Bases de Datos

- LENGUAJE DE CONSULTAS (SQL)
- CARACTERÍSTICAS DE SQL
- DEFINICIÓN DE DATOS
- CONSULTA

Lenguaje de Consultas (SQL)

- Lenguaje de Consultas (SQL)
- Características de SQL
- Definición de datos
- Consulta

Lenguaje de Consultas (SQL)

Introducción

El álgebra relacional abordada previamente, proporciona una notación concisa y formal para la representación de las consultas.

Sin embargo, los sistemas de bases de datos comerciales necesitan un lenguaje de consultas más cómodo para el usuario.

SQL usa una combinación de constructores del álgebra relacional y del cálculo relacional.

Lenguaje de Consultas (SQL)

Introducción

Aunque se haga referencia al lenguaje SQL (Structured Query Language) como “lenguaje de consultas”, puede hacer mucho más que consultar las bases de datos.

Usando SQL es posible además definir la estructura de los datos, modificar los datos de la base de datos y especificar restricciones de seguridad.

Lenguaje de Consultas (SQL)

IBM desarrolló la versión original de SQL, originalmente denominado Sequel, como parte del proyecto

System R a principios de 1970.

El lenguaje Sequel evolucionó y se convirtió en SQL (Structured Query Language, lenguaje estructurado de consultas).

Se ha establecido como el lenguaje estándar para las bases de datos relacionales.

Lenguaje de Consultas (SQL)

En 1986, ANSI (American National Standards Institute, Instituto nacional americano de normalización) e ISO (International Standards Organization, Organización internacional de normalización) publicaron una norma SQL, denominada SQL-86.

Lenguaje de Consultas (SQL)

En 1989 ANSI publicó una extensión de la norma para SQL denominada SQL-89.

La siguiente versión de la norma fue SQL-92 seguida de SQL:1999; la versión más reciente es SQL:2016.

Lenguaje de Consultas (SQL)

El lenguaje SQL tiene varios componentes:

Lenguaje de definición de datos (LDD). El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificación de los esquemas de relación.

Lenguaje de Consultas (SQL)

El lenguaje SQL tiene varios componentes:

Lenguaje interactivo de manipulación de datos (LMD). El LMD de SQL incluye un lenguaje de consultas basado tanto en el álgebra relacional como en el cálculo relacional de tuplas.

También contiene comandos para insertar, borrar y modificar tuplas.

Lenguaje de Consultas (SQL)

El lenguaje SQL tiene varios componentes:

Integridad. El LDD de SQL incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.

Las actualizaciones que violan las restricciones de integridad se rechazan.

Lenguaje de Consultas (SQL)

El lenguaje SQL tiene varios componentes:

Definición de vistas. El LDD de SQL incluye comandos para la definición de vistas.

Control de transacciones. SQL incluye comandos para especificar el comienzo y el final de las transacciones.

Lenguaje de Consultas (SQL)

El lenguaje SQL tiene varios componentes:

SQL incorporado y SQL dinámico. SQL incorporado y SQL dinámico definen cómo se pueden incorporar instrucciones de SQL en lenguajes de programación de propósito general.

Lenguaje de Consultas (SQL)

El lenguaje SQL tiene varios componentes:

Autorización. El LDD de SQL incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas.

Lenguaje de Consultas (SQL)

Definición de datos

El conjunto de relaciones de cada base de datos debe especificarse en el sistema en términos de un lenguaje de definición de datos (LDD).

El LDD de SQL no sólo permite la especificación de un conjunto de relaciones, sino también de la información relativa a esas relaciones, incluyendo:

Lenguaje de Consultas (SQL)

Definición de datos

- El esquema de cada relación.
- El dominio de valores asociado a cada atributo.
- Las restricciones de integridad.
- El conjunto de índices que se deben mantener para cada relación.
- La información de seguridad y de autorización de cada relación.
- La estructura de almacenamiento físico de cada relación en el disco.

Lenguaje de Consultas (SQL)

- Tipos básicos de dominios

Tipo	Descripción
char(n).	Una cadena de caracteres de longitud fija, con una longitud n especificada por el usuario. También se puede utilizar la palabra completa character.
varchar(n).	Una cadena de caracteres de longitud variable con una longitud máxima n especificada por el usuario. La forma completa, character varying, es equivalente.
int.	Un entero (un subconjunto finito de los enteros dependiente de la máquina). La palabra completa, integer, es equivalente.

Lenguaje de Consultas (SQL)

- Tipos básicos de dominios

Tipo	Descripción
smallint.	Un entero pequeño (un subconjunto dependiente de la máquina del tipo de dominio entero).
numeric(p; d).	Un número de coma fija, cuya precisión la especifica el usuario. El número está formado por p dígitos (más el signo), y de esos p dígitos, d pertenecen a la parte decimal. Así, numeric(3,1) permite que el número 44:5 se almacene exactamente, pero ni 444:5 ni 0:32 se pueden almacenar exactamente en un campo de este tipo.

Lenguaje de Consultas (SQL)

- Tipos básicos de dominios

Tipo	Descripción
real, double precision.	Números de coma flotante y números de coma flotante de doble precisión, con precisión dependiente de la máquina.
float(n).	Un número de coma flotante cuya precisión es, al menos, de n dígitos.

Lenguaje de Consultas (SQL)

El lenguaje SQL tiene varios componentes:

Lenguaje de definición de datos (LDD). El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificación de los esquemas de relación.

Lenguaje de Consultas (SQL)

Definición de datos

- Definición básica de esquemas en SQL

Lenguaje de Consultas (SQL)

Definición de datos

- Las relaciones se definen mediante el comando create table:

$$\textbf{create table}\ r(A_1\ D_1, A_2\ D_2, \ldots, A_n\ D_n, \langle \text{restricción-integridad}_1 \rangle, \ldots, \langle \text{restricción-integridad}_k \rangle)$$

Lenguaje de Consultas (SQL)

Definición de datos

$$\textbf{create table } r(A_1\ D_1, A_2\ D_2, \ldots, A_n\ D_n, \langle \text{restricción-integridad}_1 \rangle, \ldots, \langle \text{restricción-integridad}_k \rangle)$$

donde *r* es el nombre de la relación, cada A_i es el nombre de un atributo del esquema de la relación *r* y D_i es el tipo de dominio de los valores del dominio del atributo A_i.

Lenguaje de Consultas (SQL)

Definición de datos

- Restricción de integridad.

primary key (A_{j1}, A_{j2} ,….,A_{jm}). La especificación de **clave primaria** determina que los atributos $A_{j1,}$ A_{j2} ,…,A_{jm} forman la clave primaria de la relación.

