departamentos.cod_dep}=\text{profesores.cod_dep}}(\text{departamentos} \times \text{profesores}))$$

I

Conexión de una tabla consigo misma: los alias

1. Definir un alias $\rho(\text{alumnos}) = \text{alu}$
2. Seleccionar $\text{nom_alum} = \text{"Luis Martínez Pérez"}$ sobre la tabla ALUMNOS:

$$\sigma_{\text{alumnos.nom_alum}=\text{"Luis Martínez Pérez"}}(\text{alumnos})$$

3. Hacer el producto cartesiano de ALU con esta selección e igualar las provincias en ALUMNOS y ALU para después proyectar sobre DNI y nom_alum. Obtenemos:

$$\pi_{\text{alu.DNI, alu.nom_alum}}(\sigma_{\text{alumnos.provincia}=\text{alu.provincia}}(\sigma_{\text{alumnos.nom_alum}=\text{"Luis Martínez Pérez"}}(\text{alumnos}) \times (\text{alu})))$$

I

Los alias serán necesarios cuando haya que distinguir los atributos de una tabla que aparezca varias veces en expresiones complejas de AR.

Union, intersección y diferencia

	A	B	C		A	B	C
$r =$	a_1	b_1	c_1	$s =$	a_1	b_1	c_1
	a_2	b_2	c_2		a_2	b_2	c_2
	a_3	b_1	c_1		a_3	b_2	c_2
	a_4	b_1	c_1		a_4	b_2	c_2
	a_4	b_2	c_2		a_1	b_2	c_2

FIGURA 6.5 Ejemplo de operadores conjuntistas: tablas que intervienen.

El resultado w de hacer $w = r \cup (\cap)(-)s$ se muestra en la Figura 6.6.

	A	B	C		A	B	C		A	B	C
$r \cup s =$	a_1	b_1	c_1	$r \cap s =$	a_1	b_1	c_1	$r - s =$	a_3	b_1	c_1
	a_2	b_2	c_2		a_2	b_2	c_2		a_4	b_1	c_1
	a_3	b_1	c_1						a_4	b_2	c_2
	a_4	b_1	c_1								
	a_4	b_2	c_2								
	a_3	b_2	c_2								
	a_4	b_2	c_2								
	a_1	b_2	c_2								

Para hacer la unión, intersección o diferencia de dos tablas, han de tener el mismo esquema.

División

1. Encontrar los alumnos que están matriculados de **todas** las asignaturas de primer curso.
2. Encontrar las asignaturas en las que dan clase **todos** los profesores del área 'COMPUT' que sean de categoría 'CU'.
3. Encontrar los profesores que dan clase a **todos** los grupos de la asignatura de código 'BDI'.
4. Encontrar las aulas que están ocupadas **todos** los días de la semana.

Antes del todos: dividiendo

Después del todos: divisor

José Escribano

3

WUOLAH

Resumen SQL

Creación de tablas

Sentencia: CREATE TABLE

Tipo de dato	Descripción
INT o INTEGER o NUMERIC	Enteros con signo (su rango depende de la implementación del sistema).
REAL o FLOAT	Datos numéricos en coma flotante.
CHAR (n)	Cadena de longitud fija n.
VARCHAR (n)	Cadena de longitud variable de hasta n caracteres.
VARCHAR2 (n)	De mínimo 1 carácter y máximo 4000. Otra implementación de cadena más eficiente (específico de Oracle).
NUMBER (p, s)	Número con precisión p y escala s, donde precisión indica el número de dígitos, y escala el número de cifras decimales.
LONG	Cadena de caracteres de longitud variable de hasta 2 gigabytes (específico de Oracle).
LONG RAW (size)	Cadena de datos binarios de longitud variable de hasta 2 gigabytes (específico de Oracle).
DATE o TIME o TIMESTAMP	Fecha.

Restricciones con SQL

Clave primaria

PRIMARY KEY

El sistema controla que no se introduzcan valores duplicados para la clave primaria y que no se le asigne un valor nulo a ésta en ninguna de las tuplas de la tabla

