



## **Curso: SQL Básico**

Casos prácticos



1. introducción
2. casos prácticos conectividad
3. casos prácticos con JDBC
4. casos prácticos con Hibernate
5. casos prácticos con JPA



# 1 Introducción

- 2. casos prácticos conectividad
- 3. casos prácticos con JDBC
- 4. casos prácticos con Hibernate
- 5. casos prácticos con JPA

## introducción

Nota: Si algún software se ha instalado en un curso anterior, reutilizar la instalación.

Instrucciones generales:

- Instalación del entorno de Base de datos:
  - Base de datos MySQL 5.6.11, Community Edition.
    - Descomprimir mysql-5.6.11-win32.zip desde 0.EnvPreparation/software-persistence a C:\BpE. Se genera directorio mysql-5.6.11-win32.
    - Abrir ventana de comandos (cmd) en C:\BpE\mysql-5.6.11-win32\bin.
    - Ejecutar **mysqld.exe**. Con esto se inicia servicio MySQL.
  - Manejador de base de datos MySQL.
    - Descomprimir mysql-workbench-community-6.0.6-win32-noinstall.zip desde 0.EnvPreparation/software-persistence a C:\BpE. Se genera directorio MySQL Workbench 6.0.6 CE.

Nota: La base de datos viene configurada con usuario "root" y password "jeveris". Si se quisiera cambiar la password, en C:\BpE\mysql-5.6.11-win32\bin, y se ejecuta el comando:

**mysqladmin -u root password newPassword**

## introducción

Instrucciones generales:

- El código SQL con la solución de los casos prácticos se encuentran en la carpeta "**CP-Soluciones**".



# 2 **manejo básico de estructura**

- 3. operaciones básicas sobre datos
- 4. bases de datos relacionales
- 5. manejo de datos relacionales
- 6. otras operaciones sobre datos

## manejo básico de estructura

### Caso práctico 2-1: Creación de base de datos y tablas

Resumen del ejercicio:

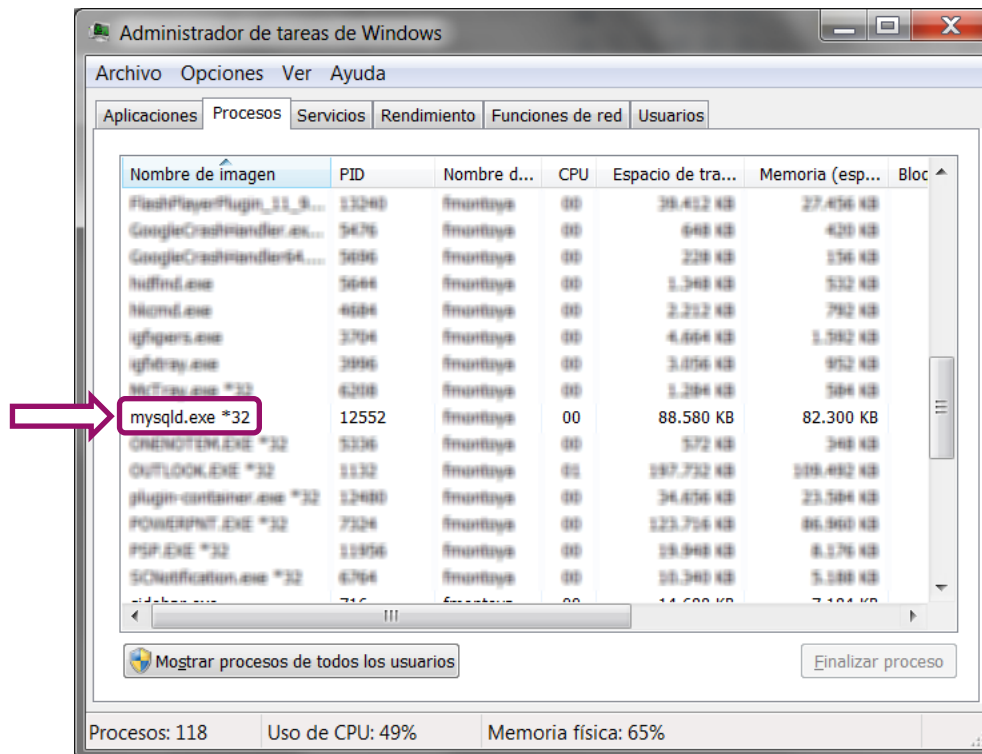
- Utilizar herramienta de manejo de base de datos MySQL.
- Crear base de datos.
- Crear una tabla de acuerdo a una definición de estructura.
- Realizar modificaciones sobre la estructura.