Los atributos de la clave primaria tienen que ser *no nulos* y *únicos.*

create table $r(A_1\ D_1, A_2\ D_2, \ldots, A_n\ D_n,$
$\langle$restricción-integridad$_1\rangle$,
$\ldots,$
$\langle$restricción-integridad$_k\rangle)$

Lenguaje de Consultas (SQL)

- Definición parcial en el LDD de SQL de la base de datos bancaria que hemos estudiado.

```
create table cliente
    (nombre_cliente     char(20),
     calle_cliente      char(30),
     ciudad_cliente     char(30),
     primary key (nombre_cliente))

create table sucursal
    (nombre_sucursal    char(15),
     ciudad_sucursal    char(30),
     activos            numeric(16,2),
     primary key (nombre_sucursal))

create table cuenta
    (número_cuenta      char(10),
     nombre_sucursal    char(15),
     saldo              numeric(12,2),
     primary key (número_cuenta))

create table impositor
    (nombre_cliente     char(20),
     número_cuenta      char(10),
     primary key (nombre_cliente, número_cuenta))
```

Lenguaje de Consultas (SQL)

Las relaciones recién creadas están inicialmente vacías. Se puede utilizar el comando **insert** para añadir datos a la relación.

Por ejemplo, si se desea añadir el hecho de que hay una cuenta C-9732 en la sucursal de Navacerrada con un saldo de 1.200. Los valores se especifican en el orden en el que se encuentran en la relación.

insert into *cuenta*
values ('C-9732', 'Navacerrada', 1200)

Lenguaje de Consultas (SQL)

El comando **delete** para borrar tuplas de una relación.

La siguiente instrucción borraría todas las tuplas de la relación cuenta.

delete from *cuenta*

Lenguaje de Consultas (SQL)

Para eliminar una relación de una base de datos SQL se utiliza el comando **drop table**.

Este comando elimina de la base de datos toda la información de la relación.

$$\textbf{drop table } r$$

Lenguaje de Consultas (SQL)

El comando **alter table** se utiliza para añadir atributos a una relación existente.

Se asigna a todas las tuplas de la relación un valor nulo como valor del atributo nuevo.

$$\textbf{alter table } r \textbf{ add } A\ D$$

donde r es el nombre de una relación ya existente, A es el nombre del atributo que se desea añadir y D es el dominio del atributo añadido.

Lenguaje de Consultas (SQL)

Se pueden eliminar atributos de una relación utilizando el comando

$$\textbf{alter table } r \textbf{ drop } A$$

donde r es el nombre de una relación ya existente y A es el nombre de un atributo de la relación.

Lenguaje de Consultas (SQL)

- **Operaciones en SQL:**

La estructura básica de una expresión SQL consta de tres cláusulas: **select**, **from** y **where**.

Lenguaje de Consultas (SQL)

- **Operaciones en SQL**

La cláusula **select** corresponde a la operación proyección del álgebra relacional.

Se usa para obtener una relación de los atributos deseados en el resultado de una consulta.

Lenguaje de Consultas (SQL)

- **Operaciones en SQL**

La cláusula **from** corresponde a la operación producto cartesiano del álgebra relacional.

Genera una lista de las relaciones que deben ser analizadas en la evaluación de la expresión.

Lenguaje de Consultas (SQL)

- **Operaciones en SQL**

La cláusula **where** se corresponde con el predicado selección del álgebra relacional.

Es un predicado que engloba a los atributos de las relaciones que aparecen en la cláusula **from**.

Lenguaje de Consultas (SQL)

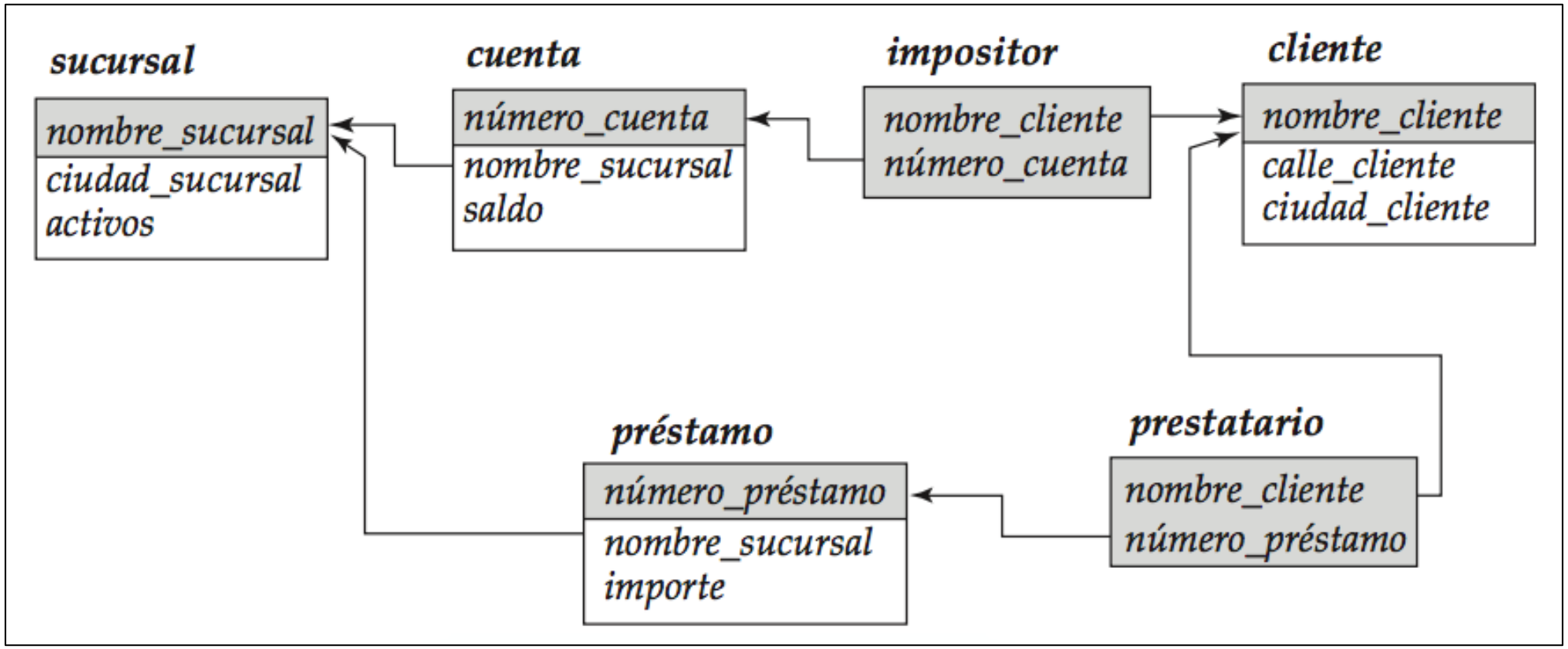
Operaciones en SQL

select $A_1,\ A_2, \ldots, A_n$
from $r_1,\ r_2, \ldots, r_m$
where P

Equivale en álgebra relacional:

$$\Pi_{A_1,\ A_2,\ldots,\ A_n}(\sigma_P(r_1 \times r_2 \times \cdots \times r_m))$$

Base de datos ejemplo. Banco



Lenguaje de Consultas (SQL)

Operaciones en SQL

Por ejemplo

select *nombre_sucursal*
from *préstamo*

select distinct *nombre_sucursal*
from *préstamo*

Lenguaje de Consultas (SQL)

Operaciones en SQL

Incluyendo la cláusula Where

Obtener todos los números de préstamo de los préstamos concedidos en la sucursal de Navacerrada

```
select número_préstamo
from préstamo
where nombre_sucursal = 'Navacerrada'
```

Lenguaje de Consultas (SQL)

- **Operaciones en SQL**

SQL usa las conectivas lógicas **and**, **or** y **not** (en lugar de los símbolos matemáticos ∧, ∨ y ¬) en la cláusula **where**.

