

## BASES DE DATOS

Desarrollo de Aplicaciones Multiplataforma

Desarrollo de Aplicaciones Web

## GESTIÓN DE BASES DE DATOS

Administración de Sistemas Informáticos en Red

### LENGUAJE DE MANIPULACIÓN DE DATOS - CONSULTAS

Luis Dorado

Vanesa Martínez

Pablo Bahillo

Alba Tortosa



# Ejercicios

# Contenido

|       |   |    |
|-------|---|----|
| 1     | CREACIÓN DE BDS, TABLAS E INSERCIÓN DE DATOS .....                  | 3  |
| 1.1   | Creación de BDs y tablas .....                                      | 3  |
| 1.1.1 | BD 'bdBasicas' .....  | 3  |
| 1.1.2 | BD 'bdPersonas' (a partir de 1.7).....                              | 3  |
| 1.1.3 | BD 'bdTrabajadores' (a partir de 1.11) .....                        | 4  |
| 1.1.4 | BD 'bdGestionVentas' (a partir de apartado 3) .....                 | 5  |
| 1.1.5 | BD 'bdTiendaInformatica' (a partir de apartado 3) .....             | 5  |
| 1.1.6 | BD 'bdTrabajadoresEdificios' (a partir de Apartado 3) .....         | 6  |
| 1.1.7 | BD 'bdEmpleadosOficinas' (a partir de apartado 3) .....             | 7  |
| 1.2   | Insertar datos en las bases de datos .....                          | 8  |
| 1.2.1 | BD 'bdBasicas' .....  | 9  |
| 1.2.2 | BD 'bdPersonas' .....   | 10 |
| 1.2.3 | BD 'bdTrabajadores' .....   | 10 |
| 1.2.4 | BD 'bdGestionVentas'.....   | 11 |
| 1.2.5 | BD 'bdTiendaInformatica'.....                                       | 11 |
| 1.2.6 | BD 'bdTrabajadoresEdificios' .....                                  | 12 |
| 1.2.7 | BD 'bdEmpleadosOficinas' .....                                      | 12 |
| 2     | LENGUAJE DE CONSULTAS SQL I .....                                   | 15 |
| 2.1   | Cláusulas SELECT y FROM: Consultas básicas (bdBasicas).....         | 15 |
| 2.2   | Cálculos y operadores aritméticos BD (bdBasicas).....               | 15 |
| 2.3   | Clausula WHERE: Evaluación de condiciones (BD 'bdBasicas').....     | 16 |
| 2.3.1 | Operadores de comparación .....                                     | 16 |
| 2.3.2 | Operadores lógicos (AND, OR, NOT).....                              | 16 |
| 2.3.3 | BETWEEN.....  | 17 |
| 2.3.4 | LIKE.....   | 18 |
| 2.3.5 | IS NULL .....   | 18 |
| 2.4   | Precedencia de operadores (Aritméticos y lógicos) (bdBasicas) ..... | 19 |
| 2.5   | Funciones sobre filas .....   | 20 |
| 2.5.1 | Funciones para cadenas de caracteres.....                           | 20 |
| 2.5.2 | Funciones matemáticas.....  | 23 |
| 2.5.3 | Funciones de fecha y hora.....                                      | 23 |
| 2.5.4 | Anidamiento de funciones (bdBasicas) .....                          | 26 |
| 2.6   | Ordenación en los resultados (bdPersonas).....                      | 26 |
| 2.7   | Limitación de resultados (bdPersonas) .....                         | 27 |
| 2.8   | Funciones de agregado (bdPersonas).....                             | 27 |
| 2.9   | GROUP BY .....  | 28 |
| 2.10  | HAVING (bdPersonas).....  | 29 |
| 3     | LENGUAJE DE CONSULTAS SQL II .....                                  | 31 |
| 3.1   | BD 'bdTrabajadoresEdificios' .....                                  | 31 |
| 3.1.1 | Repaso consultas sobre una sola tabla.....                          | 31 |
| 3.1.2 | Consultas INNER JOIN.....   | 32 |
| 3.1.3 | Consultas LEFT/RIGHT JOIN .....                                     | 33 |
| 3.2   | BD 'bdEmpleadosOficinas' .....                                      | 33 |
| 3.2.1 | Repaso consultas sobre una sola tabla.....                          | 33 |
| 3.2.2 | Consultas INNER JOIN, LEFT JOIN y otros .....                       | 34 |
| 3.3   | BD 'bdGestionVentas' .....  | 35 |
| 3.3.1 | Consultas LEFT JOIN, RIGHT JOIN y otros .....                       | 35 |
| 3.4   | BD 'bdTiendaInformatica' .....                                      | 37 |
| 3.4.1 | Consultas LEFT JOIN, RIGHT JOIN y otros .....                       | 37 |
| 4     | LENGUAJE DE CONSULTAS SQL III .....                                 | 38 |
| 4.1   | BD 'bdTrabajadoresEdificios' .....                                  | 38 |
| 4.1.1 | Subconsultas simples de comparación.....                            | 38 |
| 4.1.2 | Subconsultas simples de pertenencia .....                           | 38 |
| 4.1.3 | Subconsultas correlacionadas .....                                  | 38 |

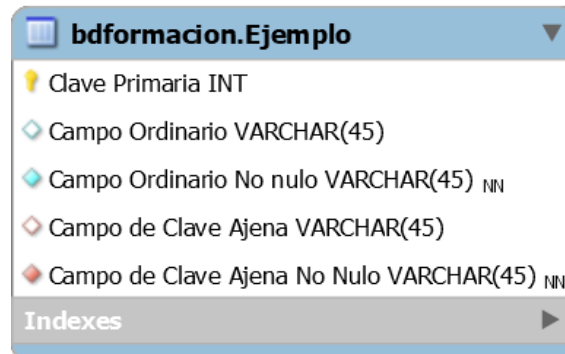
|       |                                      |    |
|-------|--------------------------------------|----|
| 4.2   | BD 'bdEmpleadosOficinas'             | 39 |
| 4.2.1 | Subconsultas simples de comparación  | 39 |
| 4.2.2 | Subconsultas simples de pertenencia  | 39 |
| 4.2.3 | Subconsultas correlacionadas         | 40 |
| 4.3   | BD 'bdGestionVentas'                 | 41 |
| 4.3.1 | Subconsultas simples de comparación  | 41 |
| 4.3.2 | Subconsultas simples de pertenencia  | 42 |
| 4.3.3 | Subconsultas con EXISTS y NOT EXISTS | 43 |
| 4.4   | BD 'bdTiendaInformatica'             | 44 |
| 4.4.1 | Subconsultas variadas                | 44 |

## 0 CREACIÓN DE BDS, TABLAS E INSERCIÓN DE DATOS

Crea las bases de datos que se te vayan indicando, sus tablas e inserta las filas en sus tablas.

### Notas:

- ✓ Comprueba que existan las BDs y las tablas antes de crearlas.
- ✓ Créalas cuando las vayas necesitando para resolver los ejercicios, no es necesario que las crees todas al inicio.
- ✓ **Recuerda: ¿Qué significan los colores asignados a cada campo?**



- Clave primaria (puede pertenecer a dos campos si es compuesta).
- Campo ordinario.
- Campo ordinario no nulo.
- Campo de clave ajena.
- Campo de clave ajena no nula.

### 0.1 Creación de BDs y tablas

#### 0.1.1 BD 'bdBasicas'

| EMPLEADOS                   |
|-----------------------------|
| DNI: VARCHAR(9) (PK)        |
| nombre: VARCHAR(30) (NN)    |
| apellidos: VARCHAR(60) (NN) |
| email: VARCHAR(60)          |
| telefono: INTEGER (NN)      |
| suelo : DECIMAL(10,2) (NN)  |
| puesto: VARCHAR(30) (NN)    |

| LIBROS                    |
|---------------------------|
| ISBN: numero (PK)         |
| titulo: VARCHAR(60) (NN)  |
| tipo: VARCHAR(30) (NN)    |
| autor: VARCHAR(90)        |
| precio: DECIMAL(8,2) (NN) |

| FACTURAS                        |
|---------------------------------|
| CÓDIGO: num (PK)                |
| destinatario: VARCHAR(90) (NN)  |
| cuenta: INTEGER (NN)            |
| importe: DECIMAL(10,2) (NN)     |
| fecha_hora: DATETIME (NN) (UNQ) |

#### 0.1.2 BD 'bdPersonas' (a partir de 1.7)

| ejercicios.personas ▼ |                         |
|-----------------------|-------------------------|
| 🔑                     | NIF VARCHAR(9)          |
| 💎                     | NSS VARCHAR(9) NN       |
| 💎                     | NOMBRE VARCHAR(25)      |
| 💎                     | APELLIDO VARCHAR(50)    |
| 💎                     | Año_NACIMIENTO INT      |
| 💎                     | FECHA_ALTA DATE         |
| 💎                     | SALARIO INT             |
| 💎                     | PUESTO ENUM(...)        |
| 💎                     | FECHA_BAJA DATE         |
| 💎                     | SEDE_TRABAJA VARCHAR(2) |
| Indexes ▶             |                         |

- ✓ NSS es una clave alternativa (ÚNIQUE y NOT NULL)
- ✓ El campo enumerado es **ENUM('JEFE', 'OPERARIO', 'JUBILADO', 'LIMPIEZA')**

### 0.1.3 BD 'bdTrabajadores' (a partir de 1.11)

| bdtrabajadores.trabajadores ▼ |                        |
|-------------------------------|------------------------|
| 🔑                             | emp_no INT NN          |
| 💎                             | apellido VARCHAR(10)   |
| 💎                             | oficio VARCHAR(10)     |
| 💎                             | dir INT                |
| 💎                             | fecha_alt DATE         |
| 💎                             | salario DECIMAL(10,2)  |
| 💎                             | comision DECIMAL(10,2) |
| 💎                             | dept_no INT NN         |
| Indexes ▶                     |                        |

SOLUCIÓN:

#### 1 • create database if not exists ejercicios;

```
CREATE TABLE empleados(
  DNI VARCHAR(30),
  nombre VARCHAR(30),
  email VARCHAR(30),
  telefono INTEGER,
  sueldo DECIMAL (10,2),
  puesto VARCHAR(30)
);
```

```
CREATE TABLE libros(
  ISBN INT,
  titulo VARCHAR(60),
  tipo VARCHAR(30),
  autor VARCHAR(90),
  precio DECIMAL(8,2)
);
```

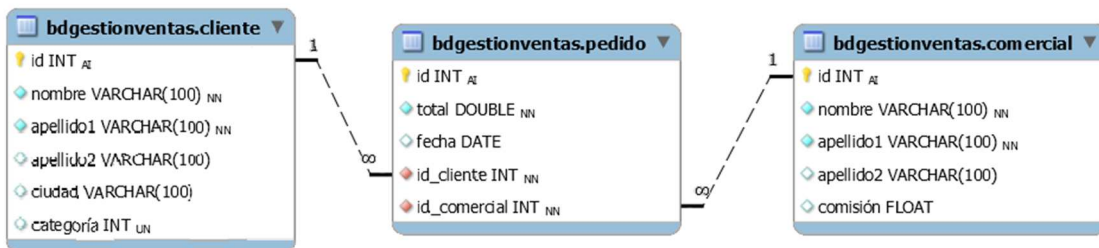
```

CREATE TABLE codigo(
CODIGO INT,
destinatario VARCHAR(90),
cuenta INTEGER,
importe DECIMAL(10,2),
fecha_hora DATETIME
);

drop table if exists personas;
create table PERSONAS(
NIF varchar(9) PRIMARY KEY,
NSS VARCHAR (9) UNIQUE NOT NULL,
NOMBRE varchar(25),
APELLIDO VARCHAR(50),
AÑO_NACIMIENTO INT,
FECHA_ALTA date,
SALARIO INT,
PUESTO ENUM('JEFE','OPERARIO','JUBILADO','LIMPIEZA'),
FECHA_BAJA DATE DEFAULT NULL,
SEDE_TRABAJA VARCHAR(2)
);

```

#### 0.1.4 BD 'bdGestionVentas' (a partir de apartado 3)



```

DROP DATABASE IF EXISTS bdGestionVentas;
CREATE DATABASE bdGestionVentas;
USE bdGestionVentas;

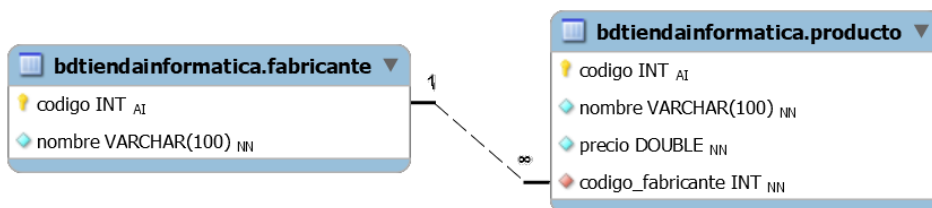
CREATE TABLE cliente (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
apellido1 VARCHAR(100) NOT NULL,
apellido2 VARCHAR(100),
ciudad VARCHAR(100),
categoría INT UNSIGNED
);

CREATE TABLE comercial (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
apellido1 VARCHAR(100) NOT NULL,
apellido2 VARCHAR(100),
comisión FLOAT
);

CREATE TABLE pedido (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
total DOUBLE NOT NULL,
fecha DATE,
id_cliente INT UNSIGNED NOT NULL,
id_comercial INT UNSIGNED NOT NULL,
FOREIGN KEY (id_cliente) REFERENCES cliente(id),
FOREIGN KEY (id_comercial) REFERENCES comercial(id)
);

```

#### 0.1.5 BD 'bdTiendaInformatica' (a partir de apartado 3)



```

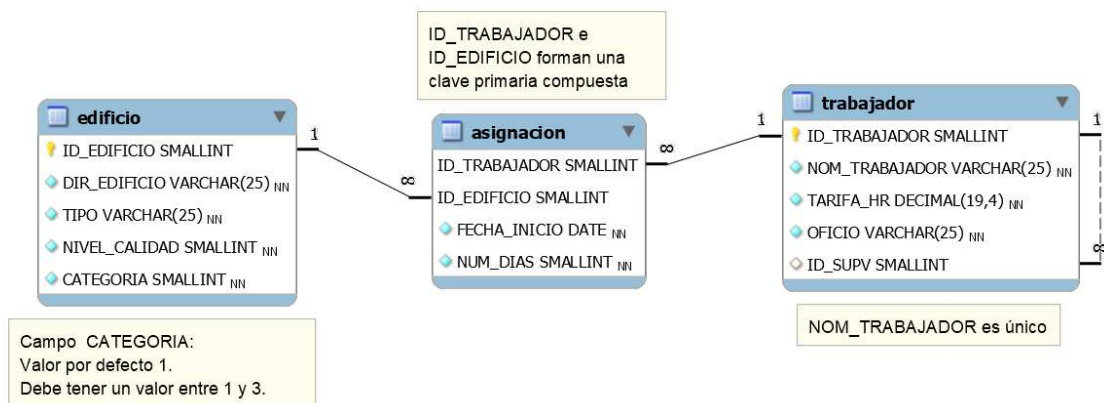
DROP DATABASE IF EXISTS bdTiendaInformatica;
CREATE DATABASE bdTiendaInformatica;
USE bdTiendaInformatica;

CREATE TABLE fabricante (
  codigo INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL
);

CREATE TABLE producto (
  codigo INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  precio DOUBLE NOT NULL,
  codigo_fabricante INT UNSIGNED NOT NULL,
  FOREIGN KEY (codigo_fabricante) REFERENCES fabricante(codigo)
);