## manejo básico de estructura

### Caso práctico 2-1: Creación de base de datos y tablas

Se pide primero iniciar el DBMS MySQL, utilizando el comando `mysqld` (directorio bin de C:\BpE\mysql-5.6.11-win32). Se puede comprobar que está iniciado el servicio en el administrador de tareas de Windows:

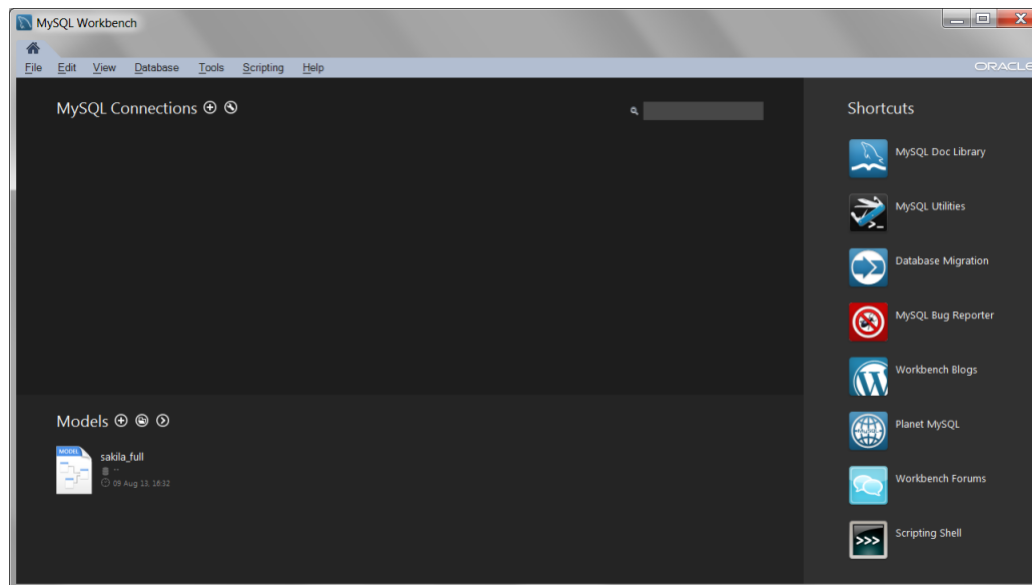




## manejo básico de estructura

### Caso práctico 2-1: Creación de base de datos y tablas

Se pide ejecutar la herramienta de manejo de base de datos MySQL llamada Workbench, desde C:\BpE\MySQL Workbench 6.0.6 CE\MySQLWorkbench.exe:



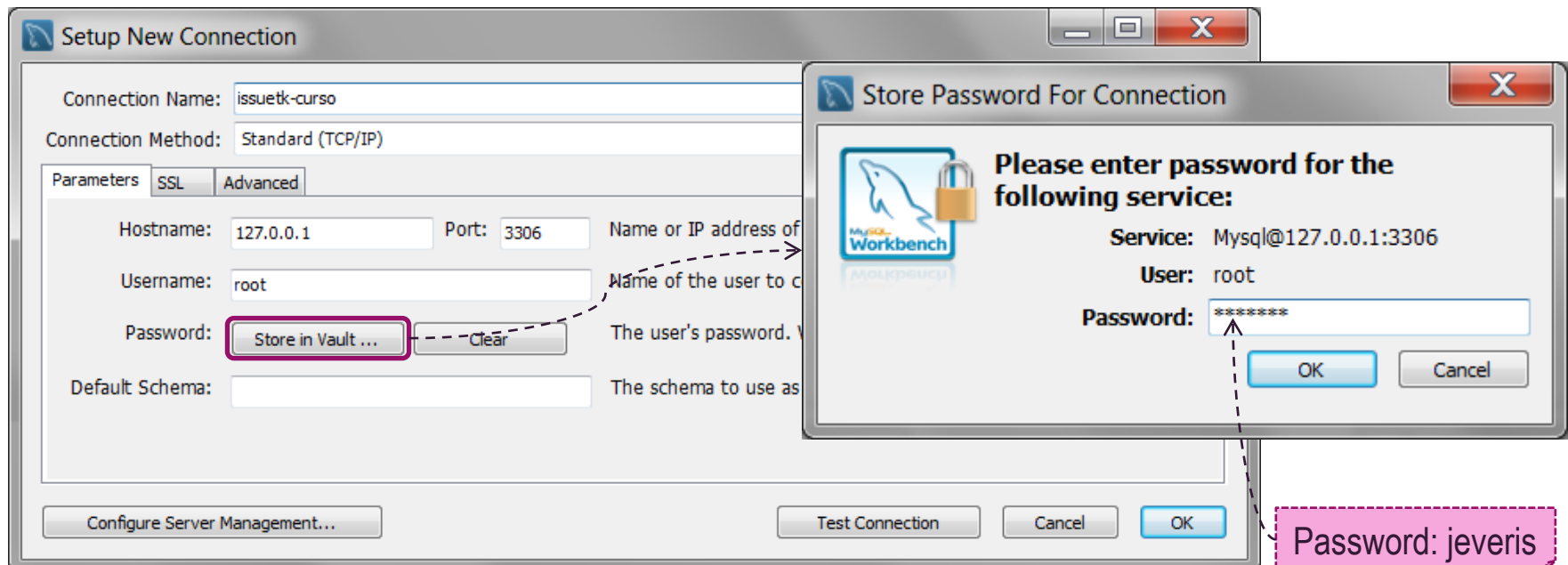
Nótese que la herramienta de manejo está separada del DBMS. Normalmente, el DBMS es un proceso que se ejecuta en alguna máquina, y la herramienta de manejo se conecta al DBMS.