Las expresiones pueden contener los operadores de comparación <, <=, >, >=, = y <>

Lenguaje de Consultas (SQL)

Operaciones en SQL

Incluyendo la cláusula Where

Obtener todos los números de préstamo de los préstamos concedidos en la sucursal de Navacerrada con importe superior a 1.200

```
select número_préstamo
from préstamo
where nombre_sucursal = 'Navacerrada' and importe > 1200
```

Lenguaje de Consultas (SQL)

- **Operaciones en SQL**

SQL incluye también un operador de comparación **between** para simplificar las cláusulas **where**

Lenguaje de Consultas (SQL)

Operaciones en SQL

El uso de From

Es el producto cartesiano de las relaciones que aparecen en la cláusula.

Lenguaje de Consultas (SQL)

- **Operaciones en SQL**

Determinar el nombre de todos los clientes que tienen concedido un préstamo en la sucursal de Navacerrada.

Lenguaje de Consultas (SQL)

$$\Pi_{nombre\_cliente}\left(\sigma_{prestatario.número\_préstamo = préstamo.número\_préstamo}\left(\sigma_{nombre\_sucursal = \text{“Navacerrada”}}\left(prestatario \times préstamo\right)\right)\right)$$

Prestatario

nombre_cliente	*número_préstamo*
Fernández	P-16
Gómez	P-11
Gómez	P-23
López	P-15
Pérez	P-93
Santos	P-17
Sotoca	P-14
Valdivieso	P-17

nombre_cliente
Fernández
López

Prestamo

número_préstamo	*nombre_sucursal*	*importe*
P-11	Collado Mediano	900
P-14	Centro	1.500
P-15	Navacerrada	1.500
P-16	Navacerrada	1.300
P-17	Centro	1.000
P-23	Moralzarzal	2.000
P-93	Becerril	500

Lenguaje de Consultas (SQL)

Operaciones en SQL

En SQL:

Select nombre_cliente

From prestatario,prestamo

Where nombre_sucursal=“Navacerrada” and prestatario.numero_prestamo = prestamo.numero_prestamo

Lenguaje de Consultas (SQL)

Operaciones en SQL

Determinar el nombre, el número de préstamo y los importes de todos los préstamos de la sucursal de Navacerrada

```
select nombre_cliente, prestatario.número_préstamo, importe
from prestatario, préstamo
where prestatario.número_préstamo = préstamo.número_préstamo and
      nombre_sucursal = 'Navacerrada'
```

Lenguaje de Consultas (SQL)

Operaciones en SQL

Para ordenarlos alfabéticamente

```
select distinct nombre_cliente
from prestatario, préstamo
where prestatario.número_préstamo = préstamo.número_préstamo and
        nombre_sucursal = 'Navacerrada'
order by nombre_cliente
```

Lenguaje de Consultas (SQL)

❖ Operaciones en SQL

Se desea ordenar la relación *prestamo* en forma descendente de *importe*. Si varios préstamos tienen el mismo importe, se ordenan de manera ascendente según sus números de préstamo:

```
select *
from préstamo
order by importe desc, número_préstamo asc
```

Lenguaje de Consultas (SQL)

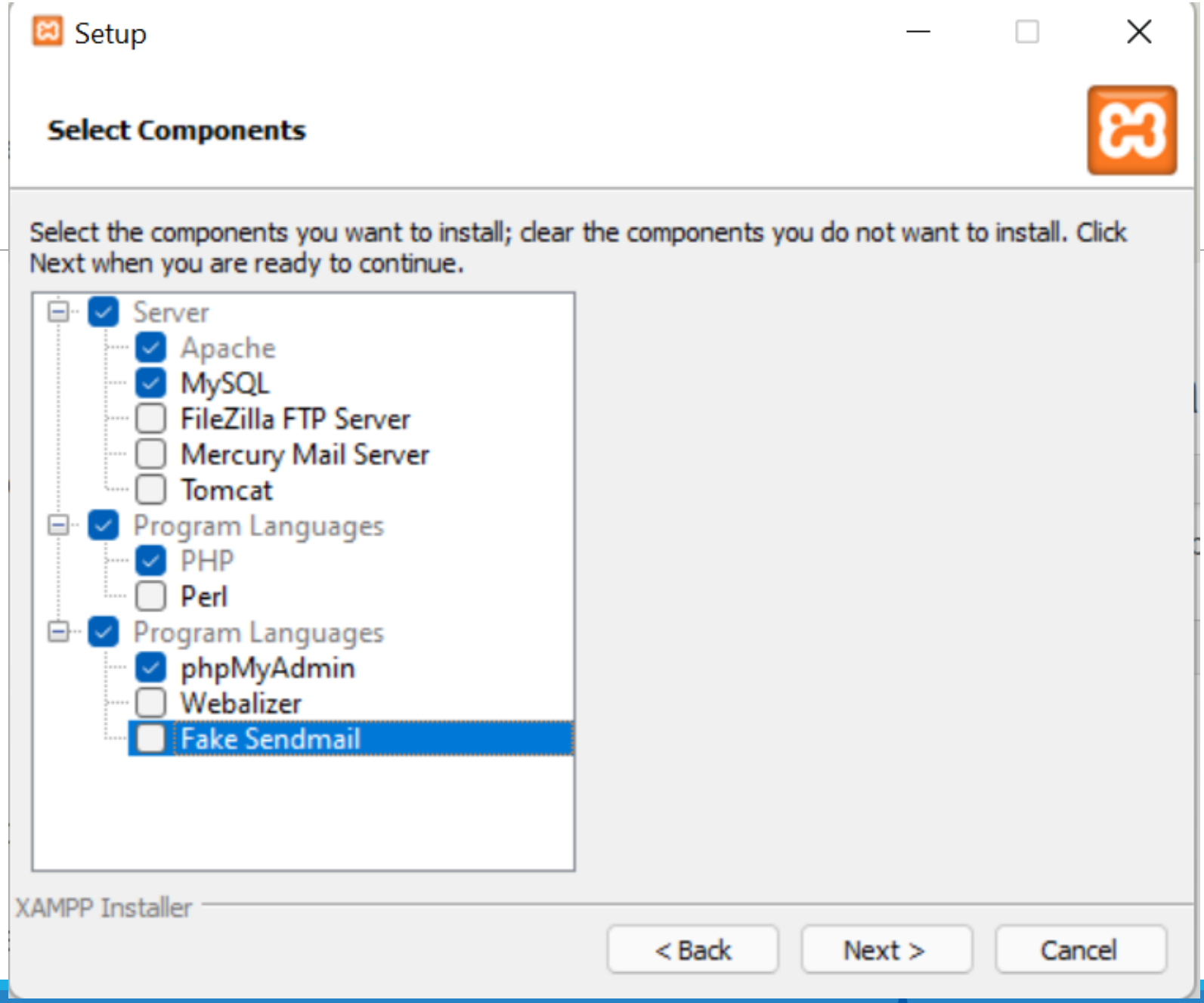
Operaciones en SQL

Unión

Todos los clientes del banco que tienen un préstamo, una cuenta o las dos cosas en el banco

