Programación Frontend y Backend

BLOQUE BBDD

Modelo Físico

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO
EOI Escuela de organización industrial
Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

01 Introducción

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO
EOI Escuela de organización industrial

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

SISTEMA NACIONAL DE GARANTÍA JUVENIL

Proceso de Diseño de BD

Fase 1:

Análisis de requisitos

Recabar información sobre el uso que se piensa dar a la base de datos.

Fase 2:

Diseño conceptual (modelo E/R)

Creación de un esquema conceptual de la base de datos independiente del DBMS.

GOBIERNO DE ESPAÑA · MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO · EOI Escuela de organización industrial
Unión Europea · Fondo Social Europeo · Iniciativa de Empleo Juvenil · El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

Proceso de Diseño de BD

Fase 3:

Elección del sistema gestor de bases de datos

Elección del modelo de datos (p.ej. relacional, OO)

Fase 4:

Diseño lógico

Creación del esquema conceptual para el modelo de datos del DBMS elegido, paso del modelo E/R al modelo relacional.

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO EOI Escuela de organización industrial
Unión Europea Fondo Social Europeo Iniciativa de Empleo Juvenil El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

Proceso de Diseño de BD

Fase 5:

Diseño físico

Creación de la base de datos utilizando el DDL (lenguaje de definición de datos).

Fase 6:

Uso y mantenimiento

Gestión de los datos utilizando el DML (lenguaje de manipulación de datos).

GOBIERNO DE ESPAÑA · MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO · EOI Escuela de organización industrial
Unión Europea · Fondo Social Europeo · Iniciativa de Empleo Juvenil · El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

Introducción

- Structured Query Language (SQL)
- Lenguaje **declarativo** de acceso a bases de datos.
- Combina construcciones del álgebra relacional y el cálculo relacional.
- Lenguaje estándar *de facto* en los SGBD comerciales
- Tiene una estructura “pseudo inglésa”

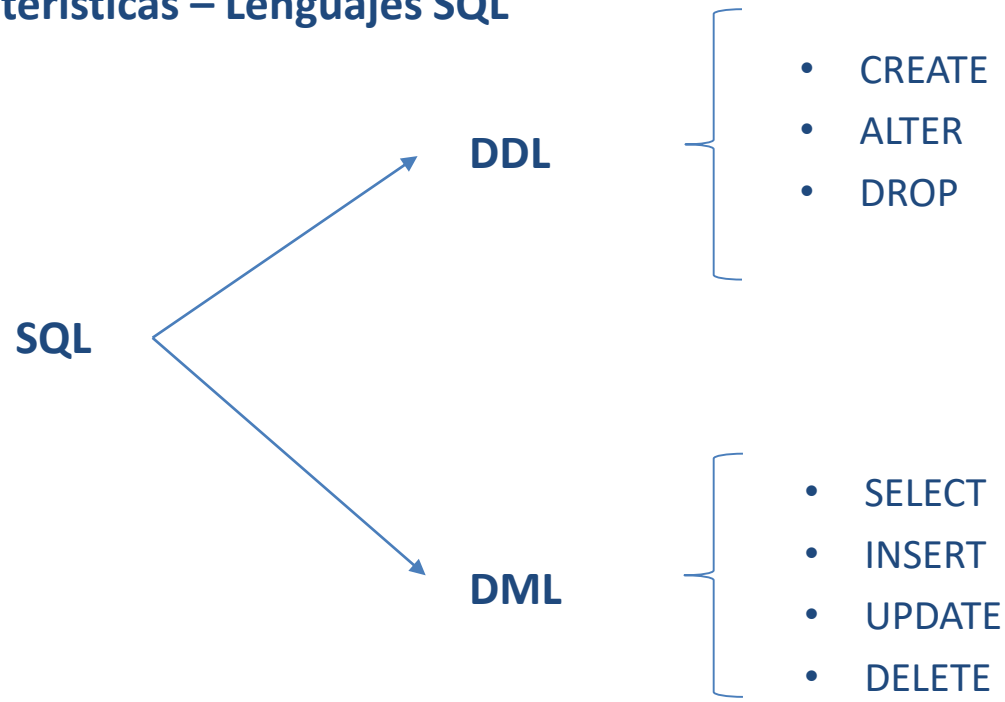
GOBIERNO DE ESPAÑA · MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO · EOI Escuela de organización industrial
Unión Europea · Fondo Social Europeo · Iniciativa de Empleo Juvenil · El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

Características

- El Lenguaje de Definición de Datos (DDL)
  - Proporciona comandos para la creación, borrado y modificación de esquemas relacionales

- El Lenguaje de Manipulación de Datos (DML)
  - Basado en el álgebra relacional y el cálculo relacional permite realizar consultas y adicionalmente insertar, borrar y actualizar de tuplas
  - Ejecutado en una consola interactiva
  - Embebido dentro de un lenguaje de programación de propósito general

Características – Lenguajes SQL



02 DDL

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO
EOI Escuela de organización industrial

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

SISTEMA NACIONAL DE GARANTÍA JUVENIL

Creación de tablas

La creación de tablas se lleva a cabo con la sentencia **CREATE TABLE**.

Ejemplo: creación del siguiente esquema de BD.

CLIENTES (DNI, NOMBRE, DIR)

SUCURSALES (NSUC, CIUDAD)

CUENTAS (COD, DNI, NSUCURS, SALDO)

:

Creación de tablas - Ejemplo

CLIENTES (DNI, NOMBRE, DIR)

SUCURSALES (NSUC, CIUDAD)

CUENTAS (COD, DNI, NSUCURS, SALDO)

```
CREATE TABLE CLIENTES (
        DNI VARCHAR(9) NOT NULL,
        NOMBRE VARCHAR(20),
        DIR  VARCHAR(30),
        PRIMARY KEY (DNI)
);

CREATE TABLE SUCURSALES (
        NSUC VARCHAR(4) NOT NULL,
        CIUDAD  VARCHAR(30),
        PRIMARY KEY (NSUC)
);

CREATE TABLE CUENTAS (
  COD VARCHAR(4) NOT NULL,
  DNI VARCHAR(9) NOT NULL,
  NSUCURS VARCHAR(4) NOT NULL,
  SALDO INT DEFAULT 0,
  PRIMARY KEY (COD, DNI, NSUCURS),
  FOREIGN KEY (DNI) REFERENCES CLIENTES (DNI),
  FOREIGN KEY (NSUCURS) REFERENCES SUCURSALES
(NSUC) );
```

Eliminación de tablas

La eliminación de tablas se lleva a cabo con la sentencia **DROP TABLE**.