```

### 0.1.6 BD 'bdTrabajadoresEdificios' (a partir de Apartado 3)



```

create database if not exists bdTrabajadorEdificio;
use bdTrabajadorEdificio;

/*CREACION DE TABLAS*/
create table trabajador(
  id_trabajador smallint,
  nom_trabajador varchar(25) UNIQUE NOT NULL,
  tarifa_hr decimal(19,4) NOT NULL,
  oficio varchar(25) NOT NULL,
  id_supv smallint,
  constraint pk_trabajador primary key(id_trabajador)
);
create table edificio(
  id_edificio smallint,
  dir_edificio varchar(25) not null,
  tipo varchar(25) not null,
  nivel_calidad smallint not null,
  categoria smallint not null default (1),
  constraint pk_edificio primary key(id_edificio),
  constraint ck_edificio check (categoria>=1 and categoria<=3)
);

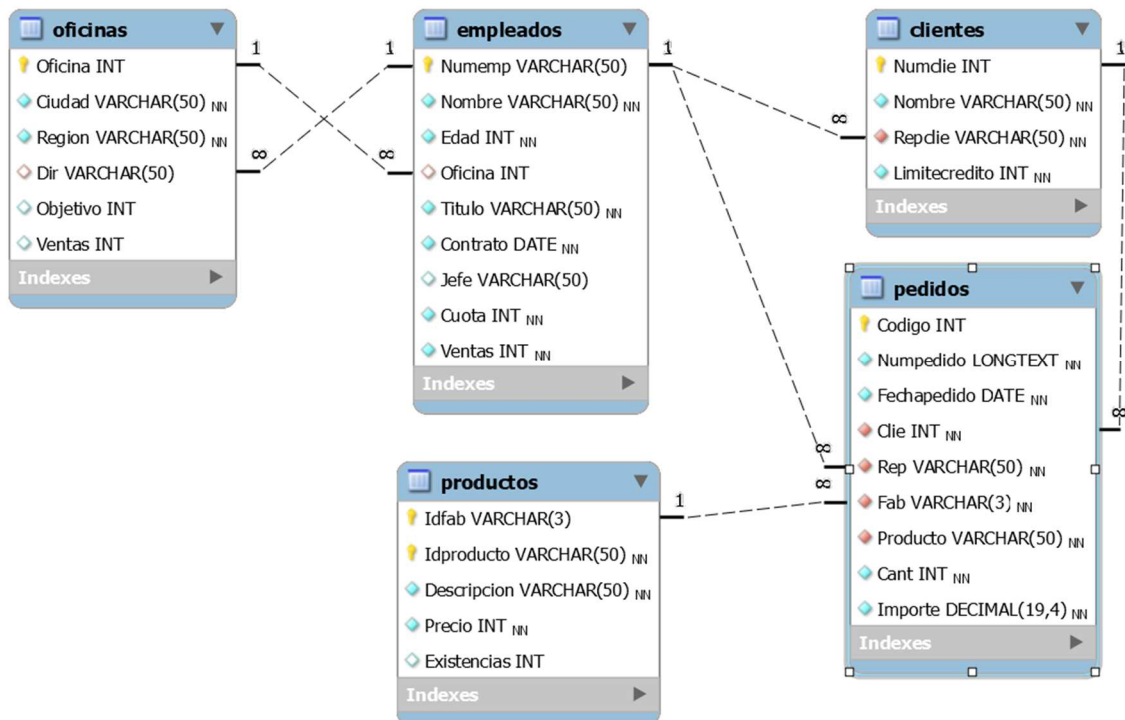
```

```

create table asignacion(
id_trabajador smallint,
id_edificio smallint,
fecha_inicio date not null,
num_dias smallint not null,
constraint fk_asignacion_trabajador foreign key (id_trabajador) references trabajador(id_trabajador),
constraint fk_asignacion_edificio foreign key (id_edificio) references edificio(id_edificio)
);

```

### 0.1.7 BD 'bdEmpleadosOficinas' (a partir de apartado 3)



**Nota importante:** Como oficinas y empleados se hallan relacionados doblemente en ambas direcciones deberán dejar al menos estas tablas sin restricción de clave ajena hasta que hayas realizado las inserciones.



```

create table empleados(
    numemp int,
    nombre varchar(20),
    edad int,
    oficina int,
    titulo varchar(20),
    contrato date,
    jefe int,
    cuota int unsigned,
    ventas int unsigned,
    primary key (numemp)
);

create table productos(
    idfab varchar(5),
    idproducto varchar(10),
    descripcion varchar(20),
    precio int unsigned,
    existencias int,
    primary key(idfab, idproducto)
);

create table oficinas(
    oficina int,
    ciudad varchar(20),
    region varchar(20),
    dir int,
    objetivo int unsigned,
    ventas int unsigned,
    primary key (oficina)
);

create table clientes(
    numclie int,
    nombre varchar(20),
    repclie int,
    limitecredito int,
    primary key (numclie)
);

create table pedidos(
    codigo int,
    numpedido longtext not null,
    fechapedido date not null,
    clie int not null,
    rep int not null,
    fab varchar(5) not null,
    producto varchar(10) not null,
    cant int,
    importe int,
    primary key (codigo)
);

```

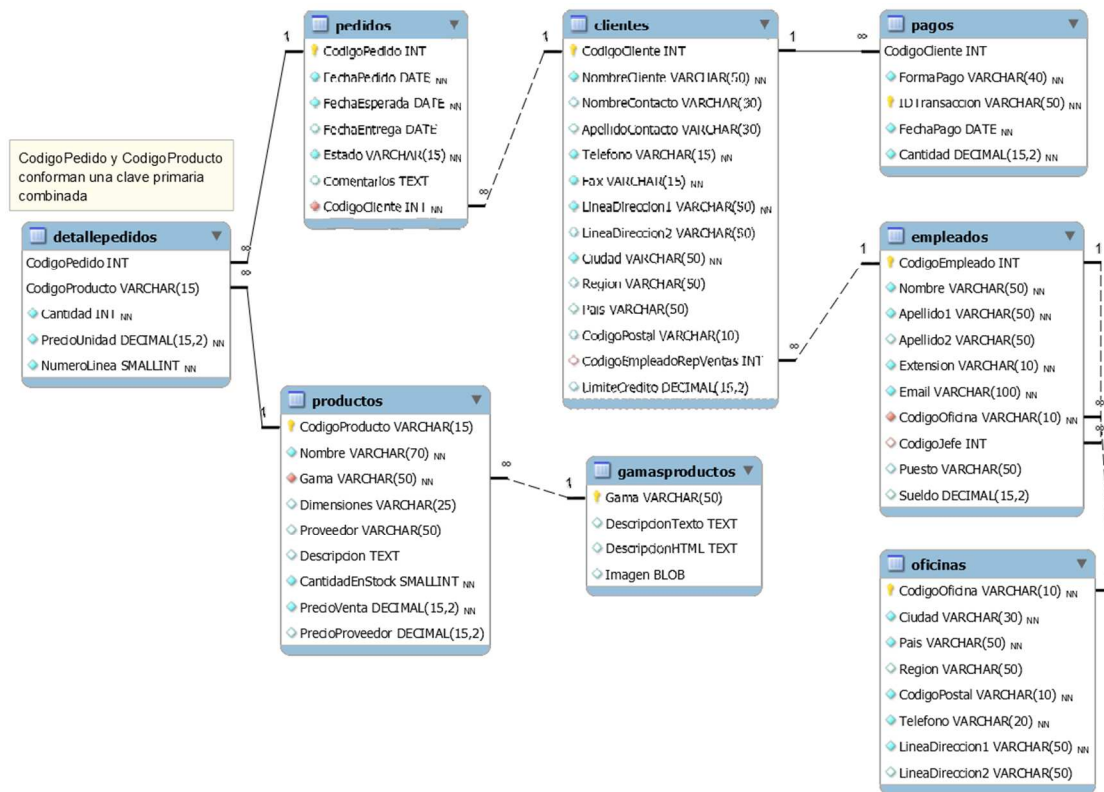
### INSERCCIONES PREVIAS AL ESTABLECIMIENTO DE RESTRICCIONES DE CLAVE AJENA

```

-- Incluir las claves foráneas
alter table empleados add constraint fk_jefe foreign key (jefe) references empleados(numemp);
alter table empleados add constraint fk_oficina foreign key (oficina) references oficinas(oficina);
alter table oficinas add constraint fk_oficinas foreign key (dir) references empleados(numemp);
alter table clientes add constraint fp_repclie foreign key (repclie) references empleados(numemp);
alter table pedidos add constraint fk_cliente foreign key (clie) references clientes(numclie);
alter table pedidos add constraint fk_rep foreign key (rep) references empleados(numemp);
alter table pedidos add constraint fk_fp foreign key (fab, producto) references productos(idfab, idproducto);

```

#### 0.1.1 BD 'bdJardinería' (a partir de apartado 3)



## 0.2 Insertar datos en las bases de datos

Las inserciones de datos suelen hallarse en un fichero SQL en la plataforma Moodle o similar.

### 0.2.1 BD 'bdBasicas'

Empleados:

| dni       | nombre   | apellidos        | email                      | telefono  | sueldo   | puesto     |
|-----------|----------|------------------|----------------------------|-----------|----------|------------|
| 11111111V | Luis     | Dorado Pérez     | ldorado@gmail.com          | 912587763 | 2130.95  | profesor   |
| 12345678A | Juan     | López Pérez      | jlope@navarra.es           | 948111111 | 1235.48  | encargado  |
| 15862483L | Marta    | González Ramos   | martitagr@navarra.es       | 979729709 | 1673.17  | operario   |
| 32497256V | Pedro    | Santos Ruiz      | NULL                       | 948963432 | 1338.67  | operario   |
| 35786521F | Ernesto  | Sevilla          | ernestosevilla@chanante.es | 917234685 | 12395.27 | actor      |
| 36218657H | Ramón    | Henández Rojo    | NULL                       | 948714282 | 1396.46  | operario   |
| 38426851B | Laura    | Garcés Sevilla   | NULL                       | 948711515 | 1349.57  | operario   |
| 65279423G | Fernando | Pastor Hernández | fphernan@hotmail.com       | 986345761 | 1268.76  | operario   |
| 67853427T | María    | León Sánchez     | maria.lsan@outlook.es      | 920675428 | 1747.76  | supervisor |
| 75681263D | Pilar    | Castro López     | pcl83@gmail.com            | 948715632 | 1376.48  | operario   |
| 75948264K | Vanessa  | Pérez Melendro   | vpm97@gmail.com            | 948367589 | 2037.67  | profesor   |
| 98642531C | Alba     | Etxeberria Luis  | alba.etxe@gmail.com        | 970256485 | 1537.95  | encargado  |
| 98765432C | David    | Herrero Sánchez  | davidherresan@hotmail.es   | 964851123 | 1398.88  | supervisor |

Libros:

| isbn      | titulo                         | tipo        | autor                      | precio |
|-----------|--------------------------------|-------------|----------------------------|--------|
| 125789351 | Aquitania                      | novela      | Eva García Sáenz de Urturi | 21.90  |
| 326541968 | El principito                  | novela      | Antoine de Saint-Exupery   | 6.95   |
| 365106701 | La buena cocina                | cocina      | Karlos Arguiñano           | 24.95  |
| 431501467 | Redes Locales                  | escolar     | varios                     | 21.89  |
| 472617812 | Sapiens: de animales a dioses  | divulgación | Yuval Noah Harari          | 19.95  |
| 542612578 | El lazarillo de Tormes         | novela      | anónimo                    | 7.86   |
| 653723689 | Pensar rápido, pensar despacio | divulgación | Daniel Kahneman            | 23.49  |
| 705940145 | La ciudad de vapor             | novela      | Carlos Ruiz Zafón          | 17.90  |
| 734521667 | Adiós al frío                  | novela      | Elvira Sastre              | 12.00  |
| 825430758 | Una tierra prometida           | biografico  | Barack Obama               | 32.90  |
| 917531860 | Poema de Mio Cid               | poesía      | NULL                       | 11.15  |
| 936541869 | Flamenca                       | novela      | NULL                       | 7.95   |
| 996378927 | Las mil y una noches           | novela      | NULL                       | 12.95  |

Facturas:

| codigo | destinatario      | cuenta   | importe | fecha_hora          |
|--------|-------------------|----------|---------|---------------------|
| 27792  | Mark Zuckerberg   | 75968423 | 846.49  | 2008-02-08 12:14:35 |
| 123456 | Jhonny Silverhand | 75481256 | 59.99   | 2020-12-10 00:00:27 |
| 246792 | Nikola Tesla      | 15862842 | 23.75   | 2005-08-31 19:57:36 |
| 335902 | Bolvar Fordragon  | 15862842 | 179.96  | 2004-11-23 13:07:49 |
| 354503 | Bill Gates        | 68452358 | 139.83  | 2009-03-25 08:09:12 |
| 456123 | Mark Hamill       | 56348956 | 59.99   | 2007-05-25 17:15:47 |
| 487016 | John Von Neumann  | 28426684 | 49.95   | 2001-11-09 23:14:39 |
| 578925 | Alan Turing       | 85325486 | 658.75  | 2015-06-15 18:37:24 |
| 629430 | Steve Wozniak     | 62842668 | 386.02  | 2017-09-04 07:57:23 |
| 710349 | Linus Torvalds    | 86284266 | 30.19   | 2012-08-09 17:15:23 |