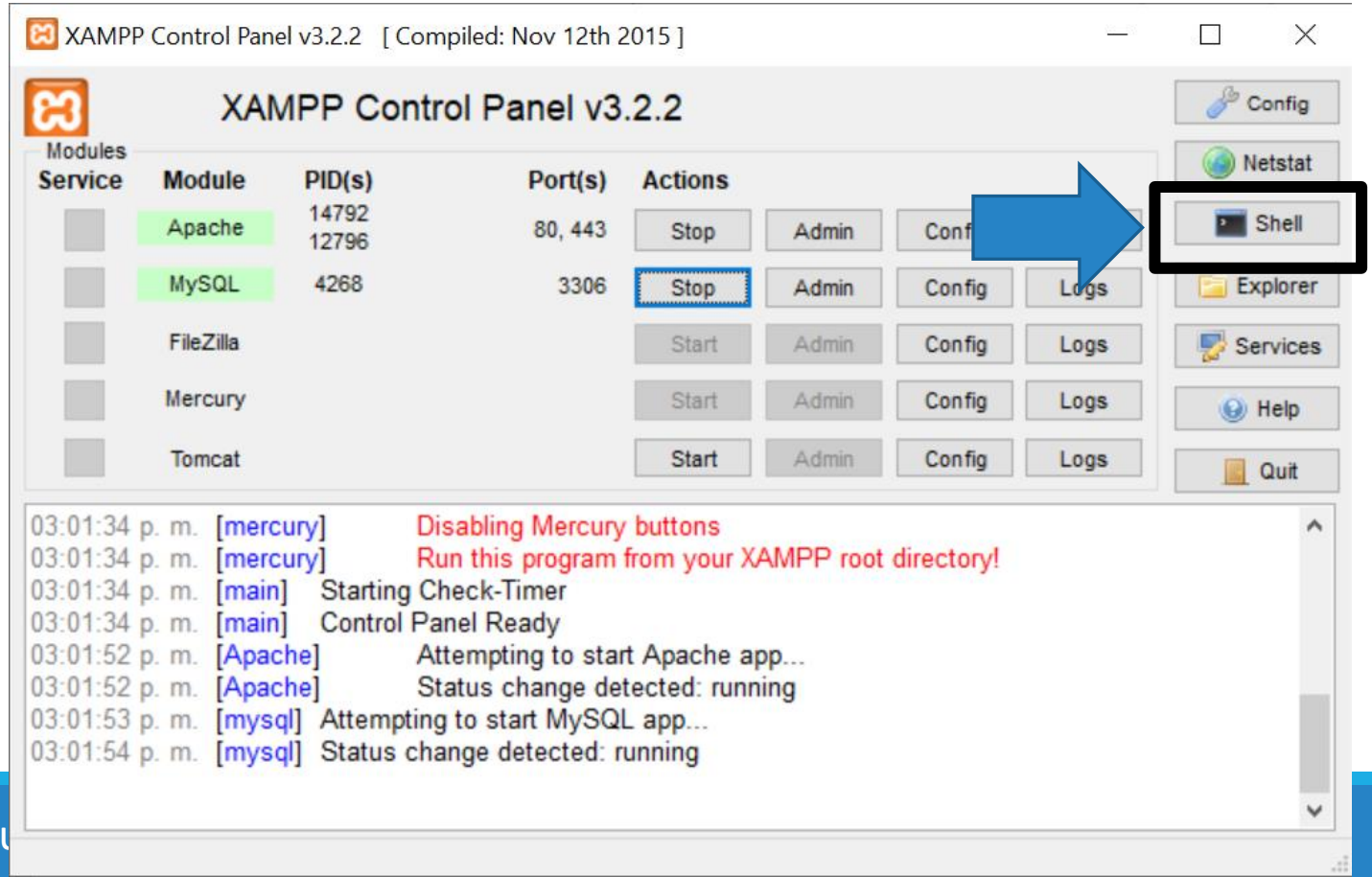
```
(select nombre_cliente
 from impositor)
union
(select nombre_cliente
 from prestatario)
```

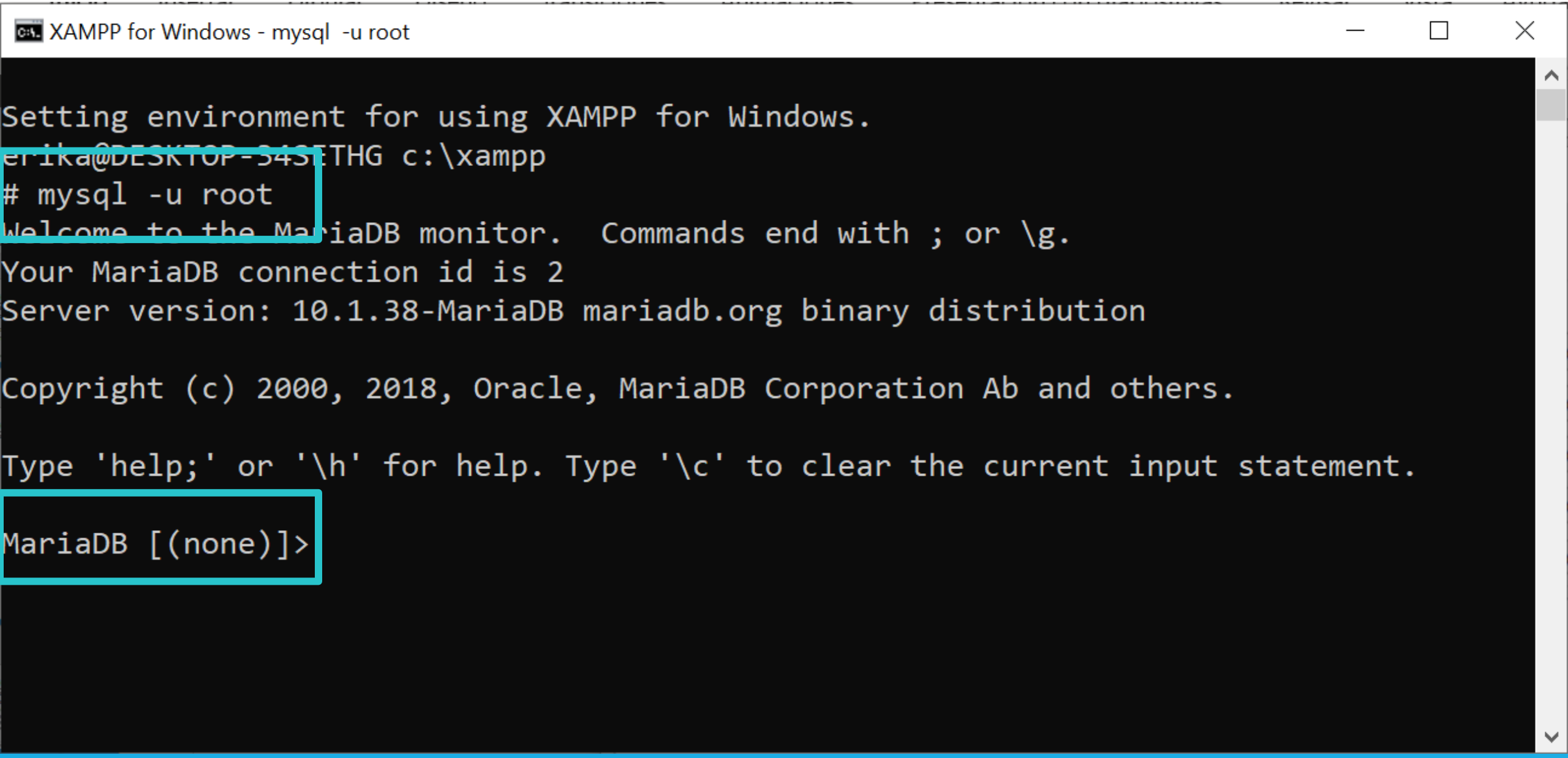
La operación **union** (unión) elimina los valores duplicados automáticamente.

