



DeepL

Suscríbete a DeepL Pro para poder traducir archivos de mayor tamaño.
Más información disponible en www.DeepL.com/pro.



I.E.S. Rodrigo Caro

Departamento de Informática

www.informaticarodrigocaros.es

Informática
Rodrigo
Caro

Unidad 5

Consultas de acceso

ASIR



Introducción a las consultas

Como sabes, utilizamos **tablas** para **almacenar información**. En una tabla puede haber un gran número de datos. Cada **columna** de la tabla es un **campo** o propiedad de la tabla y **cada fila es una instancia** o elemento de ese tipo. Sin embargo, puede ocurrir que no desee obtener toda la información almacenada en la tabla. Quizá sólo te interesen algunas columnas o quizá quieras obtener algunas instancias con algunas propiedades. Para ello se utilizan las consultas.

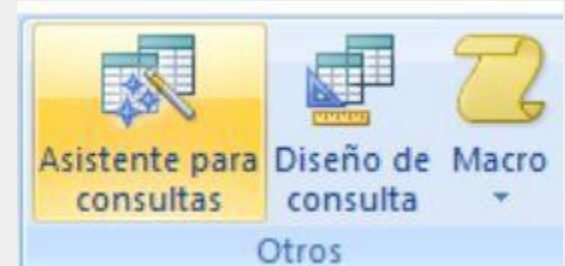
Una consulta permite **hacer preguntas** sobre la información almacenada en las tablas de una base de datos.

Las consultas son **objetos de la base de datos** que nos permiten **manipular la información almacenada** en las tablas mediante una serie de operaciones básicas fundamentales: **añadir, eliminar, consultar y modificar**.

Consulta con Access

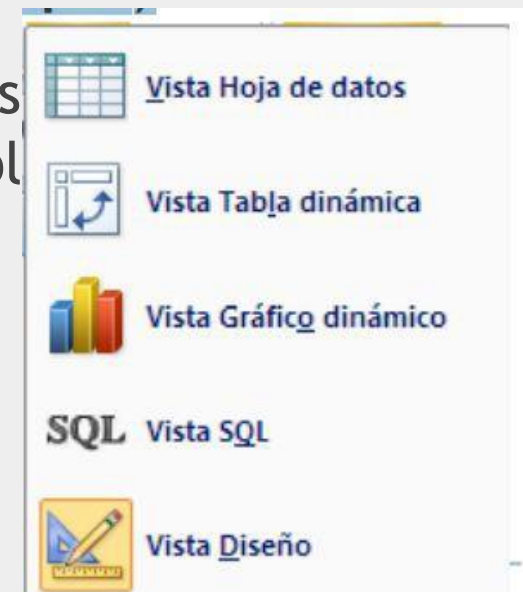
Para crear una consulta puedes

- Utilizar el asistente
- O utilice el Diseñador de consultas (mejor esta opción)



Trabajando con consultas existen estas vistas:

- Vista SQL: Sentencia SQL.
- Vista de la hoja de datos: resultado de la consulta
- Vista de diseño ->QBE Grid (Query By Example)



Tipos de consultas

- **Seleccionar:** elige un grupo de registros de una o varias tablas que cumplan determinadas condiciones.
- **Insertar o Añadir datos:** añade un grupo de registros de una tabla a otra.
- **Borrar:** borra de una tabla los registros que cumplen determinadas condiciones.
- **Actualización:** Modifica los registros de una tabla que cumplen determinadas condiciones.

Seleccionar tipos de consulta

Objetivo: Seleccionar un grupo de registros (tuplas) de una o varias tablas

que cumplan determinadas condiciones.