Ejemplo: Eliminación del siguiente esquema de BD.

```
DROP TABLE CUENTAS;
DROP TABLE SUCURSALES ;
DROP TABLE CLIENTES;
```

Modificación de tablas

La modificación de tablas se lleva a cabo con la sentencia **ALTER TABLE**.

Añadir campo nuevo

ALTER TABLE CLIENTES **ADD** PAIS **VARCHAR**(10);

Modificar un campo

ALTER TABLE CLIENTES **MODIFY** PAIS **VARCHAR**(20);

Eliminación de un campo

ALTER TABLE CLIENTES **DROP** PAIS;

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO EOI Escuela de organización industrial
Unión Europea Fondo Social Europeo Iniciativa de Empleo Juvenil El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

Modificación de tablas

MySQL:

```
ALTER TABLE Customer DROP INDEX Con_First;
```

Note that MySQL uses DROP INDEX for index-type constraints such as **UNIQUE**.

Oracle:

```
ALTER TABLE Customer DROP CONSTRAINT Con_First;
```

SQL Server:

```
ALTER TABLE Customer DROP CONSTRAINT Con_First;
```

03

DML

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO
EOI Escuela de organización industrial

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

SISTEMA NACIONAL DE GARANTÍA JUVENIL

Tipo	Ejemplo
BIGINT	8589934592
INTEGER	186282
SMALLINT	186
NUMERIC(8,2)	999999.99 (precisión, escala)
DECIMAL(8,2)	999999.99 (precisión, escala)
REAL	6.02257E23
DOUBLE PRECISION	3.141592653589
FLOAT	6.02257E23
CHARACTER(*max*)	'GREECE ' (15 caracteres)
VARCHAR(*n*)	'hola'
DATE	date 'YYYY-MM-DD'
TIME	time 'hh:mm:ss.ccc'
TIMESTAMP	timestamp 'YYYY-MM-DD hh:mm:ss.ccc'

Modificación de la BBDD

Las instrucciones SQL que permiten modificar el estado de la BBDD son:

INSERT : Añade filas a una tabla

UPDATE : Actualiza filas de una tabla

DELETE : Elimina filas de una tabla

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO
EOI Escuela de organización industrial
Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

La instrucción INSERT

La **inserción** de tuplas se realiza con la sentencia **INSERT**,

Es posible insertar directamente valores.

O bien insertar el conjunto de resultados de una consulta.

En cualquier caso, los valores que se insertan deben pertenecer al dominio de cada uno de los atributos de la relación.

Ejemplos: CLIENTES (DNI, NOMBRE, DIR)

La inserción

```
INSERT INTO CLIENTES VALUES (1111,'Mario',
'C/. Mayor, 3');
```

Es equivalente a las siguientes sentencias

```
INSERT INTO CLIENTES (NOMBRE,DIR,DNI)
VALUES ('Mario','C/. Mayor, 3',1111);
```

```
INSERT INTO CLIENTES (DNI,DIR,NOMBRE)
VALUES (1111,'C/. Mayor, 3','Mario');
```

La instrucción UPDATE

La **modificación** de tuplas se realiza con la sentencia **UPDATE**,

Es posible elegir el conjunto de tuplas que se van a actualizar usando la clausula **WHERE**.

Ejemplos: CUENTAS (COD, DNI, NSUCURS, SALDO)

Suma del 5% de interés a los saldos de todas las cuentas.

```
UPDATE CUENTAS SET SALDO = SALDO * 1.05;
```

Suma del 1% de bonificación a aquellas cuentas cuyo saldo sea superior a 100.000 €.

```
UPDATE CUENTAS SET SALDO = SALDO * 1.01
WHERE SALDO > 100000;
```

Modificación de DNI y saldo simultáneamente para el código 898.

```
UPDATE CUENTAS SET DNI='555', SALDO=10000
WHERE COD LIKE '898';
```

La instrucción DELETE

La **eliminación** de tuplas se realiza con la sentencia **DELETE**:

```
DELETE FROM R WHERE P; -- WHERE es opcional
```

Elimina tuplas completas, no columnas. Puede incluir subconsultas.

Ejemplos: para la BD de CLIENTES, CUENTAS, SUCURSALES.

Eliminar todas cuentas con código entre 1000 y 1100.

```
DELETE FROM CUENTAS WHERE COD BETWEEN 1000 AND 1100;
```

La instrucción `SELECT`

Consta de tres cláusulas:

SELECT

La lista de los atributos que se incluirán en el resultado de una consulta.

FROM

Especifica las relaciones que se van a usar como origen en el proceso de la consulta.

WHERE

Especifica la condición de filtro sobre las tuplas en términos de los atributos de las relaciones de la cláusula **FROM**.

GOBIERNO DE ESPAÑA · MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO · EOI Escuela de organización industrial
Unión Europea · Fondo Social Europeo · Iniciativa de Empleo Juvenil · El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

La instrucción `SELECT`

Ejemplo: CLIENTES (DNI, NOMBRE, DIR)

Devuelve una lista de todos los clientes.

SELECT * FROM CLIENTES;

Devuelve los nombres de todos los clientes.

SELECT NOMBRE FROM CLIENTES;

Devuelve los DNI de aquellos clientes cuyo nombre sea JUAN.

SELECT DNI FROM CLIENTES WHERE NOMBRE LIKE 'JUAN';

	A	B	C	D	E	F	G	H	I
1	Product Sales								
2									
3	**Date**	**Region**	**Product**	**Qty**	**Cost**	**Amt**	**Tax**	**Total**	
4	1-Apr	East	Paper	73	12.95	945.35	66.17	1,011.52	
5	1-Apr	West	Paper	33	12.95	427.35	29.91	457.26	
6	2-Apr	East	Pens	14	2.19	30.66	2.15	32.81	
7	2-Apr	West	Pens	40	2.19	87.60	6.13	93.73	
8	3-Apr	East	Paper	21	12.95	271.95	19.04	290.99	
9	3-Apr	West	Paper	10	12.95	129.50	9.07	138.57	
10									
11									

Ejercicios Guiados

- Lista de todos los productos
- Lista de todas las regiones en las que se ha vendido papel.
- Lista de todos los productos de los que hayamos vendido 40 unidades.
- Lista de productos y cantidad de ellos que se han vendido en la región West.
- Lista de fechas en las que se han vendido 15 productos o más.
- Lista de fechas, cantidad y total de ventas de bolígrafos (pen).
- Lista de productos que se han vendido en la región West en una cantidad igual o inferior a 20.
- Lista de productos ordenados por cantidad.