Control de unicidad de los valores de una columna

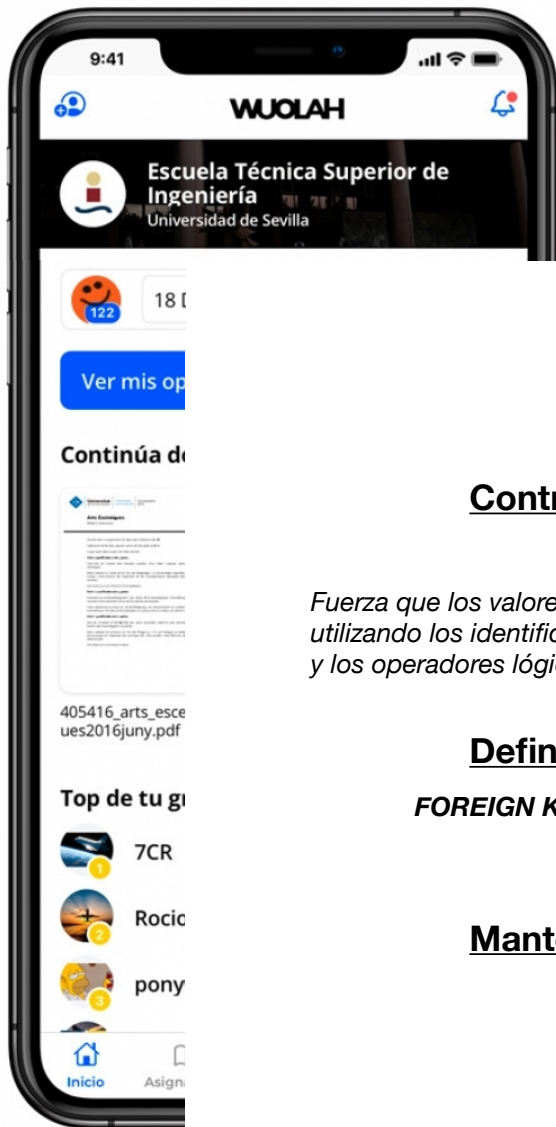
UNIQUE

El atributo o combinación de atributos debe tener un valor único (o nulo) para cada una de las tuplas de la tabla (clave candidata)

Control de valores nulos

NOT NULL

Se evita la aparición de valores nulos en un atributo o combinación de atributos (clave candidata)



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



Control del rango de valores permitidos en columna

CHECK (<condicion>)

Fuerza que los valores de las tuplas cumplan la condición indicada. Puede construirse utilizando los identificadores de los atributos, los operadores relacionales de comparación y los operadores lógicos AND, OR y NOT.

Definición de claves externas

FOREIGN KEY (<nombre_clave>, ...) REFERENCES <nombre_tabla>
(<nombre_clave>, ...)

Mantenimiento de integridad

ON DELETE <SET NULL/CASCADE>

ON UPDATE <SET NULL/CASCADE>

Definición de valores por defecto

DEFAULT <valor>

Por defecto, el sistema rellena el valor de una tupla a *null*. Sin embargo, se puede cambiar con esta clausula.

Eliminación de tablas

DROP TABLE <id_tabla> (CASCADE CONSTRAINTS);

Si usamos CASCADE CONSTRAINTS, eliminamos la tabla aunque hayan restricciones por clave externa. Sin esa clausula, daría error si hay clave externa.

Consultas para una tabla

```
SELECT <id_columna>[{,<id_columna>}]  
FROM <id_tabla>  
[WHERE <condicion>;]
```

- **SELECT:** Contiene las columnas que queremos mostrar.
- **FROM:** Indica la tabla sobre la que queremos consultar.
- **WHERE:** Clausula opcional que impone una condición boleana que deben cumplir las tuplas
 - = (igual a...)

- \neq (distinto a...)
- *BETWEEN* 'x' *AND* 'y' (entre 'x' e 'y')
- *IN* ('x', 'y') (un valor se encuentra entre los elementos de un conjunto)
- *LIKE* (Comparación entre cadena de caracteres)
 - A. '%': Cualquier número de caracteres, cero o más.
 - B. '_': Representa un único caracter
- *IS [NOT] NULL* (Para hallar tuplas con valor nulo o no nulo)

Equivalencia con AR:

$$\pi_{\langle id_columna \rangle, \dots, \langle id_columna \rangle} (\sigma_{\langle condicion \rangle} (\langle id_tabla \rangle))$$

Eliminación de valores duplicados

SELECT DISTINCT $\langle id_columna \rangle$ *FROM* $\langle id_tabla \rangle$

Ordenación de los resultados

SELECT $\langle id_columna \rangle$ *FROM* $\langle id_tabla \rangle$ *WHERE* $\langle condición \rangle$

ORDER BY $\langle id_columna \rangle$ [*ASC*/*DESC*] [, $\langle id_columna \rangle$ [*ASC*/*DESC*]]