### 0.2.2 BD 'bdPersonas'

| NIF       | NSS       | NOMBRE  | APELLIDO         | AÑO_NACIMIENTO | FECHA_ALTA | SALARIO | PUESTO   | FECHA_BAJA | SEDE_TRABAJA |
|-----------|-----------|---------|------------------|----------------|------------|---------|----------|------------|--------------|
| 12121212Q | 56565656T | CAMILO  | SAMUEL PAZO      | 1978           | 2009-05-02 | 900     | JUBILADO | 2020-10-06 | AC           |
| 52563932N | 32569885M | JOSE    | MARIN GIO        | 1998           | 2010-05-02 | 1120    | OPERARIO | NULL       | AC           |
| 5858585J  | 28484844G | PEDRO   | AGUILAS REDONDEO | 1960           | 2000-05-02 | 850     | JUBILADO | 2020-11-02 | AC           |
| 74663363Z | 20000010D | VALERIA | ALBORAN LOPEZ    | 1985           | 2000-05-12 | 1150    | OPERARIO | NULL       | AC           |
| 7472471PZ | 25856944S | ALONSO  | FERNANDEZ ROBLES | 1970           | 2000-02-25 | 3520    | JEFE     | NULL       | AC           |
| 74725894L | 25858985P | LEO     | MARTIN PEREZ     | 1981           | 2000-10-20 | 1120    | OPERARIO | 2020-11-06 | AC           |
| 74725898T | 8888888K  | LALI    | MARTINEZ ACENA   | 1982           | 1999-10-05 | 1000    | LIMPIEZA | 2009-12-06 | AC           |
| 74895951T | 89568956Y | LIO     | RUTA GOLA        | 1980           | 2010-02-05 | 1000    | LIMPIEZA | NULL       | AC           |
| 89898989J | 96969696A | AMADA   | GARAYOA SOLDADO  | 1971           | 1998-05-02 | 1420    | OPERARIO | NULL       | AC           |

### 0.2.3 BD 'bdTrabajadores'

| emp_no | apellido  | oficio     | dir  | fecha_alt  | salario | comision | dept_no |
|--------|-----------|------------|------|------------|---------|----------|---------|
| 7369   | SÁNCHEZ   | EMPLEADO   | 7902 | 1990-12-17 | 1040.00 | NULL     | 20      |
| 7499   | ARROYO    | VENDEDOR   | 7698 | 1990-02-20 | 1500.00 | 390.00   | 30      |
| 7521   | SALA      | VENDEDOR   | 7698 | 1991-02-22 | 1625.00 | 650.00   | 30      |
| 7566   | JIMÉNEZ   | DIRECTOR   | 7839 | 1991-04-02 | 2900.00 | NULL     | 20      |
| 7654   | MARTÍN    | VENDEDOR   | 7698 | 1991-09-29 | 1600.00 | 1020.00  | 30      |
| 7698   | NEGRO     | DIRECTOR   | 7839 | 1991-05-01 | 3005.00 | NULL     | 30      |
| 7782   | CEREZO    | DIRECTOR   | 7839 | 1991-06-09 | 2885.00 | NULL     | 10      |
| 7788   | GIL       | ANALISTA   | 7566 | 1991-11-09 | 3000.00 | NULL     | 20      |
| 7839   | REY       | PRESIDENTE | NULL | 1991-11-17 | 4100.00 | NULL     | 10      |
| 7844   | TOVAR     | VENDEDOR   | 7698 | 1991-09-08 | 1350.00 | 0.00     | 30      |
| 7876   | ALONSO    | EMPLEADO   | 7788 | 1991-09-23 | 1430.00 | NULL     | 20      |
| 7900   | JIMENO    | EMPLEADO   | 7698 | 1991-12-03 | 1335.00 | NULL     | 30      |
| 7902   | FERNÁN... | ANALISTA   | 7566 | 1991-12-03 | 3000.00 | NULL     | 20      |
| 7934   | MUÑOZ     | EMPLEADO   | 7782 | 1992-01-23 | 1690.00 | NULL     | 10      |

#### 0.2.4 BD 'bdGestionVentas'

| id | nombre    | apellido1 | apellido2 | ciudad  | categoría |
|----|-----------|-----------|-----------|---------|-----------|
| 1  | Aarón     | Rivero    | Gómez     | Almería | 100       |
| 2  | Adela     | Salas     | Díaz      | Granada | 200       |
| 3  | Adolfo    | Rubio     | Flores    | Sevilla | NULL      |
| 4  | Adrián    | Suárez    | NULL      | Jaén    | 300       |
| 5  | Marcos    | Loyola    | Méndez    | Almería | 200       |
| 6  | María     | Santana   | Moreno    | Cádiz   | 100       |
| 7  | Pilar     | Ruiz      | NULL      | Sevilla | 300       |
| 8  | Pepe      | Ruiz      | Santana   | Huelva  | 200       |
| 9  | Guillermo | López     | Gómez     | Granada | 225       |
| 10 | Daniel    | Santana   | Loyola    | Sevilla | 125       |

| id | total   | fecha      | id_cliente | id_comercial |
|----|---------|------------|------------|--------------|
| 1  | 150.5   | 2017-10-05 | 5          | 2            |
| 2  | 270.65  | 2016-09-10 | 1          | 5            |
| 3  | 65.26   | 2017-10-05 | 2          | 1            |
| 4  | 110.5   | 2016-08-17 | 8          | 3            |
| 5  | 948.5   | 2017-09-10 | 5          | 2            |
| 6  | 2400.6  | 2016-07-27 | 7          | 1            |
| 7  | 5760    | 2015-09-10 | 2          | 1            |
| 8  | 1983.43 | 2017-10-10 | 4          | 6            |
| 9  | 2480.4  | 2016-10-10 | 8          | 3            |
| 10 | 250.45  | 2015-06-27 | 8          | 2            |
| 11 | 75.29   | 2016-08-17 | 3          | 7            |
| 12 | 3045.6  | 2017-04-25 | 2          | 1            |
| 13 | 545.75  | 2019-01-25 | 6          | 1            |
| 14 | 145.82  | 2017-02-02 | 6          | 1            |
| 15 | 370.85  | 2019-03-11 | 1          | 5            |
| 16 | 2389.23 | 2019-03-11 | 1          | 5            |

| id | nombre  | apellido1 | apellido2 | comisión |
|----|---------|-----------|-----------|----------|
| 1  | Daniel  | Sáez      | Vega      | 0.15     |
| 2  | Juan    | Gómez     | López     | 0.13     |
| 3  | Diego   | Flores    | Salas     | 0.11     |
| 4  | Marta   | Herrera   | Gil       | 0.14     |
| 5  | Antonio | Carretero | Ortega    | 0.12     |
| 6  | Manuel  | Domínguez | Hernández | 0.13     |
| 7  | Antonio | Vega      | Hernández | 0.11     |
| 8  | Alfredo | Ruiz      | Flores    | 0.05     |

#### 0.2.5 BD 'bdTiendaInformatica'

| codigo | nombre          |
|--------|-----------------|
| 1      | Asus            |
| 2      | Lenovo          |
| 3      | Hewlett-Packard |
| 4      | Samsung         |
| 5      | Seagate         |
| 6      | Crucial         |
| 7      | Gigabyte        |
| 8      | Huawei          |
| 9      | Xiaomi          |

| codigo | nombre                       | precio | codigo_fabricante |
|--------|------------------------------|--------|-------------------|
| 1      | Disco duro SATA3 1TB         | 86.99  | 5                 |
| 2      | Memoria RAM DDR4 8GB         | 120    | 6                 |
| 3      | Disco SSD 1 TB               | 150.99 | 4                 |
| 4      | GeForce GTX 1050Ti           | 185    | 7                 |
| 5      | GeForce GTX 1080 Xtreme      | 755    | 6                 |
| 6      | Monitor 24 LED Full HD       | 202    | 1                 |
| 7      | Monitor 27 LED Full HD       | 245.99 | 1                 |
| 8      | Portátil Yoga 520            | 559    | 2                 |
| 9      | Portátil Ideapad 320         | 444    | 2                 |
| 10     | Impresora HP Deskjet 3720    | 59.99  | 3                 |
| 11     | Impresora HP Laserjet Pro... | 180    | 3                 |



## 0.2.6 BD 'bdTrabajadoresEdificios'

## TRABAJADOR

| ID_TRABAJADOR | NOM_TRABAJADOR | TARIFA_HR | OFICIO       | ID_SUPV |
|---------------|----------------|-----------|--------------|---------|
| 1235          | M Faraday      | 12,50     | Electricista | 1311    |
| 1412          | C Nemo         | 13,75     | Fontanero    | 1520    |
| 2920          | R Garret       | 10,00     | Albañil      | 2920    |
| 3231          | P Mason        | 17,40     | Carpintero   | 3231    |
| 1520          | H Rickover     | 11,75     | Fontanero    | 1520    |
| 1311          | C Coulomb      | 15,50     | Electricista | 1311    |
| 3001          | J Barrister    | 8,20      | Carpintero   | 3231    |

## EDIFICIO

| ID_EDIFICIO | DIR_EDIFICIO | TIPO       | NIVEL_CALIDAD | CATEGORÍA |
|-------------|--------------|------------|---------------|-----------|
| 312         | 123 Elm      | Oficina    | 2             | 2         |
| 435         | 456 Maple    | Comercio   | 1             | 1         |
| 515         | 789 Oak      | Residencia | 3             | 1         |
| 210         | 1011 Birch   | Oficina    | 3             | 1         |
| 111         | 1213 Aspen   | Oficina    | 4             | 1         |
| 460         | 1415 Beech   | Almacén    | 3             | 3         |

## ASIGNACIÓN

| ID_TRABAJADOR | ID_EDIFICIO | FECHA_INICIO | NUM_DÍAS |
|---------------|-------------|--------------|----------|
| 1235          | 312         | 10/10        | 5        |
| 1412          | 312         | 01/10        | 10       |
| 1235          | 515         | 17/10        | 22       |
| 2920          | 460         | 05/10        | 18       |
| 1412          | 460         | 08/12        | 18       |
| 2920          | 435         | 28/10        | 10       |
| 2920          | 210         | 10/11        | 15       |
| 3231          | 111         | 10/11        | 8        |
| 1412          | 435         | 15/10        | 15       |
| 1412          | 515         | 05/11        | 8        |
| 3231          | 312         | 24/10        | 20       |
| 1520          | 515         | 09/10        | 14       |
| 1311          | 435         | 08/10        | 12       |
| 1412          | 210         | 15/11        | 12       |
| 1412          | 111         | 01/12        | 4        |
| 3001          | 111         | 08/10        | 14       |
| 1311          | 460         | 23/10        | 24       |
| 1520          | 312         | 30/10        | 17       |
| 3001          | 210         | 27/10        | 14       |

## 0.2.7 BD 'bdEmpleadosOficinas'

## Clientes

| numdie | nombre            | repdie | limitecredito |
|--------|-------------------|--------|---------------|
| 2101   | Luis Garcia Anton | 106    | 65000         |
| 2102   | Alvaro Rodriguez  | 101    | 65000         |
| 2105   | Antonio Canales   | 101    | 45000         |
| 2106   | Juan Suarez       | 102    | 65000         |
| 2107   | Julian Lopez      | 110    | 35000         |
| 2108   | Julia Antequera   | 109    | 55000         |
| 2109   | Alberto Juanes    | 103    | 25000         |
| 2111   | Cristobal Garcia  | 103    | 50000         |
| 2112   | Maria Silva       | 108    | 50000         |
| 2113   | Luis Maron        | 104    | 20000         |
| 2114   | Cristini Bulini   | 102    | 20000         |
| 2115   | Vicente Martinez  | 101    | 20000         |
| 2117   | Carlos Tena       | 106    | 35000         |
| 2118   | Junipero Alvarez  | 108    | 60000         |
| 2119   | Salomon Bueno     | 109    | 25000         |
| 2120   | Juan Malo         | 102    | 50000         |
| 2121   | Vicente Rios      | 103    | 45000         |
| 2123   | Jose Libros       | 102    | 40000         |
| 2124   | Juan Bolto        | 107    | 40000         |

**Empleados**

| numemp | nombre          | edad | oficina | titulo        | contrato   | jefe | cuota  | ventas |
|--------|-----------------|------|---------|---------------|------------|------|--------|--------|
| 101    | Antonio Viger   | 45   | 12      | Representante | 1986-10-20 | 104  | 300000 | 305000 |
| 102    | Alvaro Jaumes   | 48   | 21      | Representante | 1986-12-10 | 108  | 350000 | 474000 |
| 103    | Juan Rovira     | 29   | 12      | Representante | 1987-03-01 | 104  | 275000 | 286000 |
| 104    | Vicente Pantal  | 37   | 13      | Representante | 1998-02-12 | 104  | 350000 | 368000 |
| 106    | Luis Antonio    | 52   | 11      | Dir General   | 1988-06-14 | 104  | 275000 | 299000 |
| 107    | Jorge Gutierrez | 49   | 22      | Representante | 1988-11-14 | 108  | 300000 | 186000 |
| 108    | Ana Bustamante  | 62   | 21      | Dir Ventas    | 1989-10-12 | 106  | 350000 | 362100 |
| 109    | Maria Suta      | 31   | 11      | Representante | 1999-10-12 | 106  | 300000 | 392000 |
| 110    | Juan Victor     | 41   | NULL    | Repesentante  | 1990-01-13 | 104  | 0      | 76000  |

**Oficinas**

| oficina | ciudad    | region | dir  | objetivo | ventas |
|---------|-----------|--------|------|----------|--------|
| 11      | Valencia  | Este   | 106  | 575000   | 693000 |
| 12      | Alicante  | Este   | 104  | 800000   | 735000 |
| 13      | Castellon | Este   | 104  | 350000   | 368000 |
| 21      | Badajoz   | Oeste  | 108  | 725000   | 836000 |
| 22      | A Coruña  | Oeste  | 108  | 300000   | 186000 |
| 23      | Madrid    | Centro | 108  | NULL     | NULL   |
| 24      | Madrid    | Centro | 108  | 250000   | 150000 |
| 26      | Pamplona  | Norte  | NULL | NULL     | NULL   |
| 28      | Valencia  | Este   | NULL | 900000   | NULL   |
| 30      | Madrid    | centro | NULL | 100000   | NULL   |
| 40      | Madrid    | centro | NULL | 100000   | 0      |