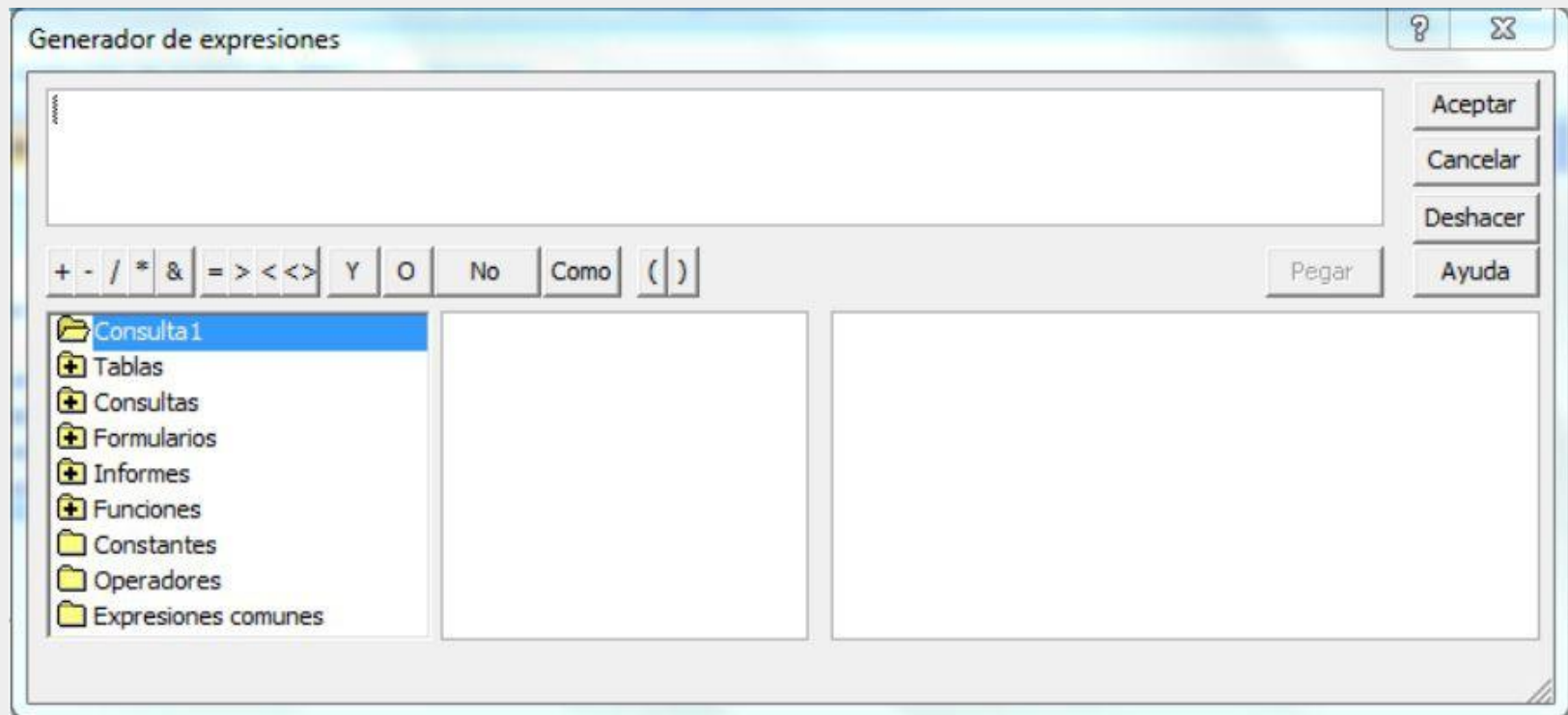
Seleccionaremos unos elementos u otros en función de determinados criterios o condiciones.

Criterios: Son condiciones que limitan la búsqueda de datos en una tabla. Los criterios se aplican a **cada uno de los campos de la tabla**.

Para especificar los criterios utilizamos **expresiones** y para crear expresiones podemos utilizar el "Generador de expresiones".

Seleccionar tipos de consulta : Expresiones

Como puedes ver, las expresiones podrían utilizar fields de muchos objetos en nuestra expresión: tablas, otras consultas, formularios, informes, etc.