- Por defecto, *ORDER BY* ordena de forma ascendente (ASC)

Consultas sobre varias tablas

```
SELECT <id_columna>[{,<id_columna>}]
FROM <id_tabla> [{,<id_tabla>}]
[WHERE <condici\ 'on>];
```

Equivalencia en AR:

$$\pi_{\langle id_columna \rangle, \dots, \langle id_columna \rangle} \sigma_{\langle condicion \rangle} (\langle id_tabla \rangle \times \dots \times \langle id_tabla \rangle)$$

Subconsultas

- En la clausula *WHERE* de una consulta se puede realizar una consulta adicional para recuperar cierta información de la base de datos que es necesaria para solventar la consulta principal.
- Ejemplo: Mostrar el DNI de los alumnos que son de la misma provincia que el alumno 'JUAN LOPEZ'

```
SELECT DNI
FROM alumnos
WHERE provincia = (
                    SELECT provincia
                    FROM alumnos
                    WHERE nom_alum='JUAN LOPEZ');
```

Alias (usar más de una vez la misma tabla)

Mostrar aquellos profesores que tienen la misma categoría que su director de departamento.

```
SELECT p1.NRP
FROM profesores p1
WHERE categoria = (
                  SELECT categoria
                  FROM profesores p2, departamentos
                  WHERE p2.NRP=departamentos.director
                  AND departamento.cod_dep=p1.cod_dep);
```

Operadores para las subconsultas

- *IN* (<subconsulta>) (un valor se encuentra entre los elementos de un conjunto)
- *[NOT] EXISTS* (<subconsulta>) (indica si esta devuelve al menos una tupla, sin importar contenido ni numero de campos de esa tupla)

Subconsultas en la cláusula FROM

Mostrar los datos (nombre y número de créditos) de las asignaturas que se imparten en un aula con capacidad mayor que 100.

```
SELECT nom_asig, creditos
FROM (SELECT DISTINCT cod_asig
      FROM clase, aula
      WHERE clase.cod_aula=aula.cod_aula
      AND capacidad>100) al, asignaturas
WHERE al.cod_asig=asignaturas;
```

Comparación de un valor con los elementos de un conjunto

- Dos casos:

- Si el valor cumple el criterio de comparación con al menos uno de los valores del conjunto (ANY, equivalente a IN):

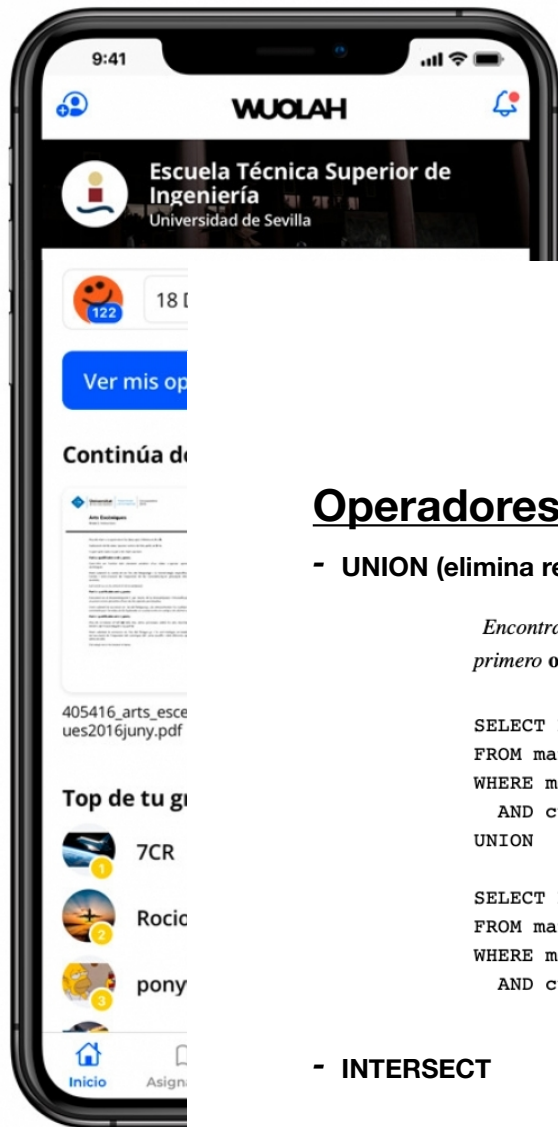
Mostrar el DNI de los alumnos que están matriculados en alguna asignatura de tercero.

```
SELECT cod_al
FROM matriculas
WHERE cod_asig IN
      (SELECT cod_asig
       FROM asignaturas
       WHERE curso=3 );
```