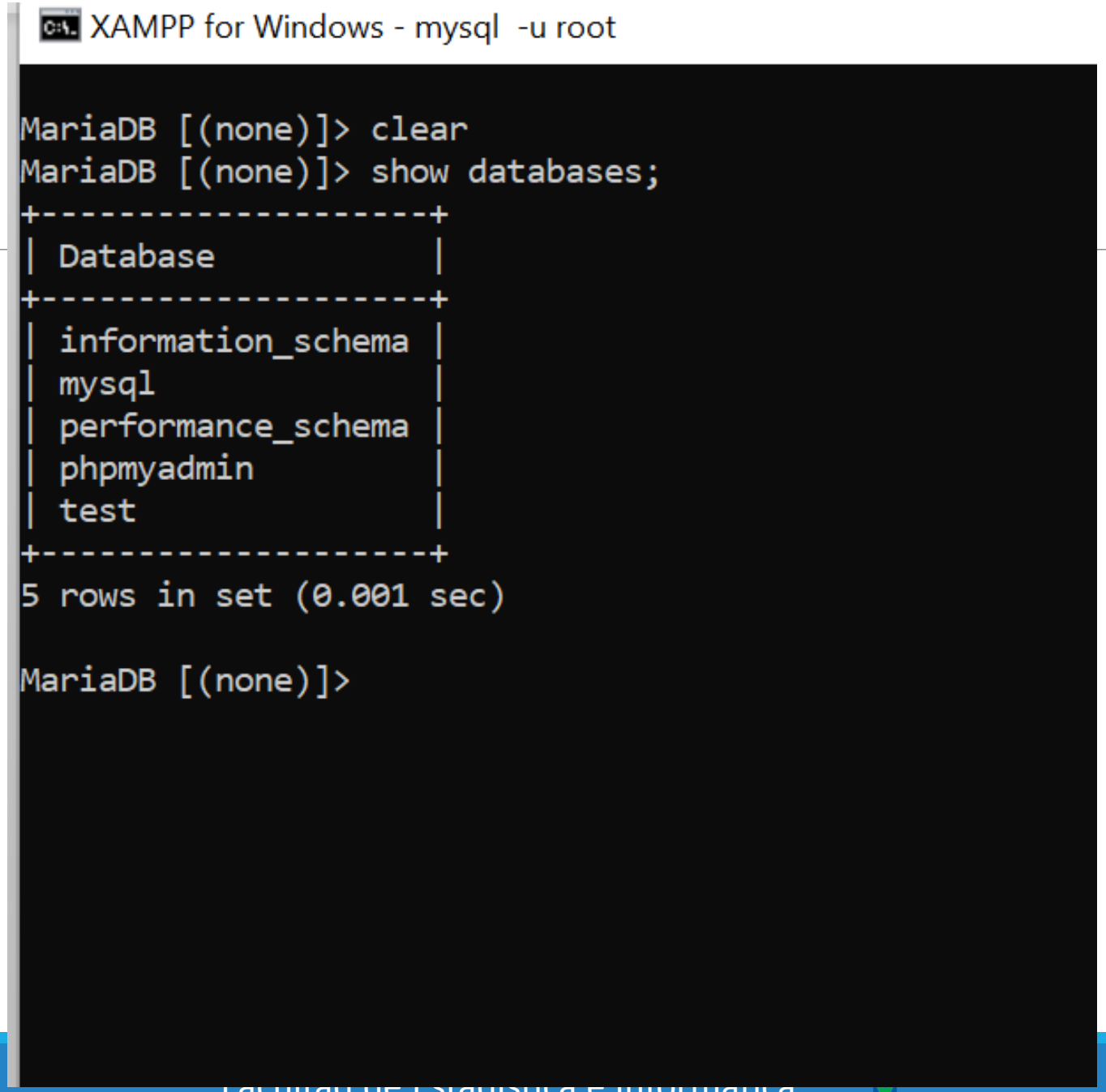

Setup
Select Components
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Server
Apache
MySQL
FileZilla FTP Server
Mercury Mail Server
Tomcat
Program Languages
PHP
Perl
Program Languages
phpMyAdmin
Webalizer
Fake Sendmail
XAMPP Installer
< Back
Next >
Cancel

Lenguaje de Consultas (SQL)

MariaDB desde línea de comandos




XAMPP for Windows - mysql -u root
Setting environment for using XAMPP for Windows.
THG c:\xampp
mysql -u root
iaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.38-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>