Salida ordenada de los datos

- SQL permite controlar el orden en el que se presentan las tuplas de una relación mediante la cláusula ORDER BY
- La ordenación se realiza tras haber ejecutado la consulta sobre las tuplas resultantes, puede convertirse en una operación costosa dependiendo del tamaño de la relación resultante.

SELECT * FROM CLIENTES ORDER BY NOMBRE ASC;

SELECT * FROM CLIENTES ORDER BY NOMBRE DESC;

SELECT * FROM CLIENTES ORDER BY NOMBRE ASC, DNI DESC;

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO EOI Escuela de organización industrial
Unión Europea Fondo Social Europeo Iniciativa de Empleo Juvenil El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

04 Expresiones SQL

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO
EOI Escuela de organización industrial

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

SISTEMA NACIONAL DE GARANTÍA JUVENIL

Estructura completa de la sentencia SELECT

SELECT A1, …, An

- Describe la salida deseada con:
  - Nombres de columnas
  - Expresiones aritméticas
  - Literales
  - Funciones escalares
  - Funciones de columna

FROM T1, …, Tn
WHERE P

- Nombres de las tablas / vistas
- Condiciones de selección de filas

GROUP BY Ai1, …, Ain
HAVING Q

- Nombre de las columnas
- Condiciones de selección de grupo

ORDER BY Aj1, …, Ajn

- Nombres de columnas

GOBIERNO DE ESPAÑA · MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO · EOI Escuela de organización industrial
Unión Europea · Fondo Social Europeo · Iniciativa de Empleo Juvenil · El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

SQL permite duplicados en el resultado, para eliminar las tuplas repetidas se utiliza la cláusula **DISTINCT**. También es posible pedir explícitamente la inclusión de filas repetidas mediante el uso de la cláusula **ALL**.

```
SELECT ALL ADMRDEPT
FROM DEPARTMENT
```

ADMRDEPT

A00
A00
A00
A00
D01
D01
A00
E01

```
SELECT DISTINCT ADMRDEPT
FROM DEPARTMENT
```

ADMRDEPT

A00
D01
E01

La cláusula **WHERE** permite filtrar las filas de la relación resultante.

El predicado de la cláusula WHERE puede ser simple o complejo

- Se utilizan los conectores lógicos **AND** (conjunción), **OR** (disyunción) y **NOT** (negación)

Las expresiones pueden contener

- Predicados de comparación
- BETWEEN / NOT BETWEEN
- IN / NOT IN (con y sin subconsultas)
- LIKE / NOT LIKE
- IS NULL / IS NOT NULL
- ALL, SOME/ANY (subconsultas)
- EXISTS (subconsultas)

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO EOI Escuela de organización industrial
Unión Europea Fondo Social Europeo Iniciativa de Empleo Juvenil El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

WHERE

- Predicados de comparación
    - Operadores: **=**, <> (es el ≠), <, <=, >=. >

- **BETWEEN Op1 AND Op2**
    - Es el operador de comparación para intervalos de valores o fechas.

- **IN** es el operador que permite comprobar si un valor se encuentra en un conjunto.
    - Puede especificarse un conjunto de valores **(Val1, Val2, …)**
    - Puede utilizarse el resultado de otra consulta **SELECT**.

WHERE

- **LIKE** es el operador de comparación de cadenas de caracteres.
  - SQL distingue entre mayúsculas y minúsculas
  - Las cadenas de caracteres se incluyen entre comillas simples
  - SQL permite definir patrones a través de los siguientes caracteres:
    - '%', que es equivalente a "cualquier subcadena de caracteres“
    - '_', que es equivalente a "cualquier carácter“

- **IS NULL** es el operador de comparación de valores nulos.

Múltiples condiciones AND/OR

- Necesito el número de empleado, el trabajo y el nivel de formación de los analistas con un nivel de educación 16

```
SELECT EMPNO, JOB, EDLEVEL
FROM EMPLOYEE
WHERE JOB='ANALYST' AND EDLEVEL=16
```

EMPNO	JOB	EDLEVEL
000130	ANALYST	16

```
SELECT EMPNO, JOB, EDLEVEL
FROM EMPLOYEE
WHERE (JOB='ANALYST' AND EDLEVEL=16)
      OR EDLEVEL=18
ORDER BY JOB, EDLEVEL
```

EMPNO	JOB	EDLEVEL
000130	ANALYST	16
000140	ANALYST	18
000220	DESIGNER	18
000020	MANAGER	18
000010	PRES	18

SELECT con BETWEEN

Obtener el número de empleado y el nivel de todos los empleados con un nivel entre 12 y 15

```
SELECT EMPNO, EDLEVEL
FROM EMPLOYEE
WHERE EDLEVEL BETWEEN 12 AND 15
ORDER BY EDLEVEL
```

→

EMPNO	EDLEVEL
000310	12
000290	12
000300	14
000330	14
000100	14
000230	14
000120	14
000270	15
000250	15

SELECT con IN

Obtener el número de empleado y el nivel de todos los empleados con un nivel de 12 o 15

```
SELECT EMPNO, EDLEVEL
FROM EMPLOYEE
WHERE EDLEVEL IN (12,15)
ORDER BY EDLEVEL
```


EMPNO	EDLEVEL
000310	12
000290	12
000300	14
000330	14
000100	14
000230	14
000120	14
000270	15
000250	15

Búsqueda parcial - LIKE

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE LASTNAME LIKE 'T%';
```

→ THOMPSON

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE LASTNAME LIKE '%SON';
```

→ THOMPSON, JEFFERSON, JOHNSON

```
SELECT LASTNAME
FROM EMPLOYEE
WHERE LASTNAME LIKE '_O%';
```

→ JOHNSON

Búsqueda parcial – NOT LIKE

```
SELECT DEPTNO, DEPTNAME
FROM DEPARTMENT
WHERE DEPTNO NOT LIKE 'D%'
```

DEPTNO	DEPTNAME
A00	SPIFFY COMPUTER SERVICE DIV.
B01	PLANNING
C01	INFORMATION CENTER
E01	SUPPORT SERVICES
E11	OPERATIONS
E21	SOFTWARE SUPPORT

Expresiones

Aunque SQL no es un lenguaje de programación de propósito general, permite definir expresiones calculadas, estas expresiones pueden contener

- Referencias a columnas
- Valores literales
- Operadores aritméticos
- Llamadas a funciones

Los operadores aritméticos son los habituales: +, -, * y /

- Estos operadores sólo funcionan con valores numéricos.
- Los operadores '+' y '–' habitualmente funcionan para fechas.