**Pedidos**

| codigo | numpedido | fechapedido | die  | rep | fab | producto | cant | importe |
|--------|-----------|-------------|------|-----|-----|----------|------|---------|
| 1      | 110036    | 1997-01-02  | 2107 | 110 | aci | 4100z    | 9    | 2250    |
| 2      | 110036    | 1997-01-02  | 2117 | 106 | rei | 2a44l    | 7    | 31500   |
| 4      | 112968    | 1990-01-11  | 2102 | 101 | aci | 41004    | 34   | 3978    |
| 5      | 112975    | 1997-02-11  | 2111 | 103 | rei | 2a44g    | 6    | 2100    |
| 6      | 112979    | 1989-10-12  | 2114 | 108 | aci | 4100z    | 6    | 15000   |
| 9      | 112989    | 1997-12-10  | 2101 | 106 | fea | 114      | 6    | 1458    |
| 10     | 112992    | 1990-04-15  | 2118 | 108 | aci | 41002    | 10   | 760     |
| 11     | 112993    | 1997-03-10  | 2106 | 102 | rei | 2a45c    | 24   | 1896    |
| 12     | 112997    | 1997-07-04  | 2124 | 107 | bic | 41003    | 1    | 652     |
| 13     | 113003    | 1997-02-05  | 2108 | 109 | imm | 779c     | 3    | 5625    |
| 14     | 113007    | 1997-01-01  | 2112 | 108 | imm | 773c     | 3    | 2925    |
| 15     | 113012    | 1997-05-05  | 2111 | 104 | aci | 41003    | 35   | 3745    |
| 16     | 113013    | 1997-08-06  | 2118 | 108 | bic | 41003    | 1    | 652     |
| 17     | 113024    | 1997-07-04  | 2114 | 108 | qsa | xk47     | 20   | 7100    |
| 19     | 113034    | 1997-11-05  | 2107 | 110 | rei | 2a45c    | 8    | 632     |
| 20     | 113042    | 1997-01-01  | 2113 | 101 | rei | 2a44r    | 5    | 22500   |
| 21     | 113045    | 1997-07-02  | 2112 | 108 | rei | 2a44r    | 10   | 45000   |
| 22     | 113048    | 1997-02-02  | 2120 | 102 | imm | 779c     | 2    | 3750    |
| 23     | 113049    | 1997-04-04  | 2118 | 108 | qsa | xk47     | 2    | 776     |
| 24     | 113051    | 1997-07-06  | 2118 | 108 | qsa | xk47     | 4    | 1420    |
| 25     | 113055    | 2009-04-01  | 2108 | 101 | aci | 4100x    | 6    | 150     |
| 26     | 113057    | 1997-11-01  | 2111 | 103 | aci | 4100x    | 24   | 600     |
| 27     | 113058    | 1989-07-04  | 2108 | 109 | fea | 112      | 10   | 1480    |
| 28     | 113062    | 1997-07-04  | 2124 | 107 | bic | 41003    | 10   | 2430    |
| 29     | 113065    | 1997-06-03  | 2106 | 102 | qsa | xk47     | 6    | 2130    |
| 30     | 113069    | 1997-08-01  | 2109 | 107 | imm | 773c     | 22   | 31350   |
| 31     | 114070    | 2009-03-15  | 2124 | 107 | ACI | 41001    | 50   | 12340   |

### Productos

| idfab | idproducto | descripcion | precio | existencias |
|-------|------------|-------------|--------|-------------|
| aci   | 41001      | Arandela    | 58     | 277         |
| aci   | 41002      | Bisagra     | 80     | 167         |
| aci   | 41003      | Art t3      | 112    | 207         |
| aci   | 41004      | Art t4      | 123    | 139         |
| aci   | 4100x      | Junta       | 26     | 37          |
| aci   | 4100y      | Extractor   | 2888   | 25          |
| aci   | 4100z      | Mont        | 2625   | 28          |
| bic   | 41003      | Manivela    | 652    | 3           |
| bic   | 41009      | Rodamiento  | 225    | 78          |
| bic   | 41672      | Plato       | 180    | NULL        |
| fea   | 112        | Cubo        | 148    | 115         |
| fea   | 114        | Cubo        | 243    | 15          |
| imm   | 773c       | Reostato    | 975    | 28          |
| imm   | 775c       | Reostato2   | 1425   | 5           |
| imm   | 779c       | Reostato3   | 1875   | NULL        |
| imm   | 887h       | Caja Cables | 54     | 223         |
| imm   | 887p       | Perno       | 25     | 24          |
| imm   | 887x       | Manivela    | 475    | 32          |
| qsa   | xk47       | Red         | 355    | 38          |
| qsa   | xk48       | Red         | 134    | 203         |
| qsa   | xk48a      | Red         | 117    | 37          |
| rei   | 2a44g      | Pas         | 350    | 14          |
| rei   | 2a44l      | Bomba l     | 4500   | 12          |
| rei   | 2a44r      | Bomba r     | 4500   | 12          |
| rei   | 2a45c      | Junta       | 79     | 210         |

## 1 Lenguaje de consultas SQL I

### 1.1 Cláusulas SELECT y FROM: Consultas básicas (bdBasicas)

#### DIFICULTAD BAJA

- 1.1.1.1 Selecciona todos los datos de la tabla Empleados

SOLUCIÓN:

```
select * from empleados;
```

- 1.1.1.2 Selecciona todos los datos de la tabla Libros

SOLUCIÓN:

```
select * from libros;
```

- 1.1.1.3 Selecciona todos los datos de la tabla Facturas

SOLUCIÓN:

```
select * from facturas;
```

- 1.1.1.4 Selecciona únicamente las columnas nombre, apellidos y puesto de la tabla Empleados

SOLUCIÓN:

```
select nombre, apellidos, puesto from empleados;
```

- 1.1.1.5 Selecciona únicamente las columnas título y precio de la tabla Libros

SOLUCIÓN:

```
select titulo, precio from libros;
```

- 1.1.1.6 Selecciona únicamente las columnas destinatario, importe y fecha\_hora de la tabla Facturas

SOLUCIÓN:

```
select destinatario, importe, fecha_hora from facturas;
```

### 1.2 Cálculos y operadores aritméticos BD (bdBasicas)

#### DIFICULTAD BAJA

- 1.2.1.1 Selecciona los campos nombre, apellidos, sueldo y sueldo multiplicado por 1.2 (Llámalo "Subida de sueldo") de la tabla empleados

SOLUCIÓN:

```
select nombre, apellidos, sueldos, sueldos * 1.2 as "subida de sueldo" from empleados;
```

- 1.2.1.2 Selecciona los campos título, precio y precio multiplicado por 0.75 de la tabla libros. Llama a este último "Precio con descuento"

SOLUCIÓN:

```
select titulo, precio, precio * 0.75 as "precio con descuento" from libros;
```

- 1.2.1.3 Selecciona los campos destinatario, importe e importe más 2.99 de la tabla facturas. Llama a este último "Precio con gastos"

SOLUCIÓN:



```
select destinatario, importe, importe + 2.99 as "precio con gastos de envio" from facturas;
```

### 1.3 Clausula WHERE: Evaluación de condiciones (BD 'bdBasicas')

#### DIFICULTAD BAJA

#### 1.3.1 Operadores de comparación

- 1.3.1.1 Selecciona todos los campos de la tabla empleados cuyo sueldo sea menor que 2000

SOLUCIÓN:

```
select * from empleados where sueldos < 2000;
```

- 1.3.1.2 Selecciona todos los campos de la tabla empleados cuyo puesto sea "encargado"

SOLUCIÓN:

```
select * from empleados where puesto = "encargado";
```

- 1.3.1.3 Selecciona todos los campos de la tabla facturas cuyo importe sea mayor o igual que 100

SOLUCIÓN:

```
select * from facturas where importe >= 100;
```

- 1.3.1.4 Selecciona todos los campos de la tabla libros cuyo tipo sea distinto de "divulgación"

SOLUCIÓN:

```
select * from libros where tipo <> 'divulgación';
```

- 1.3.1.5 Selecciona todos los campos de la tabla empleados cuyo nombre sea distinto de "Luis"  
(No uses "<>")

SOLUCIÓN:

```
select * from empleados where nombre != 'Luis';
```

#### 1.3.2 Operadores lógicos (AND, OR, NOT)

- 1.3.2.1 Selecciona todos los campos de la tabla empleados cuyo sueldo sea menor de 2000 y cuyo nombre sea "David"

SOLUCIÓN:

```
select * from empleados where sueldos < 2000 and nombre = 'David';
```

- 1.3.2.2 Selecciona todos los campos de la tabla libros cuyo precio sea mayor que 20 o cuyo autor sea anónimo

SOLUCIÓN:

```
select * from libros where precio > 20 or autor = 'anónimo';
```

- 1.3.2.3 Selecciona todos los campos de la tabla facturas cuyo importe NO sea menor que 100

SOLUCIÓN:

```
select * from facturas where not importe < 100;
```

- 1.3.2.4 Selecciona todos los campos de la tabla empleados cuyo teléfono sea mayor que 948000000 y cuyo nombre NO sea "Juan"

SOLUCIÓN:

```
select * from empleados where telefono >= 948000000 and not nombre = 'Juan';
```

- 1.3.2.5 Selecciona todos los campos de la tabla empleados cuyo sueldo sea menor que 2000  
o cuyo sueldo sea mayor que 5000

SOLUCIÓN:

```
select * from empleados where sueldos < 2000 or sueldos > 5000;
```

- 1.3.2.6 Selecciona todos los campos de la tabla empleados cuyo sueldo sea menor que 2000  
y cuyo sueldo sea mayor que 5000

SOLUCIÓN:

```
select * from empleados where sueldos < 2000 and sueldos > 5000;
```

- 1.3.2.7 Selecciona todos los campos de la tabla empleados cuyo puesto sea “operario” o  
“supervisor”

SOLUCIÓN:

```
select * from empleados where puesto='operario' or puesto='supervisor';
```

- 1.3.2.8 Selecciona todos los campos de la tabla facturas cuyo importe sea mayor que 50 y que  
no sea mayor que 200

SOLUCIÓN:

```
select * from facturas where importe > 50 and not importe > 200;
```

- 1.3.2.9 Selecciona todos los campos de la tabla libros cuyo precio sea mayor que 30 o menor  
que 15

SOLUCIÓN:

```
select * from libros where precio > 30 or precio < 15;
```

### 1.3.3 BETWEEN

- 1.3.3.1 Selecciona todos los campos de la tabla empleados cuyo teléfono esté entre  
948000000 y 980000000

SOLUCIÓN:

```
select * from empleados where telefono between 948000000 and 980000000;
```

- 1.3.3.2 Selecciona todos los campos de la tabla libros cuyo precio esté entre 15 y 30

SOLUCIÓN:

```
select * from libros where precio between 15 and 30;
```

- 1.3.3.3 Selecciona todos los campos de la tabla libros cuyo ISBN no esté entre 431501467 y  
705940145

SOLUCIÓN:

```
select * from libros where isbn not between 431501467 and 705940145;
```

- 1.3.3.4 Selecciona todos los campos de la tabla facturas cuyo importe no esté entre 50 y 200

SOLUCIÓN:

```
select * from facturas where importe not between 50 and 200;
```

### 1.3.4 LIKE

- 1.3.4.1 Selecciona todos los campos de la tabla empleados cuyos apellidos acaben por la cadena “ez”

SOLUCIÓN:

```
select * from empleados where apellidos like 'ez';
```

- 1.3.4.2 Selecciona todos los campos de la tabla empleados cuyos apellidos contengan la cadena “ez”

SOLUCIÓN:

```
select * from empleados where apellidos like '%ez%';
```

- 1.3.4.3 Selecciona todos los campos de la tabla empleados cuyo nombre acabe por el carácter “a”

SOLUCIÓN:

```
select * from empleados where nombre like '%a';
```

- 1.3.4.4 Selecciona todos los campos de la tabla Facturas cuyo destinatario sea “Alan ” y que después haya un espacio y luego cualquier otro conjunto de caracteres.

SOLUCIÓN:

```
select * from facturas where destinatario like 'Alan %';
```

- 1.3.4.5 Selecciona todos los campos de la tabla facturas cuyo destinatario comience por un conjunto de caracteres cualquiera, que luego haya un espacio, después un carácter cualquiera y luego la cadena de caracteres “eumann”.

SOLUCIÓN:

```
select * from facturas where destinatario like '% _eumann';
```

- 1.3.4.6 Selecciona todos los campos de la tabla libros cuyo título comience por la cadena de caracteres “El ”(ojo al espacio después), luego haya un conjunto de caracteres cualquiera, luego le siga la cadena de caracteres “ de ” (espacio antes y después), tras esto haya un carácter cualquiera, la cadena de caracteres “or”, otro carácter cualquiera y termine con la cadena de caracteres “es”

SOLUCIÓN:

```
select * from libros where titulo like 'El % de _or_es';
```

- 1.3.4.7 Selecciona todos los campos de la tabla libros cuyo título comience por la cadena de caracteres “La”

SOLUCIÓN:

```
select * from libros where titulo like 'La%';
```

### 1.3.5 IS NULL

- 1.3.5.1 Selecciona todos los campos de la tabla empleados cuyo email sea nulo

SOLUCIÓN:

```
select * from empleados where email is null;
```

- 1.3.5.2 Selecciona todos los campos de la tabla libros cuyo autor sea nulo

SOLUCIÓN:

```
select * from libros where autor is null;
```

- 1.3.5.3 Selecciona todos los campos de la tabla empleados cuyo email no sea nulo

SOLUCIÓN:

```
select * from empleados where email is not null;
```

- 1.3.5.4 Selecciona todos los campos de la tabla libros cuyo autor no sea nulo

SOLUCIÓN:

```
select * from libros where autor is not null;
```

## 1.4 Precedencia de operadores (Aritméticos y lógicos) (bdBasicas)

### DIFICULTAD MEDIA

- 1.4.1.1 Realiza la siguiente operación matemática en MySQL: sueldo + 15 + 10 - 5 \* 5 / 5. Usa para ello una consulta a la tabla empleados.
- 1.4.1.2 Realiza la siguiente operación matemática en MySQL: sueldo + (15 + 10) - 5 \* 5 / 5. Usa para ello una consulta a la tabla empleados.
- 1.4.1.3 Realiza la siguiente operación matemática en MySQL: sueldo + 15 + (10 - 5 \* 5) / 5. Usa para ello una consulta a la tabla empleados.
- 1.4.1.4 Realiza la siguiente operación matemática en MySQL: sueldo + 15 + 10 - 5 \* (5 / 5). Usa para ello una consulta a la tabla empleados.
- 1.4.1.5 Realiza la siguiente operación matemática en MySQL: sueldo + 15 + (10 - 5) \* 5 / 5. Usa para ello una consulta a la tabla empleados.
- 1.4.1.6 Realiza la siguiente operación matemática en MySQL: sueldo + (15 + 10 - 5) \* 5 / 5. Usa para ello una consulta a la tabla empleados.
- 1.4.1.7 Realiza la siguiente operación matemática en MySQL: sueldo + 15 + (10 - 5 \* 5 / 5) Usa para ello una consulta a la tabla empleados.