Seleccionar tipos de consulta : Expresiones

Una expresión puede contener estos tipos:

Las expresiones están compuestas por:

- Valores literales
- Operadores aritméticos: +, -, *, / , ^
- Comparativo: <, <=, <>, =, >, >=, Entre ... Y ... , Como
- Lógico: Y, O, O, No(Denegado)
- Paréntesis
- Otros campos. Deben ir entre corchetes
por ejemplo [field]
- Otras expresiones

Seleccionar tipos de consulta : Expresiones

- **Texto:** entre comillas "Ejemplo".
- **Fechas:** entre comillas #01/02/2010#.
- **Números:** Directamente el valor sin más
- **Constantes definidas** como Null, true o false, "" (cadena vacía)
- **Caracteres especiales:**
 - * sustituye uno o varios caracteres
 - ? sustituye a UN carácter

Seleccionar expresiones lógicas

| var1 | var2 | NO var2 | var1 Y var2 | var1 O var2 |
|-----------|-----------|-----------|-------------|-------------|
| verdadero | verdadero | falso | verdadero | verdadero |
| verdadero | falso | verdadero | falso | verdadero |
| falso | verdadero | falso | falso | verdadero |
| falso | falso | verdadero | falso | falso |

- El operador **AND** es verdadero si ambas condiciones son verdaderas.

- El operador **OR** es verdadero si cualquiera de las condiciones es verdadera.

Ejemplos de consultas selectas

- Seleccionar para elegir las filas cuyo campo a coincida con un valor.
Por ejemplo, para esta tabla:

| Child | | Employee | | | | | |
|-------|------------|------------|--------------|-----------|-----------|------------|---------|
| | idEmployee | phoneNuml | registration | nameEmplo | surnameEm | numberChil | idDepto |
| + | 2 | 9879879877 | 01/01/2004 | Camilo | Sorroque | 0 | 1 |
| + | 3 | 6363636363 | 01/12/2020 | Pamela | Alba | 0 | 1 |
| + | 4 | 654654654 | 01/01/2022 | Dario | Martin | 0 | 2 |
| + | 5 | 954987987 | 01/02/2015 | David | Alba | 0 | 1 |
| * | (Nuevo) | | | | | 0 | 0 |

Ejemplos de consultas selectas

Seleccionar filas con "Alba" como apellidoEmpleado

We add all the fields we would like to select from one or more tables and we define criteria to match

| Campo: | tablename | fieldname | Mostrar: | Criteria: |
|----------|-----------|------------------|-------------------------------------|-----------|
| Employee | Employee | phoneNumber | <input checked="" type="checkbox"/> | |
| Employee | Employee | registrationDate | <input checked="" type="checkbox"/> | |
| Employee | Employee | nameEmployee | <input checked="" type="checkbox"/> | |
| Employee | Employee | surnameEmployee | <input checked="" type="checkbox"/> | Alba |
| Employee | Employee | numberChildren | <input type="checkbox"/> | |
| Employee | Employee | idDepto | <input checked="" type="checkbox"/> | |

| lastnameEm | idEmployee | phoneNumber | registration | nameEmplo | idDepto |
|------------|------------|-------------|--------------|-----------|---------|
| Alba | 3 | 6363636363 | 01/12/2020 | Pamela | 1 |
| Alba | 5 | 954987987 | 01/02/2015 | David | 1 |

```
SELECT Employee.surnameEmployee, Employee.idEmployee, Employee.phoneNumber, Employee.registrationDate, Employee.nameEmployee, Employee.idDepto
FROM Employee
WHERE (((Employee.surnameEmployee)="Alba"));
```

Ejemplos de consultas selectas

Seleccione las filas cuyo valor para el campo apellidoEmpleado sea diferente de "Alba":