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO EOI Escuela de organización industrial
Unión Europea Fondo Social Europeo Iniciativa de Empleo Juvenil El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

Expresiones - Matemáticas

Aunque las normas SQL definen un conjunto mínimo de funciones, los SGBD proporcionan una gran variedad, es necesario consultar el manual del SGBD particular.

Descripción	IBM DB2	SQL Server	Oracle	MySQL
Valor absoluto	ABSs	ABS	ABS	ABS
Menor entero >= valor	CEIL	CEILING	CEIL	CEILING
Menor entero <= valor	FLOOR	FLOOR	FLOOR	FLOOR
Potencia	POWER	POWER	POWER	POWER
Redondeo a un número de cifras decimales	ROUND	ROUND	ROUND	ROUND
Módulo	MOD.	%	MOD.	%

Expresiones – Cadenas de texto

Aunque las normas SQL definen un conjunto mínimo de funciones, los SGBD proporcionan una gran variedad, es necesario consultar el manual del SGBD particular.

Descripción	IBM DB2	SQL Server	Oracle	MySQL
Convierte todos los caracteres a minúsculas	LOWER	LOWER	LOWER	LOWER
Convierte todos los caracteres a mayúsculas	UPPER	UPPER	UPPER	UPPER
Elimina los blancos del final de la cadena	RTRIM	RTRIM	RTRIM	RTRIM
Elimina los blancos del comienzo de la cadena	LTRIM	LTRIM	LTRIM	LTRIM
Devuelve una subcadena	SUBSTR	SUBSTRING	SUBSTR	SUBSTRING
Concatena dos cadenas	CONCAT	+	CONCAT	CONCAT

Expresiones de Columna

Las funciones de columna o funciones de agregación son funciones que toman una colección (conjunto o multiconjunto) de valores de entrada y devuelve un solo valor.

Las funciones de columna disponibles son: **AVG, MIN, MAX, SUM, COUNT**.

Los datos de entrada para **SUM y AVG** deben ser una colección de números, pero el resto de operadores pueden operar sobre colecciones de datos de tipo no numérico

Por defecto las funciones se aplican a todas las tuplas resultantes de la consulta. Podemos agrupar las tuplas resultantes para poder aplicar las funciones de columna a grupos específicos utilizando la cláusula **GROUP BY**.

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO EOI Escuela de organización industrial
Unión Europea Fondo Social Europeo Iniciativa de Empleo Juvenil El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

Expresiones de Columna

En la cláusula **SELECT** de consultas que utilizan funciones de columna solamente pueden aparecer funciones de columna.

En caso de utilizar **GROUP BY**, también pueden aparecer columnas utilizadas en la agrupación.

Adicionalmente se pueden aplicar condiciones sobre los grupos utilizando la cláusula **HAVING**

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO EOI Escuela de organización industrial
Unión Europea Fondo Social Europeo Iniciativa de Empleo Juvenil El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

GROUP BY

```
SELECT    WORKDEPT, SALARY
FROM      EMPLOYEE
WHERE     WORKDEPT IN ('A00', 'B01', 'C01')
ORDER BY  WORKDEPT
```

WORKDEPT	SALARY
A00	52750.00
A00	46500.00
A00	29250.00
B01	41250.00
C01	38250.00
C01	23800.00
C01	28420.00

```
SELECT    WORKDEPT, SUM(SALARY) AS SUM
FROM      EMPLOYEE
WHERE     WORKDEPT IN ('A00', 'B01', 'C01')
GROUP BY  WORKDEPT
ORDER BY  WORKDEPT
```

WORKDEPT	SUM
A00	128500.00
B01	41250.00
C01	90470.00

05

Query Multitable

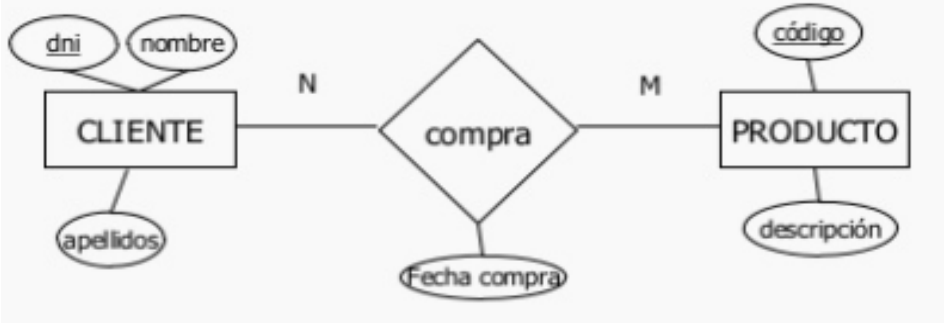
GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO
EOI Escuela de organización industrial

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

SISTEMA NACIONAL DE GARANTÍA JUVENIL

SQL JOIN

Una de las cosas más importante a la hora de poder extraer información de nuestras bases de datos son las consultas y más aún las consultas de varias tablas.



Teniendo en cuenta el ejemplo anterior, para obtener clientes que han comprado en una determinada fecha, o un determinado producto se necesita realizar una consulta en la que interviene más de una sola tabla. Si por el contrario quiero conocer los productos que ha comprado un determinado cliente, ocurre exactamente lo mismo.