SOLUCIÓN:

```

/* REALIZA LA SIGUIENTE OPERACIÓN MATEMÁTICA: sueldo + 15 + 10 - 5 * 5 / 5 */
select sueldo + 15 + 10 - 5 * 5 / 5 from empleados;
/* REALIZA LA SIGUIENTE OPERACIÓN MATEMÁTICA: sueldo + (15 + 10) - 5 * 5 / 5 */
select sueldo + (15 + 10) - 5 * 5 / 5 from empleados;
/* REALIZA LA SIGUIENTE OPERACIÓN MATEMÁTICA: sueldo + 15 + (10 - 5 * 5) / 5 */
select sueldo + 15 + (10 - 5 * 5) / 5 from empleados;
/* REALIZA LA SIGUIENTE OPERACIÓN MATEMÁTICA: sueldo + 15 + 10 - 5 * (5 / 5) */
select sueldo + 15 + 10 - 5 * (5 / 5) from empleados;
/* REALIZA LA SIGUIENTE OPERACIÓN MATEMÁTICA: sueldo + 15 + (10 - 5) * 5 / 5 */
select sueldo + 15 + (10 - 5) * 5 / 5 from empleados;
/* REALIZA LA SIGUIENTE OPERACIÓN MATEMÁTICA: sueldo + (15 + 10 - 5) * 5 / 5 */
select sueldo + (15 + 10 - 5) * 5 / 5 from empleados;
/* REALIZA LA SIGUIENTE OPERACIÓN MATEMÁTICA: sueldo + 15 + (10 - 5 * 5 / 5) */
select sueldo + 15 + (10 - 5 * 5 / 5) from empleados;

```

- 1.4.1.8 Selecciona los campos nombre, teléfono, sueldo y la suma de estos dos últimos... llama a dicho campo "campo que suma teléfonos y sueldos", de la tabla empleados

SOLUCIÓN:

```
select nombre, telefono, sueldos, (telefono + sueldos) as "campo que suma teléfonos y sueldos" from empleados;
```

- 1.4.1.9 Selecciona los campos título, ISBN, precio y la resta de estos dos últimos, llama a dicho campo "campo que resta el precio al ISBN", de la tabla libros

SOLUCIÓN:

```
select titulo, ISBN, precio, (ISBN - precio) as "campo que resta el precio al ISBN" from libros;
```

- 1.4.1.10 Selecciona los campos destinatario, código, cuenta, importe y la operación ((cuenta – (código \* importe))/importe), llama a este último "campo de prueba", de la tabla facturas

SOLUCIÓN:

```
select destinatario, codigo, cuenta, importe, ((cuenta - (codigo * importe))/importe) as "campo de prueba" from facturas;
```

## 1.5 Funciones sobre filas

### 1.5.1 Funciones para cadenas de caracteres

- 1.5.1.1 Selecciona los campos de la tabla empleados para que se muestren los resultados de la siguiente manera: (bdBasicas)

|                           |
|---------------------------|
| Luis Dorado Pérez         |
| Juan López Pérez          |
| Marta González Ramos      |
| Pedro Santos Ruiz         |
| Ernesto Sevilla           |
| Ramón Hernández Rojo      |
| Laura Garcés Sevilla      |
| Fernando Pastor Hernández |
| María León Sánchez        |
| Pilar Castro López        |
| Vanessa Pérez Melendro    |
| Alba Etxeberria Luis      |
| David Herrero Sánchez     |

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select concat(nombre, ' ', apellidos) from empleados;
```

- 1.5.1.2 Selecciona los campos de la tabla libros para que se muestren los resultados de la siguiente manera: (bdBasicas)

|  |
|--|
| Aquitania: 21.90€                      |
| El principito: 6.95€                   |
| La buena cocina: 24.95€                |
| Redes Locales: 21.89€                  |
| Sapiens: de animales a dioses: 19.95€  |
| El lazarillo de Tormes: 7.86€          |
| Pensar rápido, pensar despacio: 23.49€ |
| La ciudad de vapor: 17.90€             |
| Adiós al frío: 12.00€                  |
| Una tierra prometida: 32.90€           |
| Poema de Mio Cid: 11.15€               |
| Flamenca: 7.95€                        |
| Las mil y una noches: 12.95€           |

### DIFICULTAD BAJA

SOLUCIÓN:

```
select concat(titulo, ': ', precio, '€') from libros;
```

- 1.5.1.3 Selecciona los campos de la tabla facturas para que se muestren los resultados de la siguiente manera: (bdBasicas)

|                           |
|---------------------------|
| Mark Zuckerberg: 846.49€  |
| Jhonny Silverhand: 59.99€ |
| Nikola Tesla: 23.75€      |
| Bolvar Fordragon: 179.96€ |
| Bill Gates: 139.83€       |
| Mark Hamill: 59.99€       |
| John Von Neumann: 49.95€  |
| Alan Turing: 658.75€      |
| Steve Wozniak: 386.02€    |
| Linus Torvalds: 30.19€    |

### DIFICULTAD BAJA

SOLUCIÓN:

```
select concat(destinatario, ': ', importe, '€') from facturas;
```

- 1.5.1.4 Selecciona el campo nombre de la tabla empleados en minúsculas (bdBasicas)

### DIFICULTAD BAJA

SOLUCIÓN:

```
select lower(nombre) from empleados;
```

- 1.5.1.5 Selecciona el campo apellidos de la tabla empleados en minúsculas (bdBasicas)

### DIFICULTAD BAJA



SOLUCIÓN:

```
select lower(apellidos) from empleados;
```

- 1.5.1.6 Selecciona el campo título de la tabla libros en minúsculas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select lower(titulo) from libros;
```

- 1.5.1.7 Selecciona el campo destinatario de la tabla facturas en minúsculas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select lower(destinatario) from facturas;
```

- 1.5.1.8 Selecciona el campo nombre de la tabla empleados en mayúsculas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select upper(nombre) from empleados;
```

- 1.5.1.9 Selecciona el campo apellidos de la tabla empleados en mayúsculas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select upper(apellidos) from empleados;
```

- 1.5.1.10 Selecciona el campo título de la tabla libros en mayúsculas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select upper(titulo) from libros;
```

- 1.5.1.11 Selecciona el campo destinatario de la tabla facturas en mayúsculas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select upper(destinatario) from facturas;
```

- 1.5.1.12 Muestra la longitud de la cadena de caracteres del campo nombre. Usa la tabla personas. (bdPersonas)

**DIFICULTAD BAJA**

```
select nombre, char_length(nombre) from personas;
```

- 1.5.1.13 Muestra los apellidos y los tres primeros caracteres este último lo nombraras como 'Primeras letras' de las personas. (bdPersonas)

**DIFICULTAD BAJA**

```
select apellido, left (apellido,3) as primeras_letras from personas;
```

- 1.5.1.14 Muestra los apellidos y los dos últimos caracteres este último lo nombrarás como 'Últimas letras' de las personas. (bdPersonas)

**DIFICULTAD BAJA**

```
select apellido, right (apellido,2) as ultimas_letras from personas;
```

### 1.5.2 Funciones matemáticas

- 1.5.2.1 Selecciona los campos nombre, apellidos y sueldo, redondeando este último a 1 decimal, de la tabla empleados (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select nombre, apellidos, round(sueldos, 1) from empleados;
```

- 1.5.2.2 Selecciona los campos título y precio, redondeado este último sin decimales, de la tabla libros (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select titulo, round(precio, 0) from libros;
```

- 1.5.2.3 Selecciona los campos destinatario e importe, redondeado este último a 1 decimal, de la tabla facturas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select destinatario, round(importe, 1) from facturas;
```

### 1.5.3 Funciones de fecha y hora

- 1.5.3.1 Selecciona el destinatario y el día del mes de la tabla facturas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select destinatario, dayofmonth(fecha_hora) from facturas;
```

- 1.5.3.2 Selecciona el destinatario y el día de la semana (numérico) de la tabla facturas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select destinatario, dayofweek(fecha_hora) from facturas;
```

- 1.5.3.3 Selecciona el destinatario y el mes (numérico) de la tabla facturas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select destinatario, month(fecha_hora) from facturas;
```

- 1.5.3.4 Selecciona el destinatario y el año de la tabla facturas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select destinatario, year(fecha_hora) from facturas;
```

- 1.5.3.5 Selecciona el destinatario y la hora de la tabla facturas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:



```
select destinatario, time(fecha_hora) from facturas;
```

- 1.5.3.6 Selecciona el destinatario y los minutos de la tabla factura (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select destinatario, minute(fecha_hora) from facturas;
```

- 1.5.3.7 Selecciona el destinatario y la fecha actual de la tabla facturas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select destinatario, curdate() from facturas;
```

- 1.5.3.8 Selecciona el destinatario y el momento actual de la tabla facturas (bdBasicas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select destinatario, now() from facturas;
```

- 1.5.3.9 Selecciona los campos destinatario, fecha\_hora, y la diferencia entre el momento actual y el campo fecha\_hora de la tabla facturas (bdBasicas)

**DIFICULTAD MEDIA**

SOLUCIÓN:

```
select destinatario, fecha_hora, timestampdiff(day, fecha_hora, now()) from facturas;
```

- 1.5.3.10 Selecciona el día de la fecha de alta de las personas de la tabla personas. (bdPersonas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select DAYOFMONTH(FECHA_ALTA) FROM PERSONAS;
```

- 1.5.3.11 Lleva a cabo una consulta que muestre el nombre del día en que se les dio de alta a las personas de la tabla personas. (bdPersonas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select dayname(FECHA_ALTA) FROM PERSONAS;
```

- 1.5.3.12 Lleva a cabo una consulta que muestre el nombre del día en que se les dio de alta y el nombre de esa persona. (bdPersonas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select NOMBRE, dayname(FECHA_ALTA) FROM PERSONAS;
```

- 1.5.3.13 Lleva a cabo una consulta en la que se muestre el nombre del mes el que fueron dados de baja los empleados que pertenecían a la sede AB. (bdPersonas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select monthname(FECHA_BAJA) FROM PERSONAS WHERE SEDE_TRABAJA LIKE 'AB';
```

- 1.5.3.14 Lleva a cabo una consulta en la que se muestre los meses que han pasado entre la fecha de alta y la fecha actual y el nombre de cada una de las personas de ellas, solo tienes que tener en cuenta aquellas que sigan trabajando en la actualidad. (bdPersonas)

**DIFICULTAD MEDIA**

SOLUCIÓN:

```
select NOMBRE,timestampdiff(MONTH,FECHA_ALTA,NOW()) FROM PERSONAS WHERE FECHA_BAJA is not null;
```

- 1.5.3.15 Indica la fecha (el nombre del mes y día), nombre en que se dieron de baja aquellas personas que están jubiladas. Atentos al encabezado de la tabla resultado. (bdPersonas)

**DIFICULTAD BAJA**

SOLUCIÓN:

```
select nombre, monthname(FECHA_BAJA),dayname(FECHA_BAJA) from PERSONAS where puesto like 'JUBILADO';
```

- 1.5.3.16 Indica el nombre y los años de antigüedad (años de alta) a día de hoy. Atentos al encabezado de la tabla resultado. (bdPersonas)

| nombre | Antigüedad |
|--------|------------|
| CESAR  | 11         |
| CAMILO | 11         |
| JOSE   | 10         |
| LARA   | 10         |
| PILAR  | 20         |
| PEDRO  | 20         |

**DIFICULTAD MEDIA**

SOLUCIÓN:

```
SELECT nombre, timestampdiff(YEAR, fecha_alta, CURDATE()) as Antigüedad FROM personas;
```

- 1.5.3.17 Indica el nombre y los años de antigüedad (años de alta) que tenían las personas el día 1/06/2020. (bdPersonas)

| nombre | Antigüedad el 01/06/2020 |
|--------|--------------------------|
| CESAR  | 12                       |
| CAMILO | 12                       |
| JOSE   | 11                       |
| LARA   | 11                       |
| PILAR  | 21                       |
| PEDRO  | 21                       |

**DIFICULTAD MEDIA**

SOLUCIÓN:

```
SELECT nombre, timestampdiff(YEAR, fecha_alta, '2021-6-1') as 'Antigüedad el 01/06/2020' FROM personas;
```

- 1.5.3.18 Muestra el nombre del mes (sin que se repitan) en el que los empleados cuyo puesto es 'JUBILADO' y la sede en la que trabaja es la AC fueron dados de alta. (bdPersonas)

**DIFICULTAD BAJA**

```
select distinct(monthname(FECHA_ALTA)) FROM personas where sede_trabaja = 'AC' and PUESTO LIKE 'JUBILADO';
```

- 1.5.3.19 Muestra el nombre del día en el que se dieron de baja los empleados que nacieron después del año 1979. (bdPersonas)

**DIFICULTAD BAJA**

```
select distinct(dayname(FECHA_BAJA)) FROM personas where Año_NACIMIENTO>1979;
```

- 1.5.3.20 Muestra el fecha con el formato 'El día de mes del año' de la fecha en la que se dieron de baja las personas. (bdPersonas)

**DIFICULTAD MEDIA**

```
select date_format(fecha_alta, 'El %W de %M del %Y') from personas;
```

#### 1.5.4 Anidamiento de funciones (bdBasicas)

- 1.5.4.1 Muestra el nombre completo en mayúsculas en una sola columna como en el ejemplo. Usa la tabla empleados.

| Nombre en mayúsculas |
|----------------------|
| LUIS DORADO PÉREZ    |
| JUAN LÓPEZ PÉREZ     |
| MARTA GONZÁLEZ RAMOS |
| PEDRO SANTOS RUIZ    |
| ERNESTO SEVILLA      |

**DIFICULTAD ALTA**

SOLUCIÓN:

```
SELECT upper(concat(nombre,' ',apellidos)) as 'Nombre en mayúsculas' FROM empleados;
```

- 1.5.4.2 Muestra la primera letra del nombre, seguida de un punto, un espacio y los apellidos. Pero solo de aquellos cuyo **segundo** apellido sea 'Pérez'. Usa la tabla empleados.

| Inicial y apellidos |
|---------------------|
| L. Dorado Pérez     |
| J. López Pérez      |

**DIFICULTAD ALTA**

SOLUCIÓN:

```
select concat(left(nombre,1),'. ',apellidos) from empleados where apellidos LIKE '%Pérez';
```

#### 1.6 Ordenación en los resultados (bdPersonas)

**DIFICULTAD BAJA**

- 1.6.1.1 Selección el nombre y apellidos de las personas cuyo salario sea igual a 900 euros y ordénalos por el año de nacimiento.

SOLUCIÓN:

```
Select NOMBRE, APELLIDO FROM PERSONAS WHERE SALARIO=900 ORDER BY Año_NACIMIENTO;
```

- 1.6.1.2 Selecciona el apellido de las personas que trabajan en la sede AB y que actualmente estén trabajando. Ordena el resultado por nombre.

SOLUCIÓN:

```
Select APELLIDO FROM PERSONAS WHERE FECHA_BAJA IS NULL AND SEDE_TRABAJA LIKE 'AB' ORDER BY NOMBRE;
```

- 1.6.1.3 Selecciona el NIF y NOMBRE de la tabla PERSONAS cuyo año de nacimiento sea mayor o igual a 1978 y el salario mayor a 950 ordénalos por orden de apellido de manera descendente.

SOLUCIÓN:

```
Select NIF, NOMBRE FROM PERSONAS WHERE AÑO_NACIMIENTO >=1978 AND SALARIO > 950 ORDER BY APELLIDO DESC;
```

## 1.7 Limitación de resultados (bdPersonas)

- 1.7.1.1 Ordena las personas por el sueldo de menor a mayor y muestra las tres primeras más mayores.

SOLUCIÓN:

```
Select NOMBRE, APELLIDO, AÑO_NACIMIENTO FROM PERSONAS ORDER BY SALARIO, AÑO_NACIMIENTO LIMIT 3;
```

- 1.7.1.2 Ordena las personas cuyo apellido empiece por la letra A por la fecha de alta y muestra las dos con más antigüedad.

SOLUCIÓN:

```
SELECT NOMBRE, APELLIDO FROM PERSONAS WHERE APELLIDO LIKE 'A%' ORDER BY FECHA_ALTA LIMIT 2;
```

## 1.8 Funciones de agregado (bdPersonas)

### DIFICULTAD BAJA

- 1.8.1.1 Muestra la suma de los salarios de las personas que están en el departamento AC.

SOLUCIÓN:

```
select SUM(SALARIO) from personas where SEDE_TRABAJA LIKE 'AC';
```

- 1.8.1.2 Muestra la media de los salarios de las personas que están en el departamento AC y son operarios.

SOLUCIÓN:

```
select AVG(SALARIO) FROM personas WHERE SEDE_TRABAJA LIKE 'AC' AND PUESTO LIKE 'OPERARIO';
```

- 1.8.1.3 Cuenta el número de personas que están jubiladas.

SOLUCIÓN:

```
select COUNT(*) FROM personas WHERE PUESTO LIKE 'JUBILADO';
```

- 1.8.1.4 Muestra el año de nacimiento del jubilado más mayor del departamento AC.

SOLUCIÓN:

```
select MIN(AÑO_NACIMIENTO) FROM personas WHERE PUESTO LIKE 'JUBILADO' AND SEDE_TRABAJA LIKE 'AC';
```

## 1.9 GROUP BY

- 1.9.1.1 Muestra el número de empleados que hay por cada una de las sedes. (bdPersonas)  
**DIFICULTAD BAJA**

SOLUCIÓN:

```
select count(*) from personas group by SEDE_TRABAJA;
```

- 1.9.1.2 Cuenta los tipos de puestos de trabajo que existen en cada una de las sedes. (bdPersonas)  
**DIFICULTAD BAJA**

SOLUCIÓN:

```
SELECT SEDE_TRABAJA, COUNT(distinct PUESTO) FROM personas GROUP BY (SEDE_TRABAJA);
```

- 1.9.1.3 Seleccionar la suma de sueldos por puesto, independiente de la sede (bdPersonas)  
**DIFICULTAD BAJA**

```
select puesto,sum(salario) from personas  
group by puesto;
```

error tarea no 3, si 6

- 1.9.1.4 Seleccionar la suma de sueldos por puesto y por sede (bdPersonas)  
**DIFICULTAD BAJA**

```
select puesto,sede_trabaja, sum(salario) from personas  
group by puesto,sede_trabaja;
```

- 1.9.1.5 Seleccionar cuántas personas hay por cada puesto y sede (bdPersonas)  
**DIFICULTAD BAJA**

```
select puesto,sede_trabaja,count(*) as 'nº personas' from personas  
group by puesto,sede_trabaja;
```

- 1.9.1.6 Seleccionar la cantidad de personas clasificadas por año de nacimiento (bdPersonas)  
**DIFICULTAD BAJA**

```
select año_nacimiento,count(*) from personas  
group by año_nacimiento;
```

- 1.9.1.7 Seleccionar la fábrica que tiene una media de año de nacimiento más joven (bdPersonas)  
**DIFICULTAD BAJA**

```
select sede_trabaja,avg(año_nacimiento) from personas  
group by sede_trabaja;
```

- 1.9.1.8 Seleccionar Año y nº de altas ese año (bdPersonas)  
**DIFICULTAD MEDIA**

```
select year(fecha_alta) as Año, count(*) as 'Nº altas' from personas  
group by year(fecha_alta);
```

- 1.9.1.9 Seleccionar Año y nº de altas en ese año de personas nacidas después del año 1980 (bdPersonas)

**DIFICULTAD MEDIA**

```
select year(fecha_alta),count(*) from personas
where (año_nacimiento>1980)
group by year(fecha_alta);
```

- 1.9.1.10 Visualizar cada oficio con el su salario máximo, mínimo y medio, proporcionándole alias a cada columna. (bdTrabajadores)

**DIFICULTAD MEDIA**

```
SELECT oficio, max(salario) as Salario_Máximo, min(salario) as Salario_Mínimo, avg (salario) as Salario_Medio
FROM trabajadores GROUP BY oficio;
```

- 1.9.1.11 Visualizar cuántos trabajadores por departamento y oficio. Usa alias para la columna agregada. Ordénalo por departamento y luego por oficio. (bdTrabajadores)

**DIFICULTAD MEDIA**

```
SELECT dept_no, oficio, count(emp_no) as 'Num. Trabajadores' FROM trabajadores
GROUP BY dept_no, oficio order by dept_no, oficio;
```

- 1.9.1.12 Selecciona el departamento que más trabajadores tiene y su número de trabajadores. Usa alias. (bdTrabajadores)

**DIFICULTAD MEDIA**

```
SELECT dept_no, count(emp_no) as 'Num Personas' FROM trabajadores
GROUP BY dept_no ORDER BY count(emp_no) DESC LIMIT 1;
```

- 1.9.1.13 Oficio que más trabajadores tiene. (bdTrabajadores)

**DIFICULTAD MEDIA**

```
SELECT oficio FROM trabajadores GROUP BY oficio ORDER BY count(oficio)
DESC LIMIT 1;
```

- 1.9.1.14 Resumir los trabajadores (cuántos y salario medio) por año de fecha de alta en la empresa. Ordenar por fecha descendente. (bdTrabajadores)

**DIFICULTAD MEDIA**

```
SELECT count(*), avg(salario), year(fecha_alt)
FROM trabajadores GROUP BY year(fecha_alt) DESC;
```

- 1.9.1.15 Departamento(s) con mayor número de trabajadores que sean EMPLEADO y su nº de trabajadores. (bdTrabajadores)

**DIFICULTAD MEDIA**

```
SELECT dept_no, count(*) FROM trabajadores WHERE oficio="EMPLEADO"
GROUP BY dept_no ORDER BY count(oficio) DESC LIMIT 1;
```

## 1.10 HAVING (bdPersonas)

- 1.10.1.1 Muestra las sedes que tengan más de tres empleados trabajando para ellas. (bdPersonas)

**DIFICULTAD BAJA**

```
select sede_trabaja, count(NIF) from personas group by (sede_trabaja) HAVING count(NIF)>3;
```

- 1.10.1.2 Muestra aquellos puestos de trabajo cuya suma de todos los empleados supere los 5000 euros. (bdPersonas)

**DIFICULTAD BAJA**

```
select puesto, sum(salario) from personas group by puesto having sum(salario)>5000;
```

- 1.10.1.3 Seleccionar el puesto que contenga en su nombre la palabra 'JUBILADO' con la suma total de los salarios de las personas de ese puesto. (bdPersonas)

**DIFICULTAD BAJA**

```
select puesto, sum(salario) from personas where puesto like 'JUBILADO'
group by puesto having sum(salario)>3000;
```

- 1.10.1.4 Seleccionar la suma de sueldos por puesto, independiente de la sede, cuando la suma supera 5000 (bdPersonas)

**DIFICULTAD BAJA**

```
select puesto, sum(salario) from personas
group by puesto
having sum(salario)>5000;
```

- 1.10.1.5 Muestra que puesto de trabajo tienen una media de sueldo entre 800 y 1200 independientemente de la sede en la que trabaje. (bdPersonas)

**DIFICULTAD MEDIA**

```
select puesto, avg(salario) from personas group by (puesto) having avg(salario)>800 and avg(salario)<1200;
```

- 1.10.1.6 Seleccionar Año en el que se han dado de alta 2 o más personas nacidas después de 1980 (bdPersonas)

**DIFICULTAD MEDIA**

```
select year(fecha_alta), count(*) from personas
where (año_nacimiento>1980)
group by year(fecha_alta)
having count(*) >=2;
```



## 2 Lenguaje de consultas SQL II

### 2.1 BD 'bdTrabajadoresEdificios'

#### 2.1.1 Repaso consultas sobre una sola tabla

##### DIFICULTAD MEDIA

- 2.1.1.1 ¿Cuál es la tarifa semanal de cada electricista? (Suponemos la jornada semanal de 8 horas de lunes a viernes y tenemos el campo de tarifa/hora). Debemos de ordenar el resultado por el nombre del trabajador. Y el formato del resultado debe de ser el siguiente:

|                |                |
|----------------|----------------|
| nom_trabajador | Tarifa Semanal |
|----------------|----------------|

```
USE bdTrabajadorEdificio;
```

```
SELECT nom_trabajador, tarifa_hr*40 as 'Tarifa Semanal' from trabajador where oficio like 'Electricista' order by nom_trabajador asc;
```

- 2.1.1.2 Muestra el número de trabajadores que hay por cada uno de los oficios.

```
USE bdTrabajadorEdificio;
```

```
SELECT oficio, count(*) from trabajador group by oficio;
```

- 2.1.1.3 Por cada supervisor muestra la tarifa por horas más alta que se le paga a un trabajador de ese supervisor.

```
USE bdTrabajadorEdificio;
```

```
SELECT id_supv, MAX(tarifa_hr) as 'tarifa_maxima' from trabajador group by id_supv;
```

- 2.1.1.4 Para cada tipo de edificio, muestra cuál es el nivel de calidad medio para los edificios de categoría 1.

```
USE bdTrabajadorEdificio;
```

```
select tipo, avg(nivel_calidad) as 'nivel medio de calidad' from edificio where categoria=1 group by tipo;
```

- 2.1.1.5 Para cada supervisor que dirige a más de un trabajador ¿Cuál es la tarifa por horas más alta que se le paga a un trabajador de ese supervisor?

```
USE bdTrabajadorEdificio;
```

```
select id_supv , max(tarifa_hr) as 'tarifa máxima' from trabajador group by id_supv having count(*)>1 ;
```

- 2.1.1.6 Para cada tipo de edificio, ¿cuál es el nivel de calidad medio para los edificios de categoría 1? Solo visualizamos los tipos de edificio que tienen un nivel calidad máximo de 3.

```
select tipo, avg(nivel_calidad) as 'Nivel medio de calidad', max(nivel_calidad) as 'Nivel calidad máxima 3' from edificio where categoria=1 group by tipo having max(nivel_calidad)=3;
```



### 2.1.2 Consultas INNER JOIN

**2.1.2.1** Indica los nombres de los trabajadores con el nombre de los supervisores.

#### **DIFICULTAD BAJA**

```
USE bdTrabajadorEdificio;
select a.nom_trabajador as 'nombre trabajador', b.nom_trabajador as 'nombre supervisor'
from trabajador a inner join trabajador b on b.id_trabajador=a.id_supv;
```

**2.1.2.2** Indica los nombres de los trabajadores que sean asignados a edificios que sean oficinas.

#### **DIFICULTAD BAJA**

```
USE bdTrabajadorEdificio;
select distinct(a.nom_trabajador) from trabajador a, asignacion b, edificio e where a.id_trabajador= b.id_trabajador and b.id_edificio = e.id_edificio
and e.tipo = 'Oficina';
/*usando solo agrupacion*/
select a.nom_trabajador from trabajador a, asignacion b, edificio e where a.id_trabajador= b.id_trabajador
and b.id_edificio = e.id_edificio and e.tipo = 'Oficina' group by a.nom_trabajador;
```

**2.1.2.3** Mostrar los trabajadores asignados a edificios mostrando el nombre del trabajador, el número de edificios que tiene asignados y la calidad media de los edificios que tiene asignados. Ordenarlo por el número de edificios en ascendente y, en segundo nivel, por la calidad media en ascendente.

| NOM_TRABAJADOR | NUM_EDIFICIOS | CALIDAD_MEDIA |
|----------------|---------------|---------------|
| C Coulomb      | 2             | 2.0000        |
| H Rickover     | 2             | 2.5000        |
| M Faraday      | 2             | 2.5000        |
| P Mason        | 2             | 3.0000        |
| J Barrister    | 2             | 3.5000        |
| R Garret       | 3             | 2.3333        |
| C Nemo         | 6             | 2.6667        |

#### **DIFICULTAD ALTA**

```
SELECT TRABAJADOR.NOM_TRABAJADOR,
       COUNT(*) AS NUM_EDIFICIOS,
       AVG(EDIFICIO.NIVEL_CALIDAD) AS CALIDAD_MEDIA
FROM   (TRABAJADOR INNER JOIN ASIGNACION
        ON TRABAJADOR.ID_TRABAJADOR = ASIGNACION.ID_TRABAJADOR)
INNER JOIN EDIFICIO
        ON EDIFICIO.ID_EDIFICIO=ASIGNACION.ID_EDIFICIO
GROUP BY TRABAJADOR.NOM_TRABAJADOR
ORDER BY NUM_EDIFICIOS, CALIDAD_MEDIA DESC;
```

### 2.1.3 Consultas LEFT/RIGHT JOIN

- 2.1.3.1 Ídem que el anterior, pero mostrando TODOS los trabajadores, aunque no tengan edificios asignados.

| NOM_TRABAJADOR | NUM_EDIFICIOS | CALIDAD_MEDIA |
|----------------|---------------|---------------|
| L Dorado       | 0             | NULL          |
| J Barrister    | 2             | 3.5000        |
| P Mason        | 2             | 3.0000        |
| H Rickover     | 2             | 2.5000        |
| M Faraday      | 2             | 2.5000        |
| C Coulomb      | 2             | 2.0000        |
| R Garret       | 3             | 2.3333        |
| C Nemo         | 6             | 2.6667        |

#### DIFICULTAD ALTA

```
SELECT TRABAJADOR.NOM_TRABAJADOR,
COUNT(EDIFICIO.ID_EDIFICIO) AS NUM_EDIFICIOS, AVG(EDIFICIO.NIVEL_CALIDAD) AS CALIDAD_MEDIA
FROM (TRABAJADOR LEFT JOIN ASIGNACIÓN
ON TRABAJADOR.ID_TRABAJADOR = ASIGNACIÓN.ID_TRABAJADOR)
LEFT JOIN EDIFICIO
ON EDIFICIO.ID_EDIFICIO=ASIGNACIÓN.ID_EDIFICIO
GROUP BY TRABAJADOR.NOM_TRABAJADOR
ORDER BY NUM_EDIFICIOS, CALIDAD_MEDIA DESC;
```

- ✓ Es importante señalar que se debe usar un DOS LEFT JOIN ya que, en caso de usar un LEFT JOIN y luego un INNER JOIN, el INNER JOIN se carga el trabajador que no tiene ninguna asignación.
- ✓ También debes fijar que, en este caso, no incluimos un \* en el COUNT(). Si lo hiciéramos la columna NUM\_EDIFICIOS de L Dorado marcaría 1. Esto es debido a que COUNT(\*) cuenta el número de apariciones de L Dorado en la consulta sin agrupar. Como queremos contar el número de edificios asignados usamos la clave primaria de la tabla Edificios.

## 2.2 BD 'bdEmpleadosOficinas'

### 2.2.1 Repaso consultas sobre una sola tabla

- 2.2.1.1 Obtener una lista de todos los productos indicando para cada uno su idfab, idproducto, descripción, precio y precio con I.V.A. incluido (es el precio anterior aumentado en un 16%).

#### DIFICULTAD BAJA