## manejo básico de estructura

### Caso práctico 2-1: Creación de base de datos y tablas

A continuación, se pide configurar la conexión al DBMS:

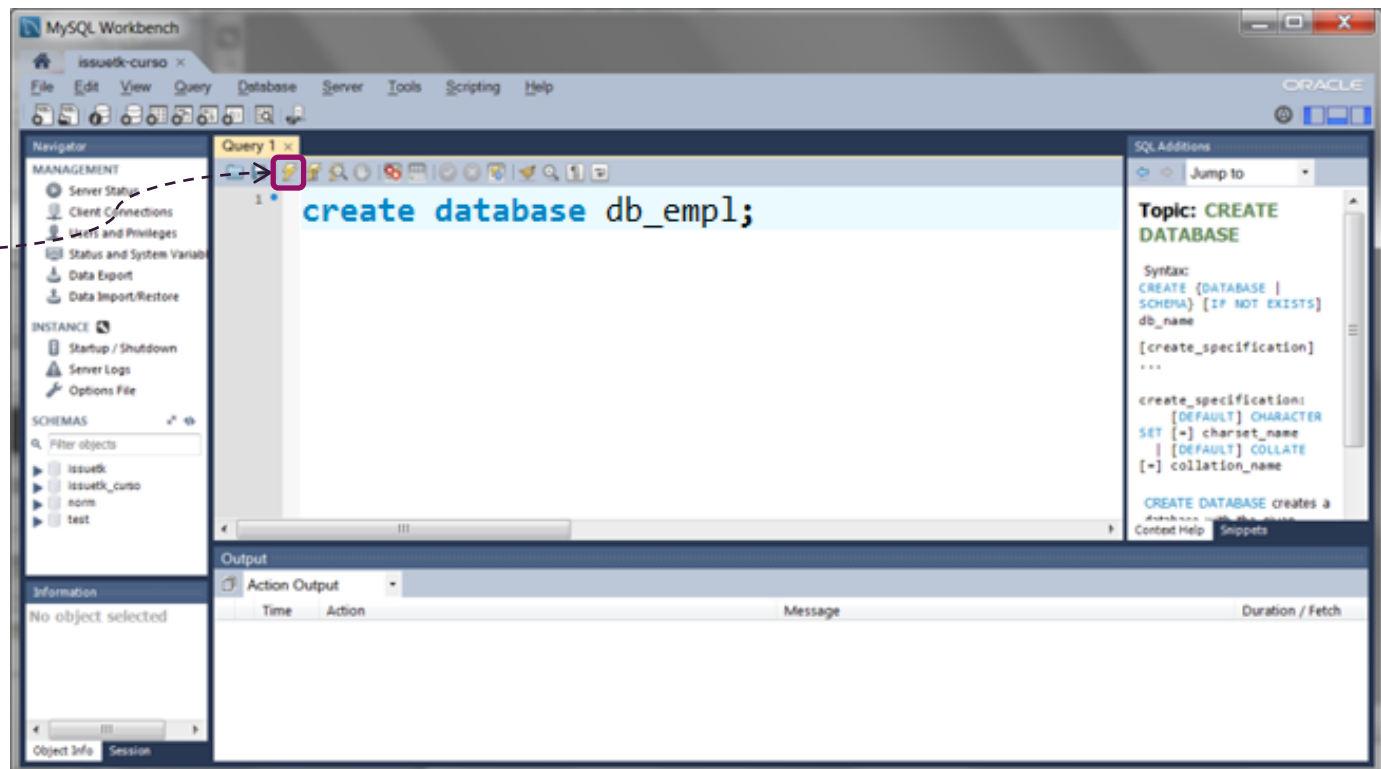
- Presionar el botón 
- Colocar los siguientes datos de conexión:



## manejo básico de estructura

### Caso práctico 2-1: Creación de base de datos y tablas

Crear la base de datos llamada "db\_empl":

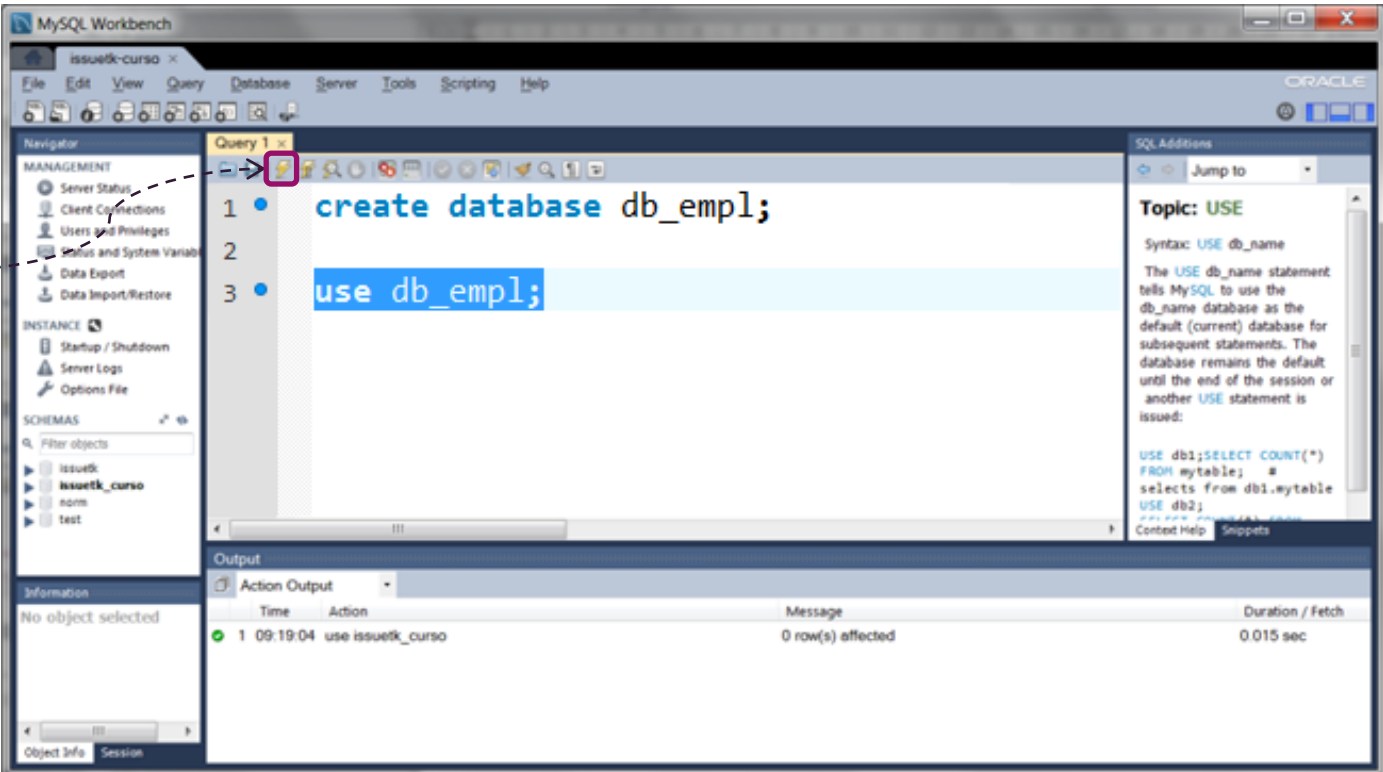


## manejo básico de estructura

### Caso práctico 2-1: Creación de base de datos y tablas

Marcar la base de datos llamada "db\_empl", lo que implica que el resto de las sentencias se ejecutan sobre ésta:

Ejecuta sentencia seleccionada.