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO EOI Escuela de organización industrial
Unión Europea Fondo Social Europeo Iniciativa de Empleo Juvenil El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL

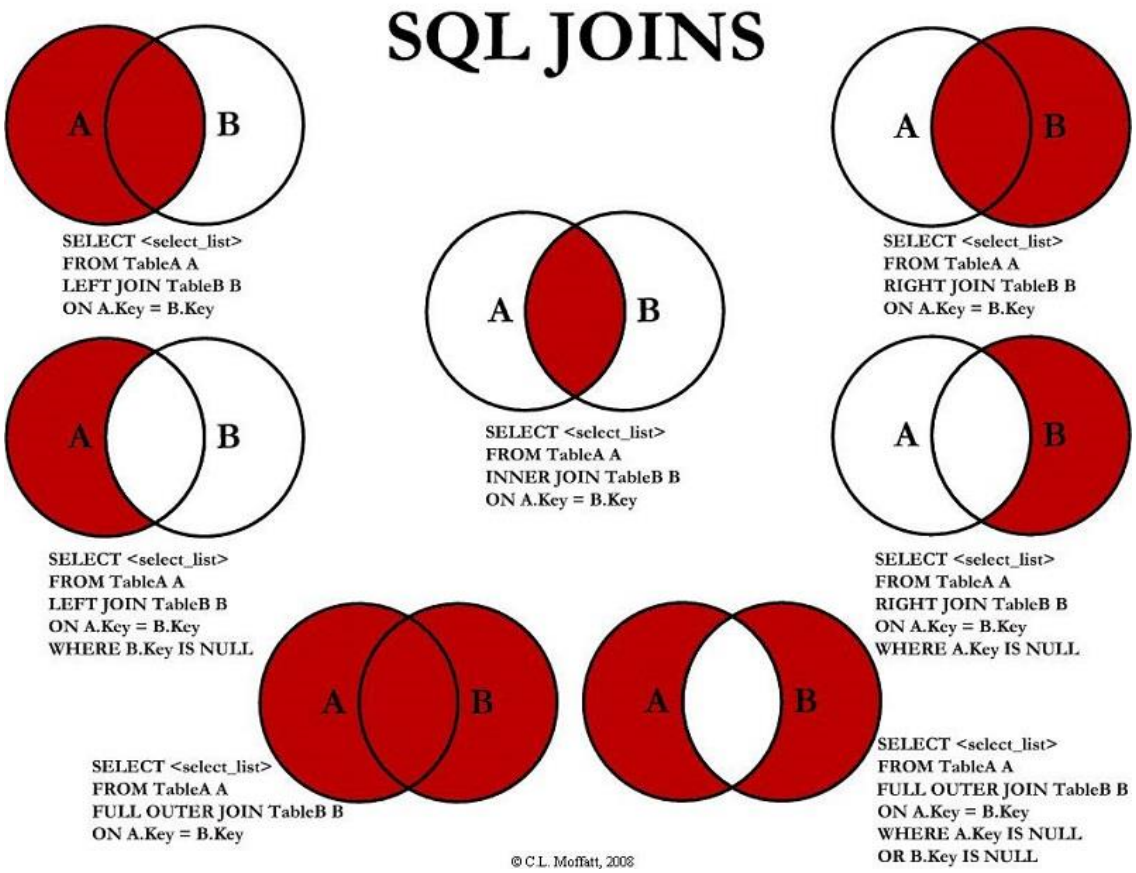
SQL JOIN

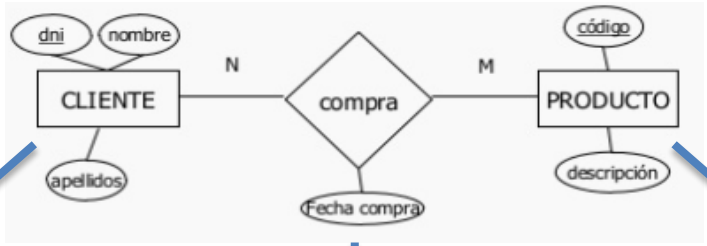
Existen tres mecanismos básicos que nos permiten unir tablas en una consulta:

- **LEFT JOIN**
- **RIGHT JOIN**
- **INNER JOIN**

https://sql-joins.leopard.in.ua/

GOBIERNO DE ESPAÑA MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO EOI Escuela de organización industrial
Unión Europea Fondo Social Europeo Iniciativa de Empleo Juvenil El FSE invierte en tu futuro
SISTEMA NACIONAL DE GARANTÍA JUVENIL


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
© C.L. Moffatt, 2008



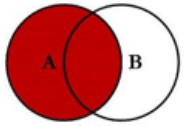
DNI	NOMBRE
0001	JUAN
0002	PEPE
0003	LAURA

DNI	FECHA	CODPROD
0001	03/05/2019	1
0001	03/05/2019	2
0003	04/05/2019	3
0003	05/05/2019	2

CODIGO	NOMBRE
1	TOMATES
2	LENTEJAS
3	CHOCOLATE
4	LECHE

LEFT/RIGHT JOIN

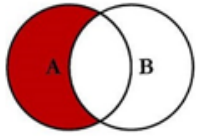
Imaginemos que queremos obtener una lista de todos los clientes junto al número de pedidos que ha realizado:




```
SELECT CLI.DNI, MIN(FECHA) AS PRIMERAVISITA FROM CLIENTE CLI
LEFT JOIN COMPRA COM ON CLI.DNI = COM.DNI
GROUP BY CLI.DNI;
```

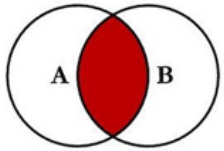
DNI	PRIMERAVISITA
0001	03/05/2019
0002	NULL
0003	04/05/2019

Imaginemos que queremos obtener una lista de todos los clientes que no han realizado un pedido todavía:




```
SELECT CLI.DNI, MIN(FECHA) AS PRIMERAVISITA FROM CLIENTE CLI
LEFT JOIN COMPRA COM ON CLI.DNI = COM.DNI
WHERE COM.DNI IS NULL
GROUP BY CLI.DNI;
```

DNI	PRIMERAVISITA
0002	NULL



INNER JOIN

Imaginemos que queremos obtener una lista de todos los clientes que han realizado pedidos, junto al número de pedidos que ha realizado:

```
SELECT CLI.DNI, COUNT(*) AS NUMPEDIDOS FROM CLIENTE CLI
INNER JOIN COMPRA COM ON CLI.DNI = COM.DNI
GROUP BY DNI;
```

DNI	NUMPEDIDOS
0001	2
0003	2

Imaginemos que queremos obtener una lista de todos los clientes que han realizado pedidos, junto a los productos que han comprado:

```
SELECT CLI.DNI, PROD.NOMBRE AS PRODUCTO FROM CLIENTE CLI
INNER JOIN COMPRA COM ON CLI.DNI = COM.DNI;
INNER JOIN PRODUCTO PROD ON COM.CODPROD = PROD.CODIGO;
```