```
select idfab, idproducto, descripcion, precio, precio * 1.16 as "iva_incluido" from productos;
```

- 2.2.1.2 De cada pedido queremos saber su número de pedido, fab, producto, cantidad, precio unitario (importe entre cantidad) e importe.

#### DIFICULTAD BAJA

```
select numpedido, fab, producto, cant, importe/cant as "precio_unitario", importe from pedidos;
```

- 2.2.1.3 Listar de cada empleado su nombre, nº de días que lleva trabajando en la empresa y su año de nacimiento (suponiendo que este año ya ha cumplido años).

#### DIFICULTAD MEDIA

```
select nombre, timestampdiff(day, contrato, curdate()) as "días_trabajados", year(curdate())-edad as "año_nacimiento" from empleados;
```

- 2.2.1.4 Listar las cuatro líneas de pedido más caras (las de mayor importe)  
**DIFICULTAD BAJA**

```
select * from pedidos order by importe desc limit 4;
```

- 2.2.1.5 Listar toda la información de los pedidos de marzo  
**DIFICULTAD BAJA**

```
select * from pedidos where month(fechapedido) = 3;
```

- 2.2.1.6 Listar los números de los empleados que tienen una oficina asignada.  
**DIFICULTAD BAJA**

```
select numemp from empleados where oficina is not null;
```

- 2.2.1.7 ¿Cuál es la cuota media y las ventas medias de todos los empleados?  
**DIFICULTAD BAJA**

```
select avg(cuota) as "cuota media", avg(ventas) as "ventas medias" from empleados;
```

- 2.2.1.8 Hallar el importe medio de pedidos, el importe total de pedidos y el precio medio de venta (el precio de venta es el precio unitario en cada pedido).  
**DIFICULTAD BAJA**

```
select avg(importe) as "importe medio", sum(importe) as "Importe total", avg(importe/cant) as "Precio medio de venta" from pedidos;
```

- 2.2.1.9 Hallar en qué fecha se realizó el primer pedido (suponiendo que en la tabla de pedidos tenemos todos los pedidos realizados hasta la fecha).  
**DIFICULTAD BAJA**

```
select min(fechapedido) as "Primer pedido" from pedidos;
```

## 2.2.2 Consultas INNER JOIN, LEFT JOIN y otros

- 2.2.2.1 Listar las oficinas de la región 'este' indicando para cada una de ellas su número de oficina, región, ciudad, números y nombres de sus empleados. Hacer una versión en la que aparecen sólo las que tienen empleados, y hacer otra en las que aparezcan también las oficinas del este que no tienen empleados.

**DIFICULTAD MEDIA**

```
select oficinas.oficina, oficinas.region, ciudad, numemp, nombre from
oficinas inner join empleados on oficinas.oficina = empleados.oficina where region = "este";
select oficinas.oficina, oficinas.region, ciudad, numemp, nombre from
oficinas left join empleados on oficinas.oficina = empleados.oficina where region = "este";
```

- 2.2.2.2 Listar los pedidos mostrando su número, importe, nombre del cliente, y el límite de crédito del cliente correspondiente (todos los pedidos tienen cliente y representante).  
**DIFICULTAD BAJA**

```
select numpedido, importe, nombre, limitecredito from
pedidos inner join clientes on pedidos.clie = clientes.numclie;
```

- 2.2.2.3 Listar los pedidos superiores a 2500 euros, incluyendo el nombre del empleado que tomó el pedido y el nombre del cliente que lo solicitó.

**DIFICULTAD MEDIA**

```
select numpedido, empleados.nombre as "Representante", clientes.nombre as "Cliente" from
(pedidos inner join empleados on pedidos.rep = empleados.numemp) inner join clientes on
pedidos.clie = clientes.numclie where pedidos.importe > 2500;
```

- 2.2.2.4 Listar los empleados con una cuota superior a la de su jefe; para cada empleado sacar su nombre, cuota y jefe (nombre y cuota del jefe).

#### **DIFICULTAD BAJA**

```
select empleados.nombre, empleados.cuota, jefes.nombre as "nombre jefe", jefes.cuota as "cuota jefe" from
empleados inner join jefes on empleados.jefe = jefes.numemp where empleados.cuota > jefes.cuota;
```

- 2.2.2.5 Mostrar los empleados (nombre y región de su oficina) adscritos a una oficina que contenga 'este' en su región y cuyas ventas sean menores a 300.000. De aquellos que estén involucrados en pedidos, mostrar la fecha e importe de sus pedidos. Ordenar por región ascendentemente y, después, por ventas descendentemente.

#### **DIFICULTAD MEDIA**

```
select empleados.nombre, oficinas.region, empleados.ventas, pedidos.fechapedido, pedidos.importe from
(oficinas inner join empleados on oficinas.oficina = empleados.oficina)
left join pedidos on empleados.numemp = pedidos.rep where oficinas.region like '%este%'
and empleados.ventas < 300000 order by region asc, ventas desc;
```

- 2.2.2.6 Listar cuántos empleados están asignados a cada región, indicar el nombre de región de y cuántos hay asignados.

#### **DIFICULTAD BAJA**

```
select oficinas.region, count(*) as "Cuantos empleados" from empleados
inner join oficinas on empleados.oficina = oficinas.oficina group by oficinas.region;
```

- 2.2.2.7 Listar de cada producto, su descripción, precio, cantidad total pedida, existencias y porcentaje del stock pedido, incluyendo sólo los productos cuya cantidad total pedida sea superior al 85% del stock; y ordenado por cantidad total pedida en descendente.

#### **DIFICULTAD ALTA**

```
select productos.descripcion, productos.precio, sum(pedidos.cant) as "cantidad total",
productos.existencias, sum(pedidos.cant)*100/productos.existencias as "Porcentaje pedido"
from productos inner join pedidos on pedidos.fab = productos.idfab and pedidos.producto
= productos.idproducto group by productos.descripcion, productos.precio, productos.existencias
having sum(pedidos.cant)>productos.existencias*0.85 order by sum(pedidos.cant) desc;
```

## **2.3 BD 'bdGestionVentas'**

### **2.3.1 Consultas LEFT JOIN, RIGHT JOIN y otros**

- 2.3.1.1 Devuelve un listado con todos los comerciales junto con los datos de los pedidos que han realizado. Este listado también debe incluir los comerciales que no han realizado ningún pedido. El listado debe estar ordenado alfabéticamente por el primer apellido, segundo apellido y nombre de los comerciales.

#### **DIFICULTAD BAJA**

```

/* LEFT JOIN */
SELECT *
FROM comercial LEFT JOIN pedido
ON comercial.id = pedido.id_comercial
ORDER BY comercial.apellido1, comercial.apellido2, comercial.nombre;

/* RIGHT JOIN */
SELECT *
FROM pedido RIGHT JOIN comercial
ON comercial.id = pedido.id_comercial
ORDER BY comercial.apellido1, comercial.apellido2, comercial.nombre;

```

- 2.3.1.2 Devuelve un listado que solamente muestre los clientes que no han realizado ningún pedido.

**DIFICULTAD MEDIA**

```

/* LEFT JOIN */
SELECT *
FROM cliente LEFT JOIN pedido
ON cliente.id = pedido.id_cliente
WHERE pedido.id_cliente IS NULL;

/* RIGHT JOIN */
SELECT *
FROM pedido RIGHT JOIN cliente
ON cliente.id = pedido.id_cliente
WHERE pedido.id_cliente IS NULL;

```

- 2.3.1.3 Devuelve un listado que solamente muestre los comerciales que no han realizado ningún pedido.

**DIFICULTAD MEDIA**

```

/* LEFT JOIN */
SELECT *
FROM comercial LEFT JOIN pedido
ON comercial.id = pedido.id_comercial
WHERE pedido.id_cliente IS NULL;

/* RIGHT JOIN */
SELECT *
FROM pedido RIGHT JOIN comercial
ON comercial.id = pedido.id_comercial
WHERE pedido.id_comercial IS NULL;

```

- 2.3.1.4 Calcula cuál es el máximo valor de los pedidos realizados durante el mismo día para cada uno de los clientes, teniendo en cuenta que sólo queremos mostrar aquellos pedidos que superen la cantidad de 2000 €.

**DIFICULTAD MEDIA**

```

SELECT cliente.id, cliente.nombre, cliente.apellido1,
       cliente.apellido2, pedido.fecha, MAX(pedido.total)
FROM cliente INNER JOIN pedido
ON cliente.id = pedido.id_cliente
GROUP BY cliente.id, pedido.fecha
HAVING MAX(total) > 2000;

```

- 2.3.1.5 Devuelve un listado con el identificador de cliente, nombre y apellidos y el número total de pedidos que ha realizado cada uno de clientes durante el año 2017.

**DIFICULTAD MEDIA**

```
SELECT cliente.id, cliente.nombre, cliente.apellido1, cliente.apellido2, COUNT(pedido.id)
FROM cliente LEFT JOIN pedido
ON cliente.id = pedido.id_cliente
WHERE YEAR(pedido.fecha) = 2017
GROUP BY cliente.id;
```

- 2.3.1.6 *AMPLIACIÓN: Devuelve un listado con los clientes que no han realizado ningún pedido y de los comerciales que no han participado en ningún pedido. Ordene el listado alfabéticamente por los apellidos y el nombre. En el listado deberá diferenciar de algún modo los clientes y los comerciales.*

**DIFICULTAD MEDIA**

```
SELECT cliente.apellido1, cliente.apellido2, cliente.nombre
FROM cliente LEFT JOIN pedido
ON cliente.id = pedido.id_cliente
WHERE pedido.id_cliente IS NULL

UNION

SELECT comercial.apellido1, comercial.apellido2, comercial.nombre
FROM comercial LEFT JOIN pedido
ON comercial.id = pedido.id_comercial
WHERE pedido.id_comercial IS NULL;
```

## 2.4 BD 'bdTiendaInformatica'

### 2.4.1 Consultas LEFT JOIN, RIGHT JOIN y otros

- 2.4.1.1 Muestra el nombre de cada fabricante, junto con el precio máximo, precio mínimo, precio medio y el número total de productos de los fabricantes que tienen un precio medio superior a 200€. Es necesario mostrar el nombre del fabricante.

**DIFICULTAD MEDIA**

```
SELECT fabricante.nombre,
       MAX(producto.precio), MIN(producto.precio),
       AVG(producto.precio), COUNT(*)
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
GROUP BY fabricante.codigo
HAVING AVG(producto.precio) > 200;
```

- 2.4.1.2 Devuelve un listado con los nombres de los fabricantes que tienen 2 o más productos.

**DIFICULTAD MEDIA**

```
SELECT fabricante.nombre, COUNT(producto.codigo)
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
GROUP BY fabricante.codigo
HAVING COUNT(producto.codigo) >= 2;
```





### 3 Lenguaje de consultas SQL III

#### 3.1 BD 'bdTrabajadoresEdificios'

##### 3.1.1 Subconsultas simples de comparación

- 3.1.1.1 Qué trabajadores reciben una tarifa por hora mayor que la del promedio

**DIFICULTAD BAJA**

```
SELECT NOM_TRABAJADOR
FROM TRABAJADOR
WHERE TARIFA_HR > (SELECT AVG(TARIFA_HR) FROM TRABAJADOR);
```

##### 3.1.2 Subconsultas simples de pertenencia

- 3.1.2.1 Indicar los nombres de los trabajadores asignados a edificios que sean oficinas. Realizar la consulta utilizando INNER JOIN en la consulta principal y **después usando una subconsulta**.

**DIFICULTAD BAJA**

```
SELECT DISTINCT NOM_TRABAJADOR
FROM TRABAJADOR T INNER JOIN ASIGNACION A
ON T.ID_TRABAJADOR=A.ID_TRABAJADOR
WHERE A.ID_EDIFICIO IN (SELECT ID_EDIFICIO FROM EDIFICIO
                        WHERE TIPO='Oficina');

SELECT NOM_TRABAJADOR
FROM TRABAJADOR T
WHERE T.ID_TRABAJADOR IN (SELECT A.ID_TRABAJADOR
                        FROM EDIFICIO E INNER JOIN ASIGNACION A
ON E.ID_EDIFICIO=A.ID_EDIFICIO
                        WHERE TIPO='Oficina');
```

##### 3.1.3 Subconsultas correlacionadas

- 3.1.3.1 Indicar los trabajadores que reciben una tarifa por hora mayor que la de su supervisor. Haz tres versiones: una con INNER JOIN, otra con subconsulta correlacionada y otra con EXISTS.

**DIFICULTAD MEDIA**

```
/* subconsulta correlacionada*/
SELECT NOM_TRABAJADOR
FROM TRABAJADOR A
WHERE A.TARIFA_HR > (SELECT B.TARIFA_HR FROM TRABAJADOR B
                    WHERE A.ID_SUPV=B.ID_TRABAJADOR);
```

```

/* con exists*/
SELECT NOM_TRABAJADOR
FROM TRABAJADOR A
WHERE EXISTS (SELECT * FROM TRABAJADOR B
              WHERE A.ID_SUPV=B.ID_TRABAJADOR
              AND A.TARIFA_HR > B.TARIFA_HR);

/* con innner join*/
SELECT A.NOM_TRABAJADOR
FROM TRABAJADOR A INNER JOIN TRABAJADOR B ON A.ID_SUPV=B.ID_TRABAJADOR
WHERE A.TARIFA_HR>B.TARIFA_HR

```

- 3.1.3.2 Indicar los ID de trabajador que no están asignados al edificio 435; hacedlo con el predicado IN y posteriormente con el predicado EXISTS.

**DIFICULTAD MEDIA**

```

SELECT ID_TRABAJADOR
FROM TRABAJADOR
WHERE ID_TRABAJADOR NOT IN (SELECT DISTINCT ID_TRABAJADOR
                           FROM ASIGNACIoN
                           WHERE ID_EDIFICIO=435);

/*con exists*/
SELECT ID_TRABAJADOR
FROM TRABAJADOR T
WHERE NOT EXISTS (SELECT *
                  FROM ASIGNACIoN ASI
                  WHERE ASI.ID_TRABAJADOR=T.ID_TRABAJADOR
                  AND ASI.ID_EDIFICIO=435);

```

## 3.2 BD 'bdEmpleadosOficinas'

### 3.2.1 Subconsultas simples de comparación

- 3.2.1.1 Listar los nombres de los clientes que tienen asignado el representante Alvaro Jaumes (suponiendo que no puede haber representantes con el mismo nombre). Implementa dos versiones: una con JOIN y otra con subconsultas.

**DIFICULTAD BAJA**

```

-- Con Inner Join
select clientes.nombre from clientes inner join empleados on clientes.repclie = empleados.numemp
where empleados.nombre = "Alvaro Jaumes";

-- Con Subconsulta
select nombre from clientes where repclie =
(select numemp from empleados where nombre = "Alvaro Jaumes");

```

### 3.2.2 Subconsultas simples de pertenencia



- 3.2.2.1 Listar los vendedores (numemp, nombre, y nº de oficina) que trabajan en oficinas "buenas" (las que tienen ventas superiores a su objetivo). Implementa dos versiones: una con JOIN y **otra con subconsultas**.

**DIFICULTAD BAJA**