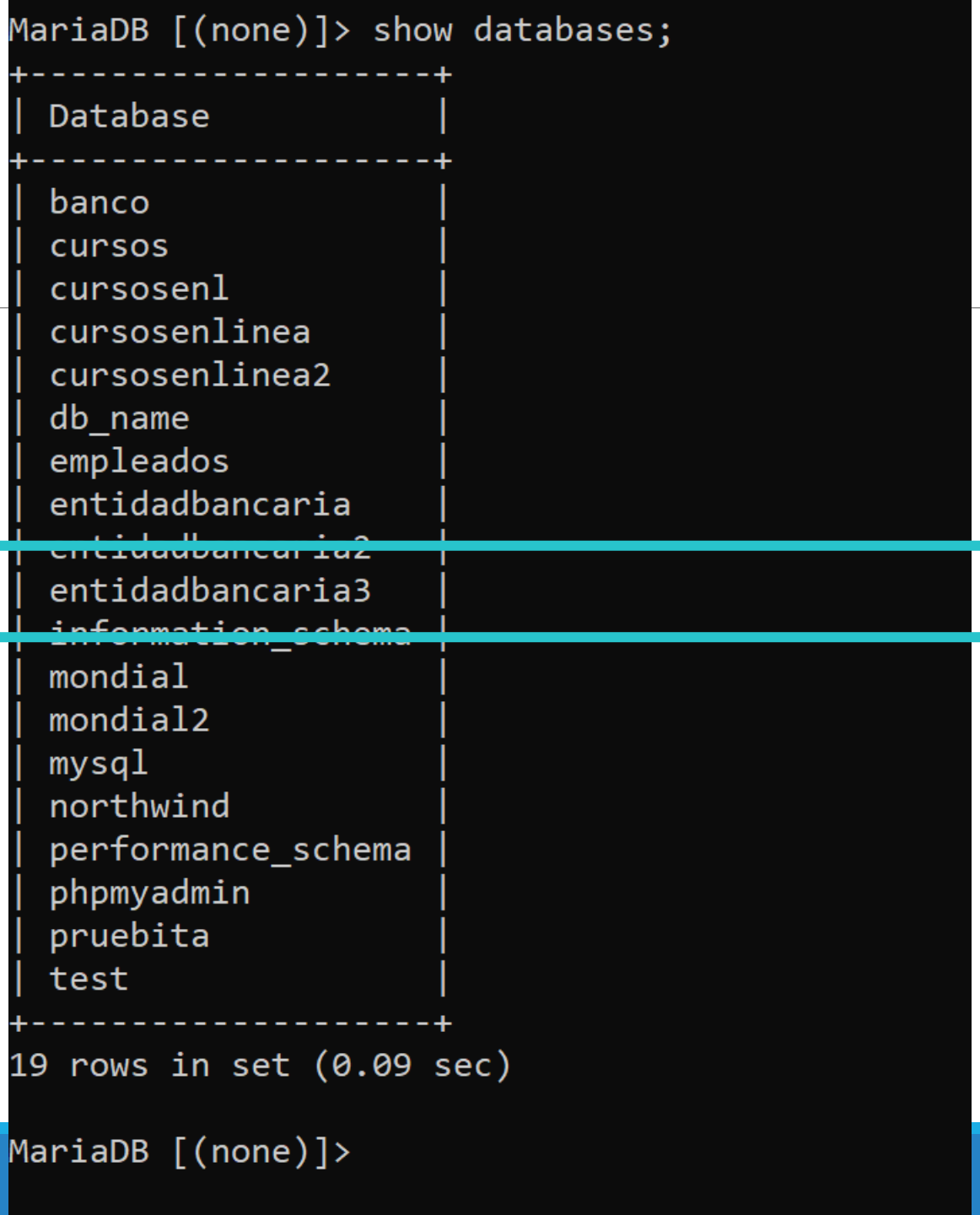

XAMPP for Windows - mysql -u root
MariaDB [(none)]> clear
MariaDB [(none)]> show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+--------------------+
5 rows in set (0.001 sec)
MariaDB [(none)]>

Lenguaje de Consultas (SQL)

Creando una base de datos:

```
CREATE DATABASE IF NOT EXISTS entidadbancaria DEFAULT CHARACTER SET
utf8 COLLATE utf8_unicode_ci;

USE entidadbancaria;
```


MariaDB [(none)]> show databases;
+--------------------+
| Database |
+--------------------+
| banco |
| cursos |
| cursosenl |
| cursosenlinea |
| cursosenlinea2 |
| db_name |
| empleados |
| entidadbancaria |
| entidadbancaria3 |
| mondial |
| mondial2 |
| mysql |
| northwind |
| performance_schema |
| phpmyadmin |
| pruebita |
| test |
+--------------------+
19 rows in set (0.09 sec)
MariaDB [(none)]>

Lenguaje de Consultas (SQL)

Agregando las tablas cliente, cuenta e impositor:

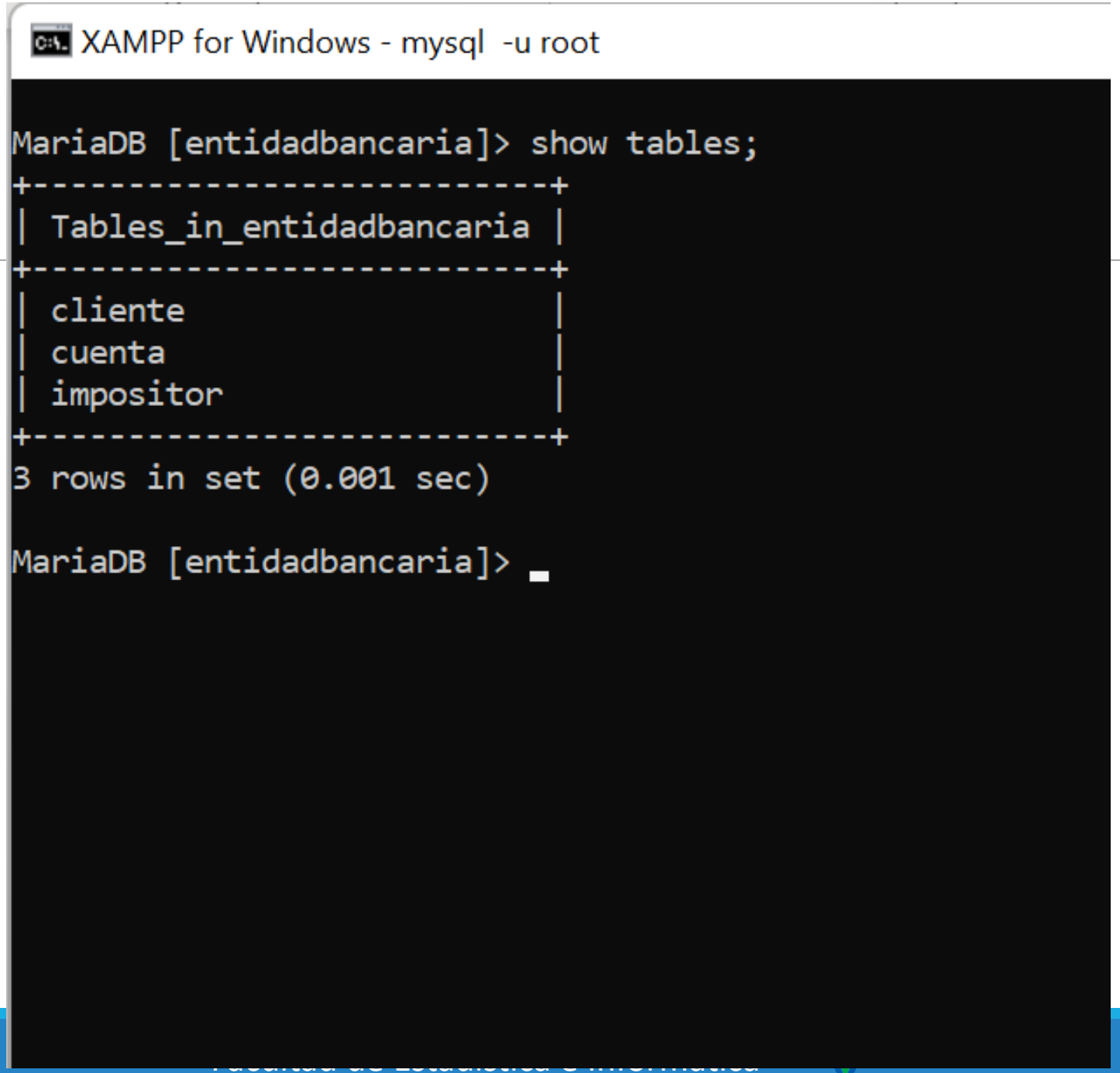
```
CREATE TABLE cliente (
  nombre_cliente varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  calle_cliente varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  ciudad_cliente varchar(255) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Lenguaje de Consultas (SQL)