- Si el valor cumple el criterio de comparación con todos y cada uno de los valores del conjunto (ALL)

Mostrar la lista de alumnos con la nota más alta en BDII.

```
SELECT DNI
FROM matriculas
WHERE cod_asig='BDII' AND calificacion >= ALL
      (SELECT calificacion
       FROM matriculas
       WHERE cod_asig='BDII');
```



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



Operadores de conjunto

- UNION (elimina repetidos) | UNION ALL (no elimina repetidos)

Encontrar los alumnos que están matriculados de, al menos, una asignatura de primero o una asignatura de segundo.

```
SELECT DNI
FROM matriculas, asignaturas
WHERE matriculas.cod_asig=asignaturas.cod_asig
      AND curso=1
UNION
```

```
SELECT DNI
FROM matriculas, asignaturas
WHERE matriculas.cod_asig=asignaturas.cod_asig
      AND curso=2;
```

- INTERSECT

Encontrar los alumnos que están matriculados de, al menos, una asignatura de primero y una asignatura de segundo.

```
SELECT DNI
FROM matriculas, asignaturas
WHERE matriculas.cod_asig=asignaturas.cod_asig
      AND curso=1
INTERSECT
```

```
SELECT DNI
FROM matriculas, asignaturas
WHERE matriculas.cod_asig=asignaturas.cod_asig
      AND curso=2;
```

- MINUS

Encontrar los alumnos que están matriculados de, al menos, una asignatura de primero y no lo están de una asignatura de segundo.

```
SELECT DNI
FROM matriculas, asignaturas
WHERE matriculas.cod_asig=asignaturas.cod_asig
      AND curso=1
```

MINUS

```
SELECT DNI
FROM matriculas, asignaturas
WHERE matriculas.cod_asig=asignaturas.cod_asig
      AND curso=2;
```

Funciones de agregación

- **sum()**: Suma de una distribución de valores

Calcular la capacidad total del centro.

```
SELECT sum(capacidad)
FROM aulas;
```

- **min()**: Valor mínimo de una distribución de valores
- **max()**: Valor máximo de una distribución de valores

Mostrar el máximo y el mínimo número de créditos que tienen las asignaturas.

```
SELECT max(creditos), min(creditos)
FROM asignaturas;
```

- **avg()**: Media de una distribución de valores
- **stddev()**: Desviación de una distribución de valores
- **count()**: Cardinal de una distribución de valores.

Contar el número de asignaturas en las que está matriculado el alumno con DNI 44444444.

```
SELECT count(*)
FROM matriculas
WHERE cod_al='44444444';
```

Determinar el número de asignaturas que se imparten los lunes en el aula 0.7.

```
SELECT count(cod_asig)
FROM clases
WHERE cod_aul='0.7' AND dia='LU';
```

Encontrar los alumnos matriculados de más de 15 asignaturas.

```
SELECT DNI
FROM alumnos
WHERE 15 < (SELECT count(*)
            FROM Matricula
            WHERE Alumnos.DNI=Matricula.DNI);
```

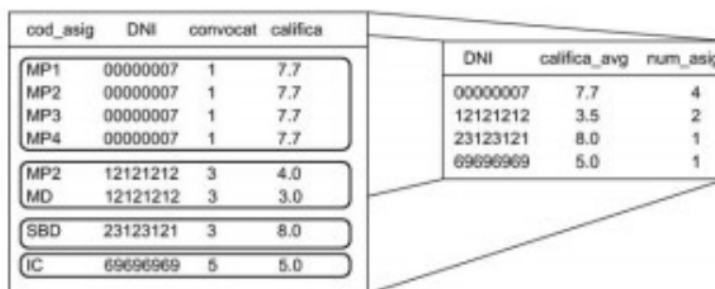
Grupos de tuplas

GROUP BY <atributo>

- Se indica el atributo o conjunto de atributos por cuyos valores queremos agrupar las tuplas, para poder aplicar funciones de agregación sobre cada uno de los distintos grupos.