```
-- Con Inner Join
select numemp, nombre, empleados.oficina from empleados inner join oficinas
on empleados.oficina = oficinas.oficina where oficinas.ventas > oficinas.objetivo;
-- Con subconsulta IN
select numemp, nombre, oficina from empleados where oficina in
(select oficina from oficinas where ventas > objetivo);
-- Con subconsulta ANY
select numemp, nombre, oficina from empleados where oficina = any
(select oficina from oficinas where ventas > objetivo);
```

- 3.2.2.2 Listar los vendedores que no trabajan en oficinas dirigidas por el empleado 108. Implementa dos versiones: una con JOIN y **otra con subconsultas**.

**DIFICULTAD MEDIA**

```
-- Con Left Join
select e.numemp, e.nombre, e.oficina from empleados e left join
oficinas o on e.oficina = o.oficina where o.dir != 108; -- o.dir <> 108;
-- Subconsulta NOT IN
select numemp, nombre, oficina from empleados where (oficina not in
(select oficina from oficinas where dir = 108));
```

### 3.2.3 Subconsultas correlacionadas

- 3.2.3.1 Listar las oficinas en donde haya un vendedor cuyas ventas representen más del 55% del objetivo de su oficina.

**DIFICULTAD MEDIA**

```
-- Subconsulta inner join
select o.* from oficinas o inner join empleados e on o.oficina = e.oficina
where e.ventas > (select o2.objetivo*0.55 from oficinas o2 where e.oficina=o2.oficina);
-- Subconsulta exists
select o.* from oficinas o where exists (select * from empleados e
where e.ventas>o.objetivo*0.55 and e.oficina=o.oficina);
```

- 3.2.3.2 Listar las oficinas que tengan un objetivo mayor que la suma de las cuotas de sus vendedores.

**DIFICULTAD MEDIA**

```
select * from oficinas where objetivo >
(select sum(cuota) from empleados where empleados.oficina = oficinas.oficina);
```

- 3.2.3.3 Listar las oficinas en donde todos los vendedores tienen ventas que superan al 50% del objetivo de su oficina.

**DIFICULTAD ALTA**

```
-- Con not exists
select o.* from oficinas o where (not exists (select * from
empleados e where e.oficina = o.oficina and e.ventas >= (o.objetivo*0.5)));

-- Con ALL
select o.* from oficinas o where o.objetivo*0.5 <= all
(select e.ventas from empleados e where e.oficina = o.oficina and e.ventas >= o.objetivo*0.5);
select o.* from oficinas o where o.objetivo*0.5 <= all
(select e.ventas from empleados e where e.oficina = o.oficina and e.ventas >= o.objetivo*0.5)
and (exists (select * from empleados e2 where e2.oficina=o.oficina));
```

- 3.2.3.4 Listar los empleados (nombre y ciudad donde trabajan) cuya cantidad total de productos vendidos sea la máxima cantidad total vendida por los empleados de los de su misma ciudad.

#### **DIFICULTAD ALTA**

```
select E.nombre, O.ciudad, SUM(P.cant)
from empleados E join oficinas O on E.oficina=O.numoficina
join pedidos P on E.numemp=P.rep
GROUP BY E.numemp
HAVING SUM(P.cant) >= ALL (SELECT SUM(P2.cant)
from empleados E2 join oficinas O2 on E2.oficina=O2.numoficina
join pedidos P2 on E2.numemp=P2.rep
WHERE o2.ciudad=o.ciudad
GROUP BY E2.numemp);
```

### 3.3 BD 'bdGestionVentas'

#### 3.3.1 Subconsultas simples de comparación

- 3.3.1.1 Devuelve un listado con todos los pedidos que ha realizado Adela Salas Díaz. (Sin utilizar INNER JOIN).

#### **DIFICULTAD BAJA**

```
SELECT *
FROM pedido
WHERE id_cliente = (SELECT id
FROM cliente
WHERE nombre = 'Adela'
AND apellido1 = 'Salas'
AND apellido2 = 'Diaz');
```

- 3.3.1.2 Devuelve el número de pedidos en los que ha participado el comercial Daniel Sáez Vega. (Sin utilizar INNER JOIN)

#### **DIFICULTAD BAJA**

```
SELECT COUNT(*)
FROM pedido
WHERE id_comercial = (SELECT id
FROM comercial
WHERE nombre = 'Daniel'
AND apellido1 = 'Sáez'
AND apellido2 = 'Vega');
```

- 3.3.1.3 Devuelve los datos del cliente que realizó el pedido más caro en el año 2019. (Sin utilizar INNER JOIN)

#### **DIFICULTAD MEDIA**

```

SELECT *
FROM cliente
WHERE id = (SELECT id_cliente
            FROM pedido
            WHERE total = (SELECT MAX(total)
                          FROM pedido
                          WHERE YEAR(fecha) = 2019));

```

NOTA: Si hubiese dos clientes que durante el año 2019 han realizado un pedido con el máximo valor entonces tendríamos que modificar la consulta con = ANY para que apareciesen todos los clientes en el resultado final.

```

SELECT *
FROM cliente
WHERE id = ANY (SELECT id_cliente
                FROM pedido
                WHERE total = (SELECT MAX(total)
                              FROM pedido
                              WHERE YEAR(fecha) = 2019));

```

- 3.3.1.4 Devuelve la fecha y la cantidad del pedido de menor valor realizado por el cliente Pepe Ruiz Santana.

#### **DIFICULTAD MEDIA**

```

SELECT fecha, total
FROM cliente INNER JOIN pedido
ON cliente.id = pedido.id_cliente
WHERE cliente.nombre = 'Pepe'
AND apellido1 = 'Ruiz'
AND apellido2 = 'Santana'
AND pedido.total = (SELECT MIN(total)
                    FROM cliente INNER JOIN pedido
                    ON cliente.id = pedido.id_cliente
                    WHERE cliente.nombre = 'Pepe'
                    AND apellido1 = 'Ruiz'
                    AND apellido2 = 'Santana');

```

- 3.3.1.5 Devuelve un listado con los datos de los clientes y los pedidos, de todos los clientes que han realizado un pedido durante el año 2017 con un valor mayor o igual al valor medio de los pedidos realizados durante ese mismo año.

#### **DIFICULTAD MEDIA**

```

SELECT *
FROM cliente INNER JOIN pedido
ON cliente.id = pedido.id_cliente
WHERE YEAR(pedido.fecha) = 2017
AND pedido.total >= (SELECT AVG(total)
                    FROM pedido
                    WHERE YEAR(fecha) = 2017);

```

### 3.3.2 Subconsultas simples de pertenencia

- 3.3.2.1 Devuelve el pedido más caro que existe en la tabla pedido si hacer uso de MAX, ORDER BY ni LIMIT.

#### **DIFICULTAD BAJA**

```
SELECT *
FROM pedido
WHERE total >= ALL (SELECT total
                    FROM pedido);
```

- 3.3.2.2 Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando ANY o ALL).

**DIFICULTAD BAJA**

```
SELECT *
FROM cliente
WHERE id != ALL (SELECT id_cliente FROM pedido);
```

- 3.3.2.3 Devuelve un listado de los comerciales que no han realizado ningún pedido. (Utilizando ANY o ALL).

**DIFICULTAD BAJA**

```
SELECT *
FROM comercial
WHERE id != ALL (SELECT id_comercial FROM pedido);
```

- 3.3.2.4 Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando IN o NOT IN).

**DIFICULTAD BAJA**

```
SELECT *
FROM cliente
WHERE id NOT IN (SELECT id_cliente FROM pedido);
```

- 3.3.2.5 Devuelve un listado de los comerciales que no han realizado ningún pedido. (Utilizando IN o NOT IN).

**DIFICULTAD BAJA**

```
SELECT *
FROM comercial
WHERE id NOT IN (SELECT id_comercial FROM pedido);
```

### 3.3.3 Subconsultas con EXISTS y NOT EXISTS

- 3.3.3.1 Devuelve un listado de los clientes que no han realizado ningún pedido. (Utilizando EXISTS o NOT EXISTS).

**DIFICULTAD BAJA**

```
SELECT *
FROM cliente
WHERE NOT EXISTS (SELECT id_cliente
                  FROM pedido
                  WHERE cliente.id = pedido.id_cliente);
```

- 3.3.3.2 Devuelve un listado de los comerciales que no han realizado ningún pedido. (Utilizando EXISTS o NOT EXISTS).

**DIFICULTAD BAJA**

```
SELECT *
FROM comercial
WHERE NOT EXISTS (SELECT id_comercial
                  FROM pedido
                  WHERE comercial.id = pedido.id_comercial);
```

### 3.4 BD 'bdTiendaInformatica'

#### 3.4.1 Subconsultas variadas

- 3.4.1.1 Devuelve los nombres de los fabricantes que tienen productos asociados. (Utilizando EXISTS o NOT EXISTS).

**DIFICULTAD BAJA**

```
SELECT nombre
FROM fabricante
WHERE EXISTS (
    SELECT codigo_fabricante
    FROM producto
    WHERE producto.codigo_fabricante = fabricante.codigo);
```

- 3.4.1.2 Devuelve los nombres de los fabricantes que no tienen productos asociados. (Utilizando EXISTS o NOT EXISTS).

**DIFICULTAD BAJA**

```
SELECT nombre
FROM fabricante
WHERE NOT EXISTS (
    SELECT codigo_fabricante
    FROM producto
    WHERE producto.codigo_fabricante = fabricante.codigo);
```

- 3.4.1.3 Lista el nombre de cada fabricante con el nombre y el precio de su producto más caro.

**DIFICULTAD MEDIA**

```
/* Con un SELECT anidado y sin INNER JOIN */
SELECT producto.nombre, producto.precio, fabricante.nombre
FROM producto, fabricante
WHERE producto.codigo_fabricante = fabricante.codigo
AND producto.precio =
(
    SELECT MAX(precio)
    FROM producto
    WHERE codigo_fabricante = fabricante.codigo
);

/* Con un SELECT anidado y con INNER JOIN */
SELECT producto.nombre, producto.precio, fabricante.nombre
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo
WHERE producto.precio =
(
    SELECT MAX(precio)
    FROM producto
    WHERE codigo_fabricante = fabricante.codigo
);
```

- 3.4.1.4 Devuelve un listado de todos los productos que tienen un precio mayor o igual a la media de todos los productos de su mismo fabricante.

**DIFICULTAD MEDIA**

```

SELECT *
FROM producto AS p1
WHERE precio >= (SELECT AVG(precio)
                 FROM producto AS p2
                 WHERE p1.codigo_fabricante = p2.codigo_fabricante);

```

- 3.4.1.5 Lista el nombre del producto más caro del fabricante Lenovo.

**DIFICULTAD MEDIA**

```

SELECT producto.nombre
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
WHERE fabricante.nombre = 'Lenovo' AND producto.precio = (
    SELECT MAX(precio)
    FROM producto
    WHERE codigo_fabricante = fabricante.codigo);

```

- 3.4.1.6 Devuelve un listado con todos los nombres de los fabricantes que tienen el mismo número de productos que el fabricante Lenovo.

**DIFICULTAD ALTA**

```

SELECT fabricante.nombre, COUNT(producto.codigo)
FROM fabricante INNER JOIN producto
ON fabricante.codigo = producto.codigo_fabricante
GROUP BY fabricante.codigo
HAVING COUNT(producto.codigo) >= (
    SELECT COUNT(producto.codigo)
    FROM fabricante INNER JOIN producto
    ON fabricante.codigo = producto.codigo_fabricante
    WHERE fabricante.nombre = 'Lenovo');

```

- 3.4.1.7 *AMPLIACIÓN: Devuelve un listado con los nombres de los fabricantes y el número de productos que tiene cada uno con un precio superior o igual a 220 €. El listado debe mostrar el nombre de todos los fabricantes, es decir, si hay algún fabricante que no tiene productos con un precio superior o igual a 220€ deberá aparecer en el listado con un valor igual a 0 en el número de productos.*

**DIFICULTAD ALTA**

```

(SELECT F.nombre, COUNT(P.codigo) FROM fabricante F
                                INNER JOIN producto P
                                ON F.codigo=P.codigo_fabricante
WHERE P.precio >= 220
GROUP BY F.codigo)
UNION
(SELECT F.nombre, 0 FROM fabricante F
WHERE F.codigo NOT IN (SELECT F.codigo FROM fabricante F
                                INNER JOIN producto P
                                ON F.codigo=P.codigo_fabricante
                                WHERE P.precio >= 220))
ORDER BY 2 desc;

```

## 3.5 BD 'bdJardineria'



- 3.5.1.1 Devuelve información sobre los productos cuya diferencia entre su precio de venta y la media de precios de venta su misma gama sea la mínima (de menor varianza).

| nombre                            | PrecioVenta | gama         | Media de su gama | Varianza de gama |
|-----------------------------------|-------------|--------------|------------------|------------------|
| Sierra de Poda 400MM              | 14.00       | Herramientas | 13.000000        | 1.000000         |
| Pala                              | 14.00       | Herramientas | 13.000000        | 1.000000         |
| Rastrillo de Jardín               | 12.00       | Herramientas | 13.000000        | 1.000000         |
| Azadón                            | 12.00       | Herramientas | 13.000000        | 1.000000         |
| Ajedrea                           | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Lavándula Dentata                 | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Mejorana                          | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Melissa                           | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Mentha Sativa                     | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Petrosilium Hortense (Peregil)    | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Salvia Mix                        | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Thymus Citriodra (Tomillo lim...) | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Thymus Vulgaris                   | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Santolina Chamaecyparys           | 1.00        | Aromáticas   | 1.000000         | 0.000000         |
| Olea-Olivos                       | 25.00       | Frutales     | 23.925926        | 1.074074         |
| Calamondin Copa                   | 25.00       | Frutales     | 23.925926        | 1.074074         |
| Phyllostachys Bambusa Spec...     | 24.00       | Ornamentales | 23.311688        | 0.688312         |

### DIFICULTAD ALTA

```

Select nombre, p.PrecioVenta, p.gama, tablaAux.mediaProductos as 'Media de su gama'
      , abs(p.precioVenta - tablaAux.mediaProductos) as 'Varianza de gama'
from productos p join (SELECT p2.gama,AVG(p2.PrecioVenta) as MediaProductos
                      from productos p2 group by p2.Gama)
                      as tablaAux on tablaAux.Gama=p.Gama
where abs(p.precioVenta - tablaAux.mediaProductos) <= ALL (select abs(p3.precioVenta - tablaAux2.mediaProductos2)
                                                          from productos p3 join (select gama,AVG(PrecioVenta) as MediaProductos2
                                                          from productos p4 group by gama) as tablaAux2
                                                          on p3.Gama= tablaAux2.Gama
                                                          where p3.gama=p.gama)

```