DNI	PRODUCTO
0001	TOMATES
0001	LENTEJAS
0003	CHOCOLATE
0003	LENTEJAS

06 Ejercicios

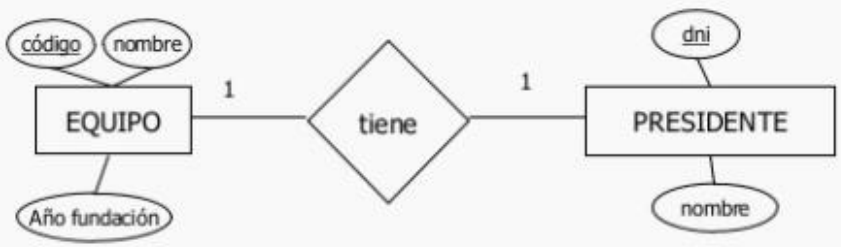
GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, COMERCIO Y TURISMO
EOI Escuela de organización industrial

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

SISTEMA NACIONAL DE GARANTÍA JUVENIL

Ejercicios DDL - EJERCICIO 1 FUTBOL

- Pasad al Modelo Físico el siguiente ejercicio:

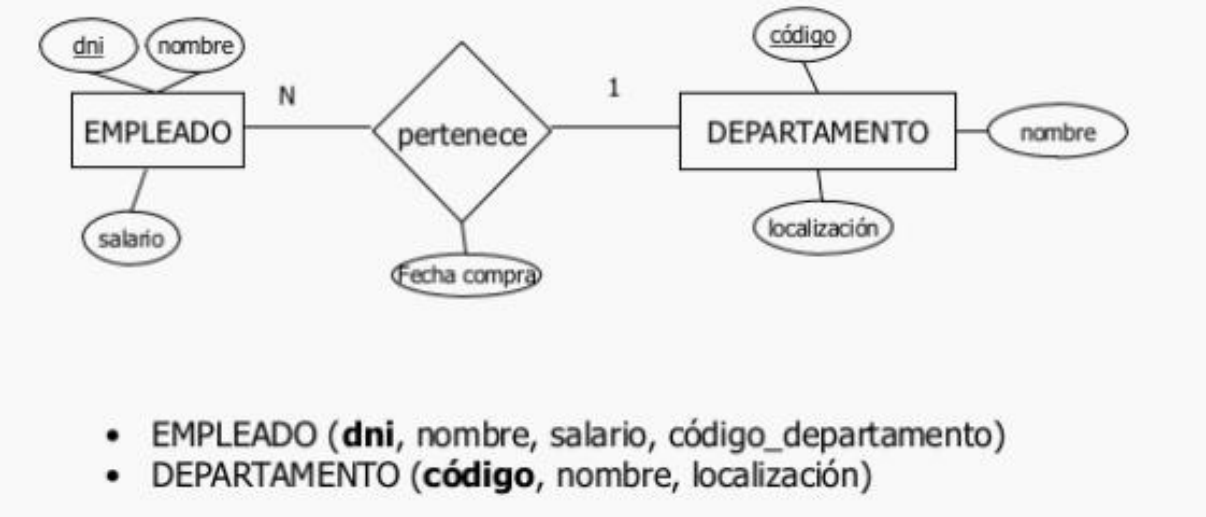


- EQUIPO (**código**, nombre, año _ fundación)
- PRESIDENTE (**dni**, nombre, código _ equipo)

- EQUIPO (**código**, nombre, año _ fundación, dni_presidente)
- PRESIDENTE (**dni**, nombre)

Ejercicios DDL - EJERCICIO 2 EMPRESA

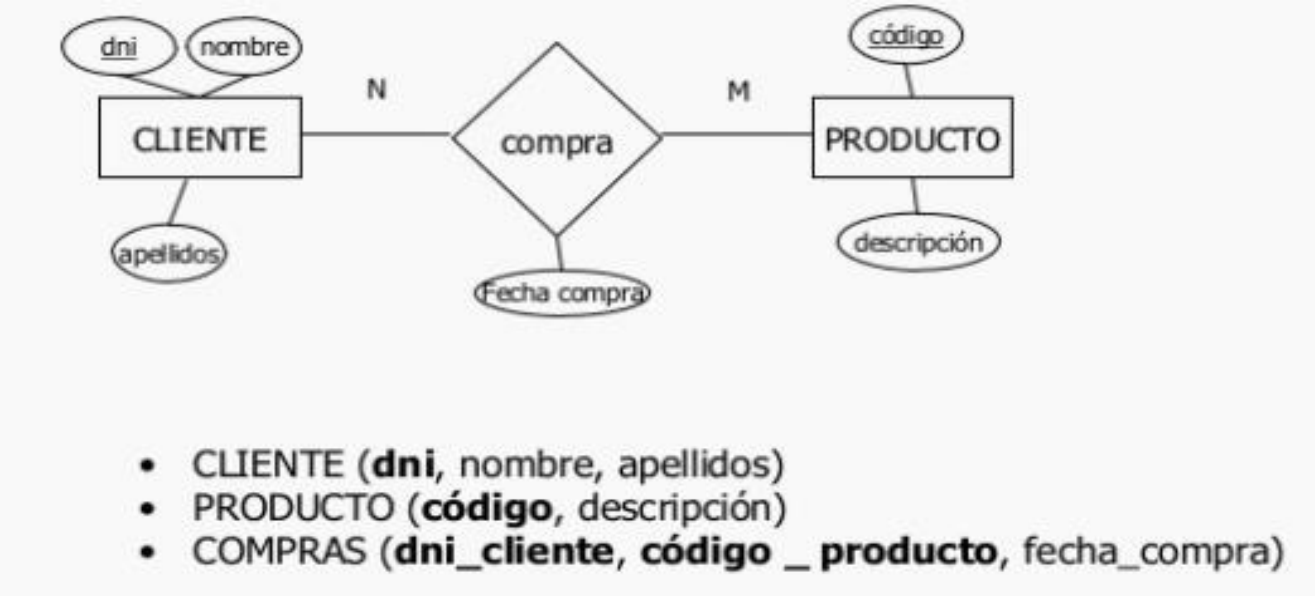
- Pasad al Modelo Físico el siguiente ejercicio:



- EMPLEADO (**dni**, nombre, salario, código_departamento)
- DEPARTAMENTO (**código**, nombre, localización)

Ejercicios DDL - EJERCICIO 3 MERCADONA

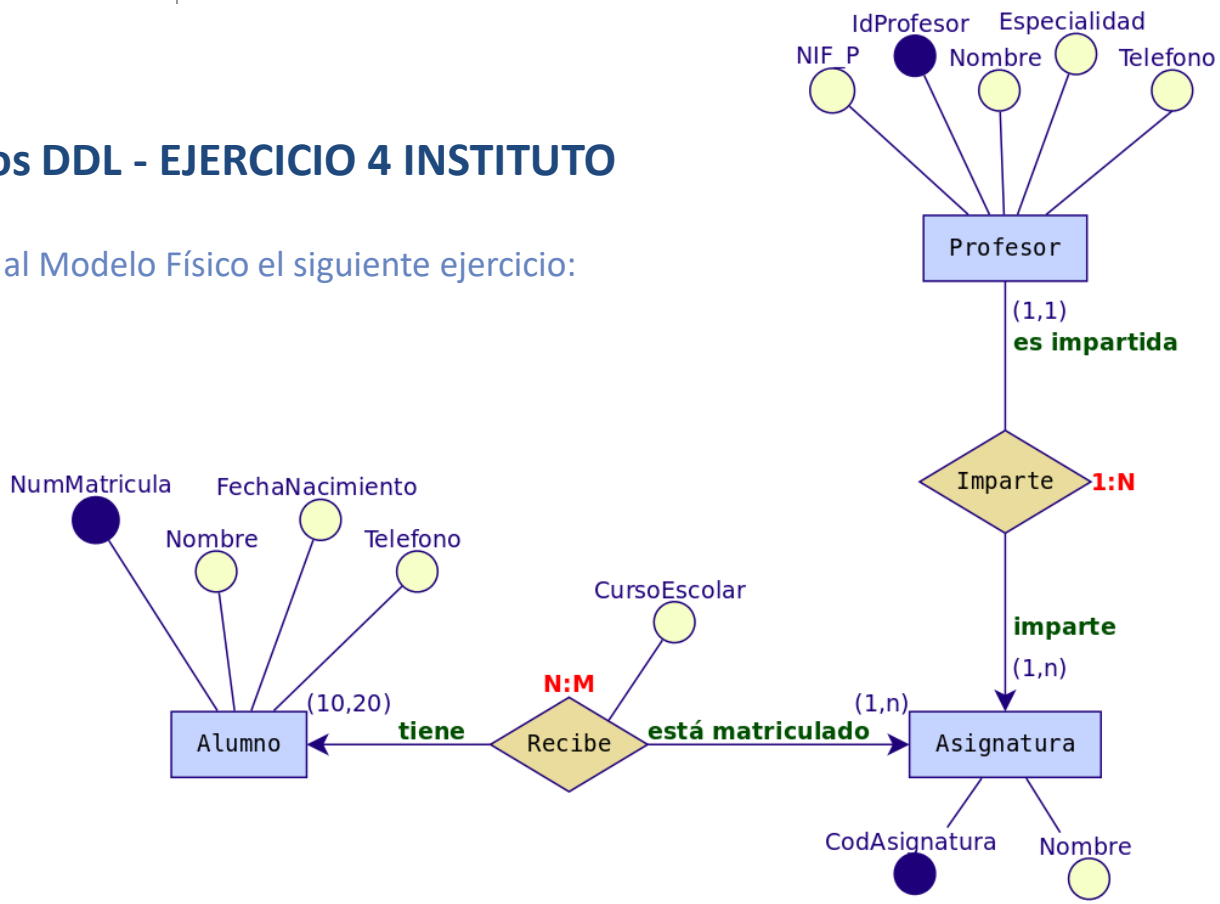
- Pasad al Modelo Físico el siguiente ejercicio:



- CLIENTE (**dni**, nombre, apellidos)
- PRODUCTO (**código**, descripción)
- COMPRAS (**dni_cliente**, **código _ producto**, fecha_compra)