Employee

phoneNumber

registrationDat

nameEmployee

surnameEmplo

numberChildre

idDepto

| | | | | | | |
|------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Campo: | surnameEmployee | idEmployee | phoneNumber | registrationDate | nameEmployee | idDepto |
| Tabla: | Employee | Employee | Employee | Employee | Employee | Employee |
| Orden: | | | | | | |
| Mostrar: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criterios: | <> "Alba" | | | | | |
| o: | | | | | | |

| surnameEm | idEmployee | phoneNuml | registration | nameEmplo | idDepto |
|-----------|------------|------------|--------------|-----------|---------|
| Sorroque | 2 | 9879879877 | 01/01/2004 | Camilo | 1 |
| Martin | 4 | 654654654 | 01/01/2022 | Dario | 2 |

Employee Consulta1

```
SELECT Employee.surnameEmployee, Employee.idEmployee, Employee.phoneNumber, Employee.registrationDate, Employee.nameEmployee, Employee.idDepto
FROM Employee
WHERE (((Employee.surnameEmployee)<>"Alba"));
```

Ejemplos de consultas selectas

Seleccione las filas cuya registrationDate esté comprendida entre 2020-2023

| | | | | | | |
|------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Campo: | surnameEmployee | idEmployee | phoneNumber | registrationDate | nameEmployee | idDepto |
| Tabla: | Employee | Employee | Employee | Employee | Employee | Employee |
| Orden: | | | | | | |
| Mostrar: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criterios: | | | | Entre #01/01/2020# Y #31/12/2023# | | |
| o: | | | | | | |

| surnameEm | idEmployee | phoneNuml | registration | nameEmplo | idDepto |
|-----------|------------|------------|--------------|-----------|---------|
| Alba | 3 | 6363636363 | 01/12/2020 | Pamela | 1 |
| Martin | 4 | 654654654 | 01/01/2022 | Dario | 2 |

```
SELECT Employee.surnameEmployee, Employee.idEmployee, Employee.phoneNumber, Employee.registrationDate, Employee.nameEmployee, Employee.idDepto
FROM Employee
WHERE (((Employee.registrationDate) Between #1/1/2020# And #12/31/2023#));
```


Ejemplos de consultas selectas

Seleccione las filas cuyo apellido empiece por "A":

| | | | | | | |
|------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Campo: | surnameEmployee | idEmployee | phoneNumber | registrationDate | nameEmployee | idDepto |
| Tabla: | Employee | Employee | Employee | Employee | Employee | Employee |
| Orden: | | | | | | |
| Mostrar: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criterios: | Como "A*" | | | | | |
| o: | | | | | | |

| surnameEm | idEmployee | phoneNuml | registration | nameEmplo | idDepto |
|-----------|------------|------------|--------------|-----------|---------|
| Alba | 3 | 6363636363 | 01/12/2020 | Pamela | 1 |
| Alba | 5 | 954987987 | 01/02/2015 | David | 1 |

Employee Consulta1

```
SELECT Employee.surnameEmployee, Employee.idEmployee, Employee.phoneNumber, Employee.registrationDate, Employee.nameEmployee, Employee.idDepto
FROM Employee
WHERE (((Employee.surnameEmployee) Like "A*"));
```

Ejemplos de consultas selectas

Seleccione las filas cuyo apellido empiece por una letra comprendida entre la A y la M:

| | | | | | | |
|------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Campo: | surnameEmployee | idEmployee | phoneNumber | registrationDate | nameEmployee | idDepto |
| Tabla: | Employee | Employee | Employee | Employee | Employee | Employee |
| Orden: | Ascendente | | | | | |
| Mostrar: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criterios: | >="A" Y <"N" | | | | | |
| o: | | | | | | |

| surnameEm | idEmployee | phoneNuml | registration | nameEmplo | idDepto |
|-----------|------------|------------|--------------|-----------|---------|
| Alba | 5 | 954987987 | 01/02/2015 | David | 1 |
| Alba | 3 | 6363636363 | 01/12/2020 | Pamela | 1 |
| Martin | 4 | 654654654 | 01/01/2022 | Dario | 2 |

```
SELECT Employee.surnameEmployee, Employee.idEmployee, Employee.phoneNumber, Employee.registrationDate, Employee.nameEmployee, Employee.idDepto
FROM Employee
WHERE (((Employee.surnameEmployee)>="A" And (Employee.surnameEmployee)<"N"))
ORDER BY Employee.surnameEmployee;
```

Tablas

Campos calculados

Son **campos virtuales** (no existen) y se utilizan para los cálculos con datos combinados de varios campos.