Mostrar, para cada alumno, el número de asignaturas de las que está matriculado y la media de las calificaciones.

```
SELECT DNI, avg(calificacion) califica_avg, count(*) num_asig
FROM matriculas
GROUP BY DNI;
```



cod_asig	DNI	convocat	califica
MP1	00000007	1	7.7
MP2	00000007	1	7.7
MP3	00000007	1	7.7
MP4	00000007	1	7.7
MP2	12121212	3	4.0
MD	12121212	3	3.0
SBD	23123121	3	8.0
IC	69696969	5	5.0

DNI	califica_avg	num_asig
00000007	7.7	4
12121212	3.5	2
23123121	8.0	1
69696969	5.0	1

- Se pueden imponer condiciones sobre los valores obtenidos tras las funciones de agregación. Para ello, se usa:

HAVING

- Usa condiciones similares a la clausula WHERE, pero se aplican a las tuplas que resultan de la agrupación.

Mostrar, para cada alumno, los créditos totales de los que está matriculado siempre que lo esté de más de 100 créditos.

```
SELECT DNI, sum(creditos)
FROM matriculas, asignaturas
WHERE asignaturas.cod_asig=matriculas.cod_asig
GROUP BY DNI
HAVING sum(creditos)>=100;
```

Mostrar la(s) asignatura(s) con el mayor número de aprobados.

```
SELECT cod_asig
FROM matriculas
WHERE calificacion>=5
GROUP BY cod_asig
HAVING count(*)>= ALL (SELECT count(*)
                        FROM matriculas
                        WHERE calificacion>=5
                        GROUP BY cod_asig);
```

División

- Partimos de la resolución de AR:

· *Mostrar los DNIs de aquellos alumnos que estén matriculados en todas las asignaturas obligatorias de segundo curso.*

La solución en Álgebra Relacional se expresa mediante:

$$\pi_{DNI, cod_asig}(Matriculas) \div \pi_{cod_asig}(\sigma_{caracter='ob' \wedge curso=2}(Asignaturas))$$

- El conjunto representado por el divisor debe estar contenido en el conjunto de valores con los que esta relacionada en el dividendo la tupla candidata a formar parte del resultado.

1. Hallamos el divisor

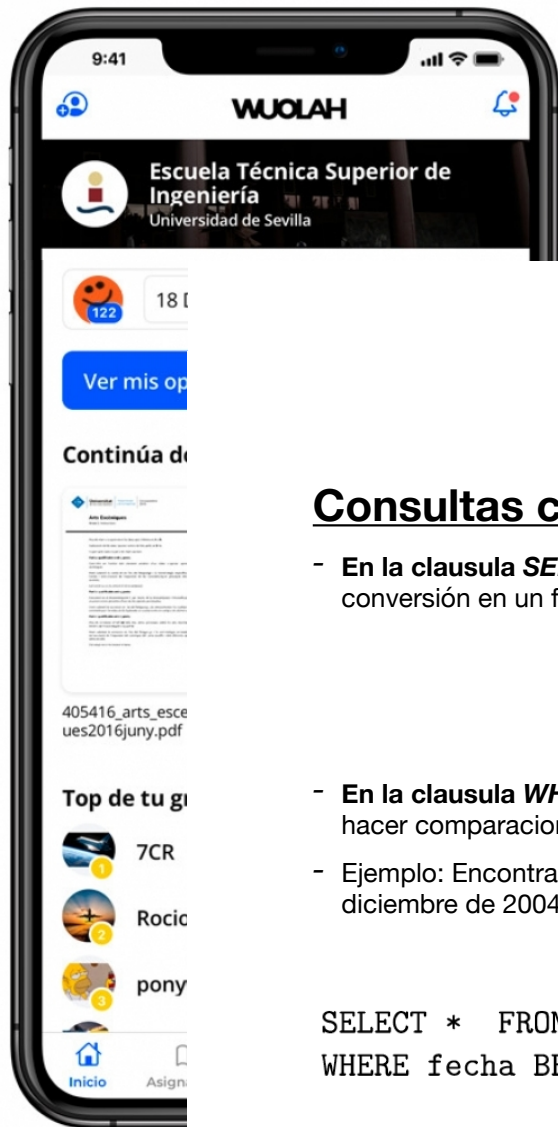
```
SELECT cod_asig
FROM asignaturas
WHERE caracter='ob' and curso=2;
```

2. Hallamos el dividendo

```
SELECT DNI
FROM alumnos
WHERE ?;
```

3. Juntamos:

```
SELECT DNI
FROM alumnos
WHERE NOT EXISTS (
    SELECT cod_asig
    FROM asignaturas
    WHERE caracter='ob' and curso=2
    MINUS
    SELECT cod_asig
    FROM matriculas
    WHERE matriculas.DNI=alumnos.DNI);
```