```
CREATE TABLE cuenta (
  numero_cuenta varchar(10) COLLATE utf8_unicode_ci NOT NULL,
  nombre_sucursal varchar(100) COLLATE utf8_unicode_ci NOT NULL,
  saldo int(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Lenguaje de Consultas (SQL)

```
CREATE TABLE impositor (
  nombre_cliente varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  numero_cuenta varchar(10) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```


XAMPP for Windows - mysql -u root
MariaDB [entidadbancaria]> show tables;
+---------------------------+
| Tables_in_entidadbancaria |
+---------------------------+
| cliente |
| cuenta |
| impositor |
+---------------------------+
3 rows in set (0.001 sec)
MariaDB [entidadbancaria]>

Lenguaje de Consultas (SQL)

Agregando las claves primarias y foráneas:

```
ALTER TABLE cliente
   ADD PRIMARY KEY (nombre_cliente);
ALTER TABLE cuenta
   ADD PRIMARY KEY (numero_cuenta);
```

Lenguaje de Consultas (SQL)

Agregando las claves primarias y foráneas:

```
ALTER TABLE impositor

  ADD CONSTRAINT impositor_ibfk_1 FOREIGN KEY (numero_cuenta)
REFERENCES cuenta (numero_cuenta),

  ADD CONSTRAINT impositor_ibfk_2 FOREIGN KEY (nombre_cliente)
REFERENCES cliente (nombre_cliente);
```

XAMPP for Windows - mysql -u root

```
at line 1
MariaDB [entidadbancaria]> show columns from cliente;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| nombre_cliente | varchar(100) | NO   | PRI | NULL    |       |
| calle_cliente  | varchar(100) | YES  |     | NULL    |       |
| ciudad_cliente | varchar(50)  | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
3 rows in set (0.07 sec)

MariaDB [entidadbancaria]>
```

XAMPP for Windows - mysql -u root

```
MariaDB [entidadbancaria]> help;

General information about MariaDB can be found at
http://mariadb.org

List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
?         (\?) Synonym for `help'.
clear     (\c) Clear the current input statement.
connect   (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
ego       (\G) Send command to mysql server, display result vertically.
exit      (\q) Exit mysql. Same as quit.
go        (\g) Send command to mysql server.
help      (\h) Display this help.
notee     (\t) Don't write into outfile.
print     (\p) Print current command.
prompt    (\R) Change your mysql prompt.
quit      (\q) Quit mysql.
rehash    (\#) Rebuild completion hash.
source    (\.) Execute an SQL script file. Takes a file name as an argument.
status    (\s) Get status information from the server.
tee       (\T) Set outfile [to_outfile]. Append everything into given outfile.
use       (\u) Use another database. Takes database name as argument.
charset   (\C) Switch to another charset. Might be needed for processing binlog with
 multi-byte charsets.
warnings  (\W) Show warnings after every statement.
nowarning (\w) Don't show warnings after every statement.

For server side help, type 'help contents'

MariaDB [entidadbancaria]>
```

Lenguaje de Consultas (SQL)

Exportar una base de datos

1. El comando se ejecuta desde el Shell de XAMPP por lo que se debe salir de MariaDB \q.
2. Comando mysqldump

```
erika@DESKTOP-34SETHG c:\xampp
# mysqldump -u root --databases entidadbancaria > entidadbancaria.sql
```

Lenguaje de Consultas (SQL)

Importar una base de datos

1. Desde MariaDB. Comando source.