La idea es evitar guardar en las tablas este tipo de información porque, aunque no se repita, puede generarse.

Nomenclatura:

En Campo Campo: *nameCalculatedField: expression*

Campos calculados

Por lo tanto, vamos a eliminar numberChildren de nuestra tabla y, a continuación, vamos a añadir este campo a una consulta (volveremos a ello después de explicar cómo unir dos tablas).

Pero antes, vamos a empezar con un campo calculado más sencillo, por ejemplo:

el número de días que cada empleado trabaja en nuestra empresa (edad en días)

| | | | | | | |
|------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--|
| Campo: | idEmployee | phoneNumber | registrationDate | nameEmployee | idDepto | ageInDays: DiffFecha("d";[registrationDate];Fecha) |
| Tabla: | Employee | Employee | Employee | Employee | Employee | |
| Orden: | | | | | | |
| Mostrar: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criterios: | | | | | | |

Employee EmployeeFiledCalculated

```
SELECT Employee.surnameEmployee, Employee.idEmployee, Employee.phoneNumber, Employee.registrationDate, Employee.nameEmployee, Employee.idDepto, DateDiff("d",[registrationDate],Date()) AS ageInDays  
FROM Employee;
```

Consultas multitabla: Unirse

Si queremos seleccionar campos de diferentes tablas, debemos combinar datos de varias tablas o consultas.

Cuando se utilizan varias tablas en las consultas, es necesario para asegurarse de que existe **alguna relación entre los campos**, de modo que Access pueda saber cómo vincular la información.

Si las tablas de una consulta no están combinadas entre sí, ni **directa ni indirectamente**, Access no sabe cómo vincular la información, por lo que presenta todas las combinaciones de registros entre las dos tablas, lo que se denomina "producto cartesiano".