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



Consultas con el tipo DATE

- En la **clausula SELECT**, usamos la función de conversión **TO_CHAR()** para la conversión en un formato determinado por los parámetros.

```
SELECT TO_CHAR(fecha, 'DD-MM-YYYY')  
FROM ventas;
```

- En la **clausula WHERE**, hacemos uso de la función de conversión **TO_DATE()** para hacer comparaciones entre fechas en formato interno.
- Ejemplo: Encontrar las ventas realizadas entre el 1 de enero de 2002 y el 31 de diciembre de 2004

```
SELECT * FROM ventas  
WHERE fecha BETWEEN TO_DATE('01/01/2002', 'DD/MM/YYYY') AND  
TO_DATE('31/12/2004', 'DD/MM/YYYY');
```

Inserción de datos de tuplas

```
INSERT INTO <id_tabla> VALUES (<valor1>, ..., <valorN>);
```

Insertar un nuevo alumno en la base de datos con los siguientes datos:

- DNI:44444444
- Nombre: Carmen Garrido
- Fecha Nacimiento: 28-09-1975
- Dirección: Avenida Príncipe Felipe
- Provincia: Granada
- Beca: NO

```
INSERT INTO alumnos VALUES  
( '44444444', 'Carmen Garrido', '28-09-1975',  
'Avenida Principe Felipe', 'Granada', 'NO');
```

- Se puede alterar el orden en el que se suministran los valores de los atributos de la tupla que se desea insertar mediante esta variante:

```
INSERT INTO <id_tabla>(<id_columnal>,...,<id_columnaN>)
VALUES (<valor1>, ..., <valorN>);
```

```
INSERT INTO alumnos
(DNI,fecha_nac,direccion,nom_alum,beca,provincia)
VALUES('44444444','28-09-1975', 'Avenida Principe Felipe',
'Carmen Garrido', 'NO','Granada');
```

- También hay otra variante que permite insertar en una tabla las tuplas resultantes de realizar una consulta sobre la base de datos:

```
INSERT INTO <id_tabla>[(<id_columnal>,...
,<id_columnaN>)] consulta;
```

```
INSERT INTO alumnos_BDII
SELECT *
FROM alumnos
WHERE DNI IN (SELECT DNI
FROM matriculas
WHERE cod_asig='BDII');
```

Borrado de datos de tuplas

```
DELETE [FROM] <id_tabla> [WHERE condici\ 'on]
```

- Si no se especifica la condición, se borran todas las tuplas de la tabla.

Actualización de datos de tuplas

```
UPDATE <id_tabla>
SET <id_columna>=<expresion> [{,<id_columna>=<expresion>}]
[WHERE condicion];
```

- Cambia el contenido de las columnas que aparecen en la clausula *SET* asignando como nuevo el valor resultado de evaluar las expresiones indicadas.

Cambiar el nombre del alumno con DNI=44444444 por el de 'JUAN LOPEZ'.

```
UPDATE alumnos SET nombre='JUAN LOPEZ' WHERE DNI='44444444';
```

Subir un punto la nota del alumno de DNI '44444444' en la asignatura de código BDII y aumentarle una convocatoria.

```
UPDATE matriculas
SET calificacion=calificacion+1,
    convocatoria=convocatoria+1
WHERE DNI='44444444';
```

Vistas de usuario

- Consultas a las que se asocia un nombre y que puede ser utilizado posteriormente como si de una tabla almacenada físicamente se tratara. Ayuda a simplificar consultas y a mantener la seguridad en el sistema (evitando mostrar atributos “privados”)
- Ejemplo: Crea una vista que contenga la cantidad media de cada pieza que se ha vendido en el año 2019.

```
create view media_2018 as
select codpie, avg(cantidad) media from ventas
where to_char(fecha,'yyyy')='2018'
group by codpie;
```

A partir de ahora, podemos usar `media-2018` en cualquier consulta. Por ejemplo,

```
select * from media_2018;
```

Modificación del esquema de una tabla

ALTER TABLE <nombre_tabla> <modificador>

- Para añadir un atributo, sustituimos *modificador* por:

```
ADD (atributo [tipo] [DEFAULT expresion] [restriccion_atributo]);
```

- Para añadir una restricción, sustituimos *modificador* por:

```
ADD CONSTRAINT nombre_restriccion [[UNIQUE | PRIMARY KEY] (lista_columnas) |  
FOREING KEY (lista_columnas) REFERENCES tabla(lista_col) |  
CHECK (condicion)];
```

- Para eliminar una restricción, sustituimos *modificador* por:

```
DROP CONSTRAINT nombre_restriccion [CASCADE];
```