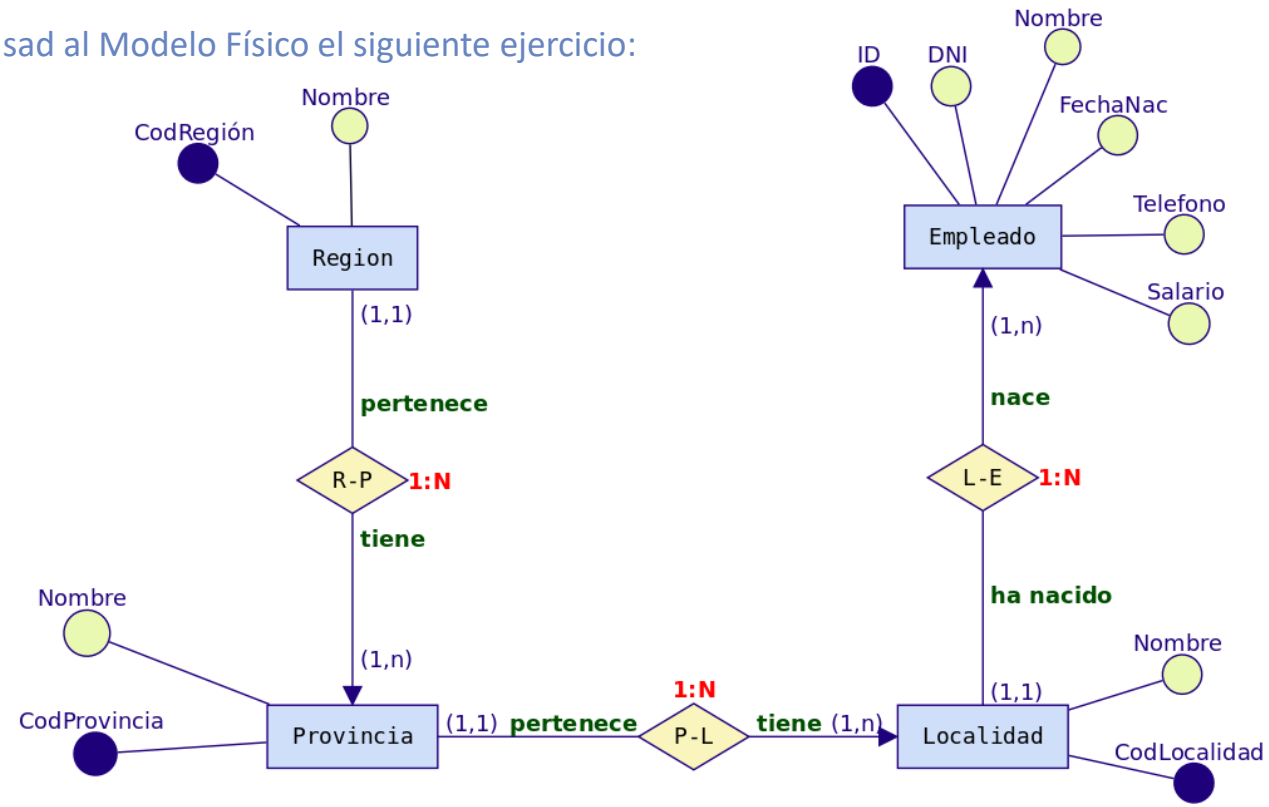
Ejercicios DDL - EJERCICIO 4 INSTITUTO

- Pasad al Modelo Físico el siguiente ejercicio:



Ejercicios DDL - EJERCICIO 5 PROVINCIAS

- Pasad al Modelo Físico el siguiente ejercicio:



Ejercicios DML

SQL_JSC(VUESTRAS INICIALES)_E1.sql

- A continuación del Ejercicio 1:
  - Debéis insertar 3 equipos y 3 presidentes (no necesariamente en este orden)
  - Modificar el nombre de uno de los dos equipos a “Equipo EOI”
  - Eliminar un Presidente

SQL_JSC(VUESTRAS INICIALES)_E2.sql

- A continuación del Ejercicio 2:
  - Debéis obtener la información para volcarla a la BD del siguiente informe:

DNI	NOMBRE	SALARIO	DEPARTAMENTO	LOCALIZACION
0001	Ángel	1200	VENTAS	ALICANTE
0002	Mario	1200	VENTAS	ALICANTE
0003	Mónica	900	BECARIOS	ALICANTE
0004	Irene	1500	DIRECTIVO	MADRID

Ejercicios DML

- A continuación del Ejercicio 3:
  - Debéis volcar las siguientes tablas en nuestra BD:

DNI	NOMBRE
0001	JUAN
0002	PEPE
0003	LAURA
0004	ALEX
0005	TOMÁS

CODIGO	NOMBRE
1	TOMATES
2	LENTEJAS
3	CHOCOLATE
4	LECHE
5	ALMENDRAS
6	PATATAS
7	NOCILLA

DNI	FECHA	CODPROD
0001	03/05/2019	1
0001	03/05/2019	2
0003	04/05/2019	3
0003	05/05/2019	2
0004	04/05/2019	5
0001	04/05/2019	6
0004	06/05/2019	4

Ejercicios DML

- A continuación del Ejercicio 4:
  - Debéis insertar el siguiente contenido:
    - Juan y Luis son alumnos de 2º de la ESO y están matriculados en Matemáticas, que es impartida por Cristobal de 42 años.
    - Lorena y Martín son alumnos de 3º de la ESO y están matriculados en Inglés, que es impartida por Sophie de 26 años.
    - Manuel es un alumno que todavía no está matriculado en nada.
    - Por último se debe tener en cuenta que se acaba de unir a la plantilla de profesores Julián de Barcelona, que impartirá Tecnología.

Ejercicios DML

- A continuación del Ejercicio 5:
  - Debéis insertar el siguiente contenido:

DNI	NOMBRE	SALARIO	LOCALIDAD	PROVINCIA
0001	Ángel	1200	SV DEL RASPEIG	ALICANTE
0002	Mario	1200	SAN JUAN	ALICANTE
0003	Mónica	900	ELCHE	ALICANTE
0004	Irene	1500	MOTRIL	GRANADA
0005	Raquel	1500	PETREL	ALICANTE
0006	Daniel	3000	MOTRIL	GRANADA
0007	Alba	2675	CAUDETE	ALBACETE
0008	Germán	1840	SV DEL RASPEIG	ALICANTE

BBDD | Structured Query Language

Ejercicios CONSULTAS SQL

- Ejercicio 1:
  - Sacar un listado con todos los equipos de futbol, incluyendo a su presidente.
  - Sacar únicamente el dni y el nombre del presidente que presida el equipo “Equipo EOI”
  - Sacar un listado de los presidentes que presidan equipos fundados a partir del 2010.

- Ejercicio 2:
  - Sacar un listado con todos los departamentos y el número de personas que trabajan en él.
  - Sacar un listado de la persona o personas que cobren el sueldo máximo de la empresa.
  - Sacar un listado de la persona o personas del departamento de VENTAS que cobren el sueldo máximo de su departamento.

Ejercicios CONSULTAS SQL

- Ejercicio 3:
  - Sacar un listado de aquellos clientes que todavía no han comprado nada.
  - Sacar el o los productos que más se han vendido.
  - Sacar la fecha en la que más productos se han vendido.
  - Sacar el o los productos que no se hayan vendido nunca.
  - Sacar los productos que han sido comprados por clientes que han comprado LENTEJAS alguna vez.

- Ejercicio 4:
  - Lista de alumnos matriculados o no y el número de asignaturas en las que está matriculado cada uno.
  - Lista de asignaturas y profesores que las imparten, junto al número de alumnos matriculados.
  - Lista de alumnos de la clase de INGLÉS.
  - Lista de Asignaturas en las que no hay alumnos matriculados.
  - Lista de Alumnos que no están matriculados de nada.

BBDD | Structured Query Language

Ejercicios CONSULTAS SQL

- Ejercicio 5:
  - Número de empleados agrupados por provincias.
  - Número de empleados agrupados por localidades que no sean de la provincia de ALICANTE.
  - Obtener el salario máximo de la provincia de GRANADA junto al empleado que lo cobra.
  - Empleados petenecientes a la región de la COMUNIDAD VALENCIANA.
  - Empleados nacidos en la provincia de ALICANTE que cobren entre 1000 y 1500 €
  - Lista de empleados de ALBACETE junto a su salario mensual y a su salario anual (12 pagas)
  - Suma de salarios mensuales y anuales agrupados y ordenados por provincias.