Consultas multitabla: Unirse

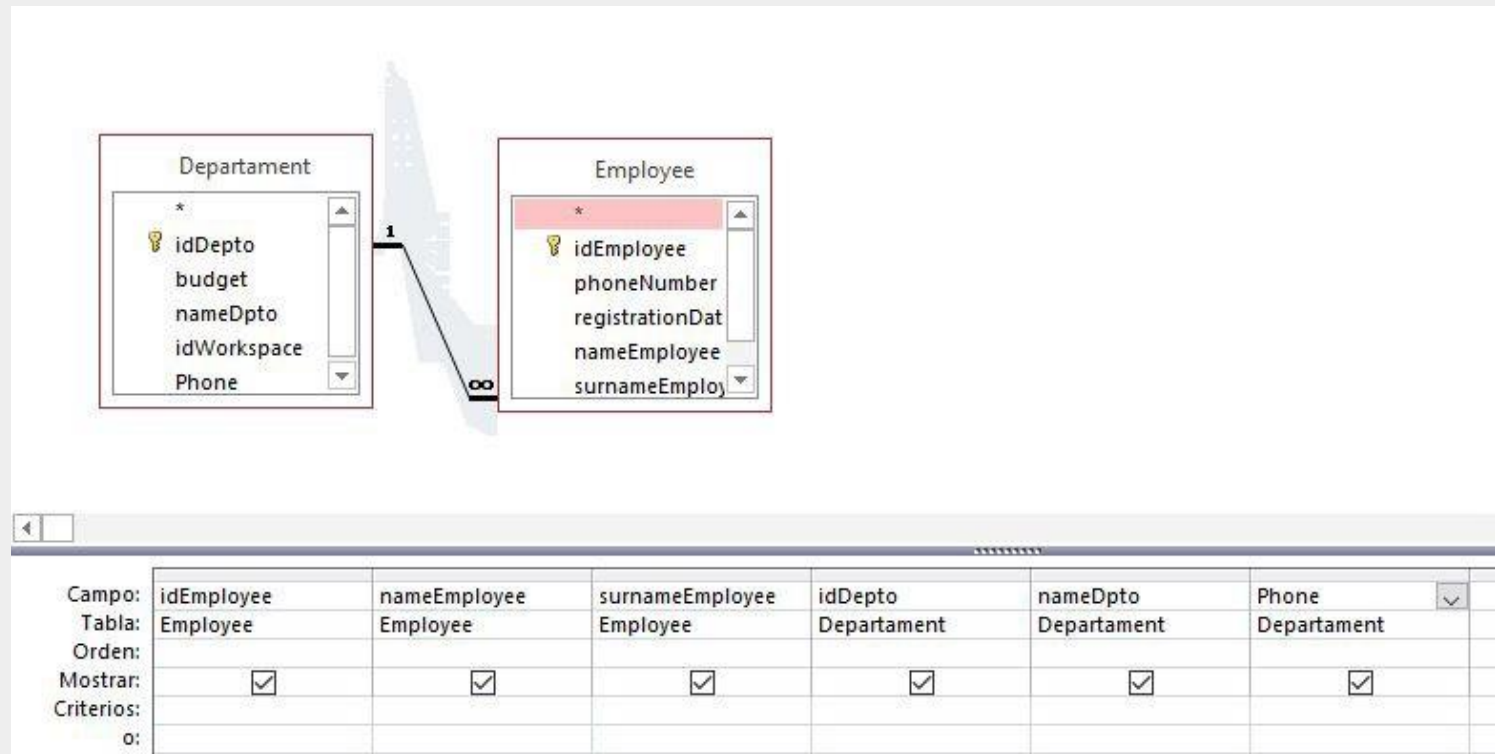
Si existen **relaciones** previamente **creadas** entre las tablas, Access **muestra automáticamente las filas de combinación** al añadir tablas relacionadas en la consulta en la vista de diseño de la consulta.

Si no se ha definido ninguna relación, puede crear filas de fusión arrastrando un campo de la tabla hasta el campo equivalente de la otra tabla.

Consultas multitabla: Unirse

Construyamos una consulta para mostrar a cada empleado la información de su departamento.

Sólo añadiendo las dos tablas, Access unirá ambas tablas



The screenshot displays the Microsoft Access interface. At the top, a relationship diagram shows two tables: 'Department' and 'Employee'. The 'Department' table has fields: idDepto (primary key), budget, nameDpto, idWorkspace, and Phone. The 'Employee' table has fields: idEmployee (primary key), phoneNumber, registrationDat, nameEmployee, and surnameEmploy. A one-to-many relationship is established between idDepto in the Department table and idEmployee in the Employee table. Below the diagram, a query design grid is shown, listing the fields from both tables to be included in the query.

| Campo: | idEmployee | nameEmployee | surnameEmployee | idDepto | nameDpto | Phone |
|------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Tabla: | Employee | Employee | Employee | Department | Department | Department |
| Orden: | | | | | | |
| Mostrar: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Criterios: | | | | | | |
| o: | | | | | | |

Consultas multitable: Unirse

| idEmployee | nameEmplo | surnameEm | idDepto | nameDpto | Phone |
|------------|-----------|-----------|---------|-----------|-------|
| 2 | Camilo | Sorroque | 1 | Computers | |
| 3 | Pamela | Alba | 1 | Computers | |
| 5 | David | Alba | 1 | Computers | |
| 4 | Dario | Martin | 2 | Analytics | |

```
SELECT Employee.idEmployee, Employee.nameEmployee, Employee.surnameEmployee,  
Departament.idDepto, Departament.nameDpto, Departament.Phone  
FROM Departament  
INNER JOIN Employee ON Departament.idDepto = Employee.idDepto;
```