The screenshot shows the MySQL Workbench interface. The left sidebar contains a 'Navigator' pane with 'MANAGEMENT' and 'SCHEMAS' sections. The 'SCHEMAS' section shows a tree of databases: 'issuetk', 'issuetk\_curso' (selected), 'norm', and 'test'. A dashed arrow points from the 'issuetk\_curso' database in the sidebar to the 'Query 1' editor. The 'Query 1' editor contains two SQL statements: 'create database db\_empl;' and 'use db\_empl;'. The second statement is highlighted. The 'Output' pane at the bottom shows the execution results for 'use issuetk\_curso', indicating '0 row(s) affected' and a duration of '0.015 sec'. The 'SQL Additions' pane on the right shows the 'USE' topic.

```
create database db_empl;

use db_empl;
```

Time	Action	Message	Duration / Fetch
1 09:19:04	use issuetk_curso	0 row(s) affected	0.015 sec

## manejo básico de estructura

### Caso práctico 2-1: Creación de base de datos y tablas

Utilizando la herramienta, crear sobre la base de datos una tabla de acuerdo a la siguiente definición, utilizando sentencias SQL:

- Nombre: T\_OFFICES
- Campos:
  - OFFC\_ID, tipo INT, obligatorio
  - OFFC\_COUNTRY, tipo texto largo máximo 30, obligatorio
  - OFFC\_DESCRIPTION, tipo texto largo máximo 80, obligatorio
  - OFFC\_NAME, tipo texto largo máximo 30, opcional

Eliminar la tabla, y volver a crearla, colocando en OFFC\_DESCRIPTION un largo máximo de 90.

## manejo básico de estructura

### Caso práctico 2-1: Creación de base de datos y tablas

Una vez creada la tabla T\_OFFICES, realizar las siguientes modificaciones:

- Eliminar campo OFFC\_NAME, pues no es necesario.
- Agregar campo:
  - OFFC\_CITY, tipo texto largo máximo 50, obligatorio
- Modificar campo:
  - OFFC\_DESCRIPTION, largo máximo 100, opcional

Para comprobar que la tabla está correctamente construida, ejecutar DESC T\_OFFICES, y comprobar que entrega el siguiente resultado:

	Field	Type	Null	Key	Default	Extra
►	OFFC_ID	int(11)	NO		HULL	
	OFFC_COUNTRY	varchar(30)	NO		HULL	
	OFFC_CITY	varchar(50)	NO		HULL	
	OFFC_DESCRIPTION	varchar(100)	YES		HULL	



# 3 operaciones básicas sobre datos

- 4. bases de datos relacionales
- 5. manejo de datos relacionales
- 6. otras operaciones sobre datos



## operaciones básicas sobre datos

### Caso práctico 3-1: Lectura y escritura básica

Resumen del ejercicio:

- Insertar, modificar y eliminar registros de tablas, utilizando sentencias SQL.
- Seleccionar registros de una tabla dados criterios de búsqueda.

## operaciones básicas sobre datos

### Caso práctico 3-1: Lectura y escritura básica

Utilizando la herramienta, realizar las siguientes operaciones de escritura:

- Utilizando la tabla creada en el ejercicio anterior, insertar un nuevo registro de T\_OFFICES, utilizando los siguientes valores:
  - ID: 10
  - COUNTRY: España
  - CITY: Madrid
  - DESCRIPTION: Oficina central
- Insertar otro registro (sin campo description):
  - ID: 11
  - COUNTRY: España
  - CITY: Barcelona

## operaciones básicas sobre datos

### Caso práctico 3-1: Lectura y escritura básica

Continuando:

- Agregar en una sola sentencia los siguientes valores:
  - ID: 20
  - COUNTRY: Chile
  - CITY: Santiago
  - DESCRIPTION: Oficina principal de Chile
  - ID: 30
  - COUNTRY: Argentina
  - CITY: Buenos aires
  - DESCRIPTION: NULL
- Seleccionar las columnas ID, COUNTRY y CITY de T\_OFFICES, con los registros que cumplen con contener la palabra 'Oficina' en el campo DESCRIPTION.
- Cambiar "Buenos aires" por "Buenos Aires" en el registro correspondiente, utilizando el ID en la condición.

## operaciones básicas sobre datos

### Caso práctico 3-1: Lectura y escritura básica

Crear la tabla T\_KNOWLEDGE\_LINES y agregarle datos, utilizando las siguientes sentencias:

```
CREATE TABLE `T_KNOWLEDGE_LINES` (  
  `KNLN_ID` INT(11) NOT NULL,  
  `KNLN_NAME` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`KNLN_ID`)  
);
```

```
INSERT INTO `T_KNOWLEDGE_LINES` (`KNLN_ID`, `KNLN_NAME`)  
VALUES (10, 'Java');
```

```
INSERT INTO `T_KNOWLEDGE_LINES` (`KNLN_ID`, `KNLN_NAME`)  
VALUES (20, '.NET');
```

```
INSERT INTO `T_KNOWLEDGE_LINES` (`KNLN_ID`, `KNLN_NAME`)  
VALUES (30, 'Mainframe');
```



# 4 bases de datos relacionales

- **clave primaria**
- estructuración relacional
- otras relaciones

5. manejo de datos relacionales

6. otras operaciones sobre datos

## clave primaria

### Caso práctico 4-1: Uso de clave primaria

Resumen del ejercicio:

- Crear una tabla con clave primaria autogenerada.
- Insertar registros utilizando clave primaria autogenerada.