```
MariaDB [entidadbancaria]> source C:/Users/erika/Documents/Ago20-Ene21/BD/Actividades/empleados.sql;
```

Lenguaje de Consultas (SQL)

Operaciones en SQL

Ejercicios de repaso

Base de datos EMPLEADOS

empleado (*nombre_empleado*, *calle*, *ciudad*)
trabaja (*nombre_empleado*, *nombre_empresa*, *sueldo*)
empresa (*nombre_empresa*, *ciudad*)
jefe (*nombre_empleado*, *nombre_jefe*)

Lenguaje de Consultas (SQL)

Base de datos EMPLEADOS.

Elabora las siguientes consultas en SQL, empleando las operaciones de proyección, selección y producto cartesiano.

1. Determinar el nombre y ciudad de residencia de todos los empleados que trabajan en el Banco BANAMEX
2. Determinar el nombre, domicilio y ciudad de residencia de todos los empleados que ganan más de 10.000.
3. Determinar el nombre, domicilio y ciudad de residencia de todos los empleados que trabajan en el Banco BANAMEX y ganan más de 10.000.
4. Determinar el nombre de los jefes con empleados en la ciudad de Xalapa.
5. Determinar todas las empresas ordenadas alfabéticamente por nombre.

Lenguaje de Consultas (SQL)

Base de datos EMPLEADOS

Determinar el nombre y ciudad de residencia de todos los empleados que trabajan en el Banco BANAMEX.

Select empleado.nombre_empleado, ciudad

From empleado, trabaja

Where empleado.nombre_empleado=trabaja.nombre_empleado and nombre_empresa="Banamex"

Determinar el nombre, domicilio y ciudad de residencia de todos los empleados que ganan más de 10.000.

Select empleado.nombre_empleado, calle, ciudad

From empleado, trabaja

Where sueldo>10000 and empleado.nombre_empleado=trabaja.nombre_empleado

Lenguaje de Consultas (SQL)

Base de datos EMPLEADOS

Determinar el nombre, domicilio y ciudad de residencia de todos los empleados que trabajan en el Banco BANAMEX y ganan más de 10.000.

Select empleado.nombre_empleado, calle, ciudad

From empleado, trabaja

Where empleado.nombre_empleado=trabaja.nombre_empleado and nombre_empresa="Banamex" and sueldo>10000

Lenguaje de Consultas (SQL)

Base de datos EMPLEADOS

Determinar el nombre de los jefes con empleados en la ciudad de Xalapa.

Select Distinct nombre_jefe

From empleado, jefe

Where empleado.nombre_empleado=jefe.nombre_empleado and ciudad="Xalapa"

Determinar todas las empresas ordenadas alfabéticamente por nombre.

Select nombre_empresa

From empresa

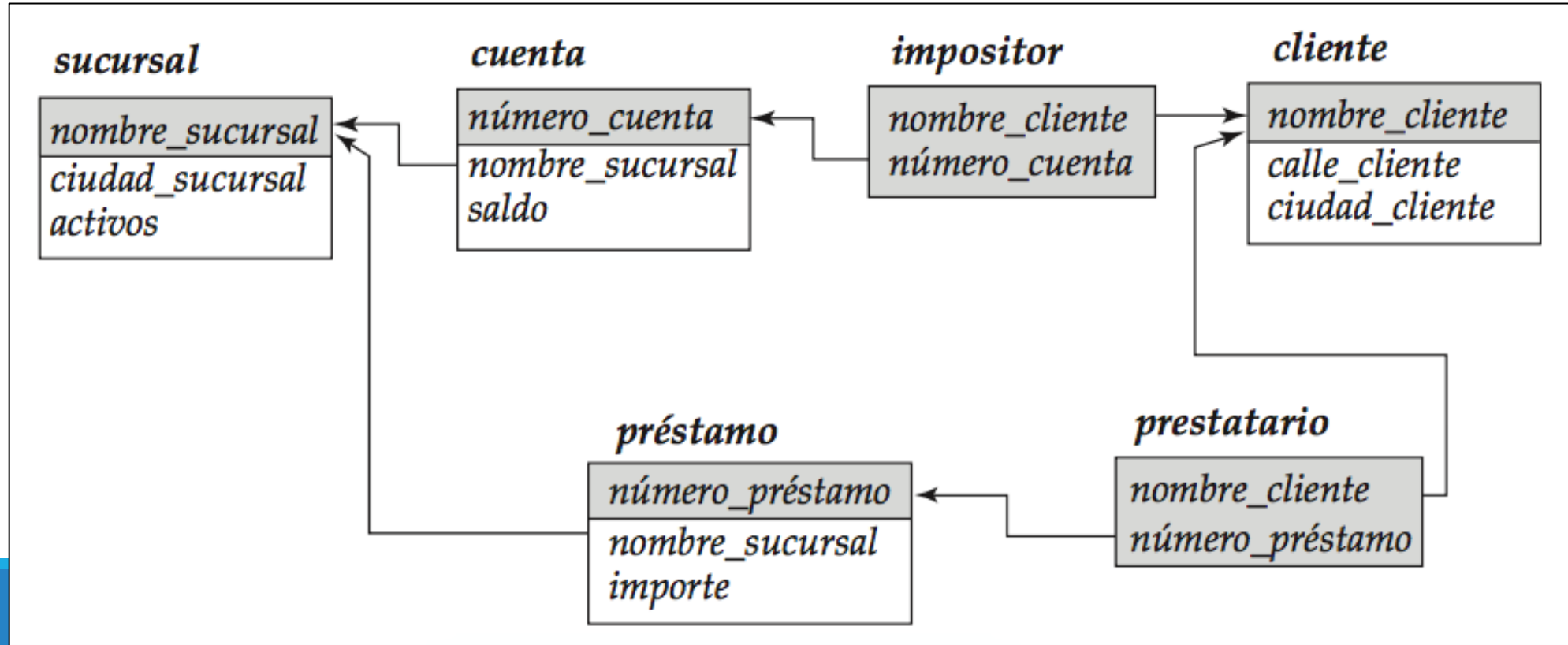
Order by nombre_empresa

Lenguaje de Consultas (SQL)

- **Operaciones en SQL**

Ejercicios de repaso

Base de datos BANCO



Lenguaje de Consultas (SQL)

Base de datos BANCO.

Elabora las siguientes consultas en SQL, empleando las operaciones de proyección, selección y producto cartesiano.

1. Mostrar el nombre de las sucursales con activos mayores a 1,000,000 y los números de cuenta que pertenecen a dichas sucursales.
2. Mostrar el nombre de los clientes que tienen préstamo en la sucursal Centro.
3. Mostrar el nombre de los clientes que tienen cuenta en la sucursal Galapagar.
4. Mostrar el nombre de los clientes, el nombre de la sucursal y el saldo de los clientes que tengan cuentas con un saldo mayor a 700.
5. Mostrar el nombre de los clientes y el importe de las personas que tienen préstamos mayores a 1000.

Lenguaje de Consultas (SQL)

Base de datos BANCO.

Elabora las siguientes consultas en SQL, empleando las operaciones de proyección, selección y producto cartesiano.

6. Mostrar el nombre de las sucursales, la ciudad sucursal y los números de préstamo otorgados en cada sucursal.

7. Mostrar el nombre de todos los clientes ordenados alfabéticamente.

8. Mostrar el nombre de las sucursales con saldos de cuenta mayores o iguales a 400.

9. Mostrar el nombre del cliente con el préstamo número P-93 y el nombre de la sucursal donde se otorgó dicho préstamo.

10. Mostrar el nombre del cliente con el número de cuenta C-222 y el nombre de la sucursal donde se tiene dicha cuenta.

Lenguaje de Consultas (SQL)

Base de datos BANCO.

Elabora las siguientes consultas en SQL, empleando las operaciones de proyección, selección y producto cartesiano.

1. Mostrar el nombre de las sucursales con activos mayores a 1,000,000 y los números de cuenta que pertenecen a dichas sucursales.

```
SELECT numero_cuenta, sucursal.nombre_sucursal FROM su
cursal,cuenta WHERE activos>3000000 and sucursal.nombr
e_sucursal=cuenta.nombre_sucursal
```

2. Mostrar el nombre de los clientes que tienen préstamo en la sucursal Centro.

```
SELECT nombre_cliente FROM prestamo, prestatario WHERE
 prestamo.numero_prestamo=prestatario.numero_prestamo
and nombre_sucursal="Centro"
```

Lenguaje de Consultas (SQL)

Base de datos BANCO.

3. Mostrar el nombre de los clientes que tienen cuenta en la sucursal Galapagar.

```
SELECT nombre_cliente FROM cuenta, impositor WHERE cuenta.numero_cuenta
=impositor.numero_cuenta and nombre_sucursal="Galapagar"
```

4. Mostrar el nombre de los clientes, el nombre de la sucursal y el saldo de los clientes que tengan cuentas con un saldo mayor a 700.

```
SELECT nombre_cliente, nombre_sucursal, saldo FROM cuenta, impositor W
HERE cuenta.numero_cuenta=impositor.numero_cuenta AND saldo>600
```

5. Mostrar el nombre de los clientes y el importe de las personas que tienen préstamos mayores a 1000.

```
SELECT nombre_cliente, importe FROM prestatario, prestamo WHERE presta
tario.numero_prestamo=prestamo.numero_prestamo AND importe>1000
```

Lenguaje de Consultas (SQL)

Base de datos BANCO.

6. Mostrar el nombre de las sucursales, la ciudad sucursal y los números de préstamo otorgados en cada sucursal.

```
SELECT sucursal.nombre_sucursal, ciudad_sucursal, numero_prestamo FROM sucursal,prestamo WHERE sucursal.nombre_sucursal=prestamo.nombre_sucursal
```

7. Mostrar el nombre de todos los clientes ordenados alfabéticamente.

```
SELECT nombre_cliente FROM cliente ORDER BY nombre_cliente
```

Lenguaje de Consultas (SQL)

Base de datos BANCO.

8. Mostrar el nombre de las sucursales con saldos de cuenta mayores o iguales a 400.

```
SELECT DISTINCT nombre_sucursal FROM cuenta WHERE saldo>=400
```

9. Mostrar el nombre del cliente con el préstamo número P-93 y el nombre de la sucursal donde se otorgó dicho préstamo.

```
SELECT nombre_cliente,nombre_sucursal FROM prestatario,prestamo WHERE prestatario.numero_prestamo=prestamo.numero_prestamo AND prestamo.numero_prestamo="P-93"
```

Lenguaje de Consultas (SQL)

Base de datos BANCO.

10. Mostrar el nombre del cliente con el número de cuenta C-222 y el nombre de la sucursal donde se tiene dicha cuenta.

```
SELECT nombre_cliente,nombre_sucursal FROM impositor,cuenta WHERE impositor.numero_cuenta=cuenta.numero_cuenta AND cuenta.numero_cuenta="C-222"
```

Lenguaje de Consultas (SQL)

Gracias por su atención