## clave primaria

### Caso práctico 4-1: Uso de clave primaria

Modificar la tabla T\_OFFICES, colocando OFFC\_ID como clave primaria:

```
ALTER TABLE `T_OFFICES`  
ADD PRIMARY KEY (`OFFC_ID`);
```

Crear la tabla T\_EMPLOYEES, incluyendo su definición de clave primaria autogenerada:

```
CREATE TABLE `T_EMPLOYEES` (  
  `EMPL_ID` INT NOT NULL ... ,  
  `OFFC_ID` INT NOT NULL ,  
  `KNLN_ID` INT ,  
  `EMPL_FORNAME` VARCHAR(50) NOT NULL ,  
  `EMPL_MIDDLE_NAME` VARCHAR(50) ,  
  `EMPL_SURNAME` VARCHAR(50) NOT NULL ,  
  `EMPL_NUMBER` INT NOT NULL ,  
  `EMPL_HIRE_DATE` DATETIME NOT NULL ,  
  `EMPL_MENTOR_ID` INT ,  
  PRIMARY KEY (`EMPL_ID`));
```

Completar



## clave primaria

### Caso práctico 4-1: Uso de clave primaria

Insertar en la tabla T\_EMPLOYEES el siguiente registro:

- OFFC\_ID: 10
- KNLN\_ID: 10
- FORNAME: Juan
- SURNAME: Pérez
- NUMBER: 150
- HIRE\_DATE: 15/abr/2005

Tener en cuenta lo siguiente:

- La clave primaria es autogenerada.
- Los campos no informados se omiten en la sentencia SQL.
- Para especificar una fecha sin hora, se hace en modo texto con formato yyyy-mm-dd



# 4 bases de datos relacionales

- **estructuración relacional**
- otras relaciones

5. manejo de datos relacionales

6. otras operaciones sobre datos

## estructuración relacional

### Caso práctico 4-2: Uso de relaciones

Resumen del ejercicio:

- Utilizar una tabla con clave primaria autogenerada.
- Insertar registros basado en la relación entre tablas.

## clave primaria

### Caso práctico 4-2: Uso de relaciones

Insertar en la tabla T\_EMPLOYEES el siguiente registro:

- Office: la de Barcelona
- Knowledge line: .NET
- FORNAME: Luis
- SURNAME: González
- NUMBER: 160
- HIRE\_DATE: 18/may/2006
- Mentor: Juan Pérez

Tener en cuenta que en las relaciones se especifica como FK la PK del registro asociado.

## clave primaria

### Caso práctico 4-2: Uso de relaciones

Insertar en la tabla T\_EMPLOYEES otro registro:

- Office: la de Santiago de Chile
- FORNAME: Pedro
- SURNAME: García
- NUMBER: 180
- HIRE\_DATE: 18/may/2006



# 4 bases de datos relacionales

- otras relaciones

5. manejo de datos relacionales

6. otras operaciones sobre datos

## otras relaciones

### Caso práctico 4-3: Tipos de relaciones

Resumen del ejercicio:

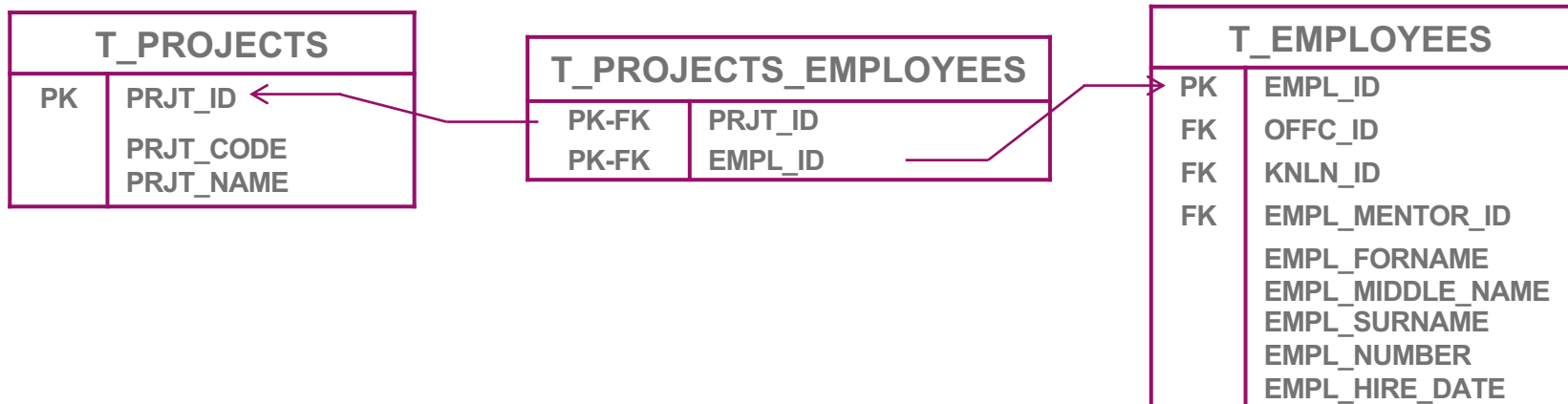
- Crear la tabla de proyectos.
- Crear la tabla de relación intermedia entre proyectos y empleados.
- Añadir datos de proyectos y de asociaciones entre proyecto y empleado.



## otras relaciones

### Caso práctico 4-3: Tipos de relaciones

La relación entre proyectos y empleados es many-to-many, es decir, un empleado puede estar en varios proyectos, y un proyecto puede tener varios empleados:



## otras relaciones

### Caso práctico 4-3: Tipos de relaciones

Crear la tabla de proyectos:

```
CREATE TABLE `T_PROJECTS` (  
  `PRJT_ID` INT NOT NULL AUTO_INCREMENT,  
  `PRJT_CODE` VARCHAR(16) NOT NULL,  
  `PRJT_NAME` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`PRJT_ID`));
```

Crear la tabla de relación intermedia entre proyectos y empleados:

```
CREATE TABLE `T_PROJECTS_EMPLOYEES` (  
  `PRJT_ID` INT NOT NULL,  
  `EMPL_ID` INT NOT NULL,  
  PRIMARY KEY (...));
```

Completar. La PK de la tabla de relación intermedia es compuesta por las dos FK, separadas por coma.

## otras relaciones

### Caso práctico 4-3: Tipos de relaciones

Añadir datos de proyectos:

- CODE: EXT-001000-01234, NAME: Gestión de usuarios
- CODE: INT-001000-03200, NAME: Cursos de formación

Añadir asociaciones entre proyecto y empleado:

- Juan Pérez se asocia con los proyectos Gestión de usuarios y Cursos de formación
- Luis González se asocia al proyecto Gestión de usuarios.



# 5 manejo de datos relacionales

- **claves foráneas**
- joins
- vista

## 6. otras operaciones sobre datos

## claves foráneas

### Caso práctico 5-1: Creación de claves foráneas

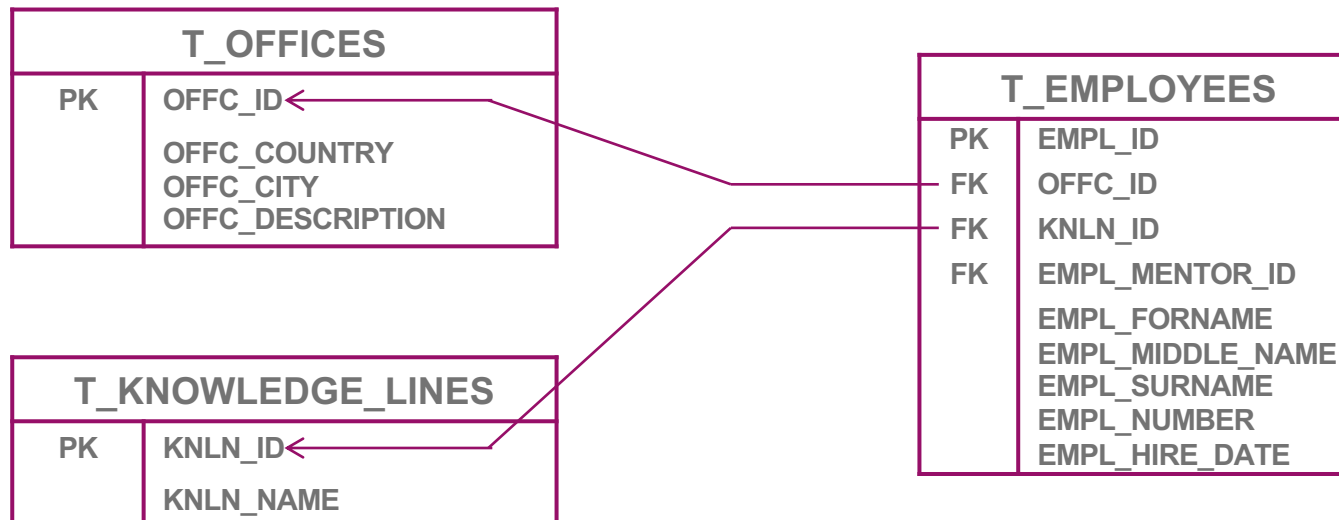
Resumen del ejercicio:

- Crear las claves foráneas especificadas en el manual para tablas del modelo.
- Comprobar la integridad referencial.
- Crear las claves foráneas de una relación many-to-many.

## claves foráneas

### Caso práctico 5-1: Creación de claves foráneas

Crear las siguientes claves foráneas, de las relaciones many-to-one, utilizando las sentencias de la lámina siguiente.



## claves foráneas

### Caso práctico 5-1: Creación de claves foráneas

Creación de claves foráneas. Las sentencias son las colocadas en el manual:

- Desde T\_EMPLOYEE a T\_OFFICE, con el nombre FK\_EMPL\_OFFC:

```
ALTER TABLE T_EMPLOYEES
  ADD INDEX FK_EMPL_OFFC (OFFC_ID) ,
  ADD CONSTRAINT FK_EMPL_OFFC
    FOREIGN KEY (OFFC_ID)
    REFERENCES T_OFFICES (OFFC_ID) ;
```

- Desde T\_EMPLOYEE a T\_KNOWLEDGE\_LINES, con el nombre FK\_EMPL\_KNLN:

```
ALTER TABLE T_EMPLOYEES
  ADD INDEX FK_EMPL_KNLN (KNLN_ID) ,
  ADD CONSTRAINT FK_EMPL_KNLN
    FOREIGN KEY (KNLN_ID)
    REFERENCES T_KNOWLEDGE_LINES (KNLN_ID) ;
```

## claves foráneas

### Caso práctico 5-1: Creación de claves foráneas

Comprobar la integridad referencial, verificando que las siguientes operaciones **producen error**:

- Modificar la oficina del usuario Juan Pérez por el valor 40 (que no existe).
- Eliminar la oficina 10, que es referenciada por un usuario.

Si alguna de las operaciones anteriores se logra ejecutar, entonces la creación de la clave foránea no es correcta.



## claves foráneas

### Caso práctico 5-1: Creación de claves foráneas

Crear las siguientes claves foráneas, de la relación many-to-many:

- Desde T\_PROJECTS\_EMPLOYEES a T\_PROJECTS, nombre FK\_PREM\_PRJT.
- Desde T\_PROJECTS\_EMPLOYEES a T\_EMPLOYEES, nombre FK\_PREM\_EMPL.





# 5 manejo de datos relacionales

- joins
- vista

## 6. otras operaciones sobre datos

## joins

### Caso práctico 5-2: Uso de join

Resumen del ejercicio:

- Realizar consultas con inner join y left outer join.
- Se utilizan relaciones tipo many-to-one y many-to-many.

## joins

### Caso práctico 5-2: Uso de join

Se pide realizar la siguiente consulta, con tabla principal T\_EMPLOYEES:

- Campos a obtener:
  - Nombre del empleado (FORNAME)
  - Apellido del empleado (SURNAME)
  - Ciudad de su oficina
  - Nombre de línea de conocimientos, si tiene, o NULL si no
- Condiciones:
  - El país es España o Chile
  - No tiene mentor
- Resultado esperado:

	EMPL FORNAME	EMPL SURNAME	OFFC CITY	KNLN NAME
▶	Juan	Pérez	Madrid	Java
	Pedro	García	Santiago	NULL

## joins

### Caso práctico 5-2: Uso de join

Se pide realizar la siguiente consulta:

- Campos a obtener:
  - Código del proyecto
  - Nombre del proyecto
  - Número de empleado
  - Apellido de empleado
- Condiciones:
  - El nombre del empleado es Juan
- Resultado esperado:

	PRJT CODE	PRJT NAME	EMPL NUMBER	EMPL SURNAME
▶	EXT-001000-01234	Gestión de usuarios	150	Pérez
	INT-001000-03200	Cursos de formación	150	Pérez



# 5 manejo de datos relacionales

- vista

6. otras operaciones sobre datos

## **vista**

### Caso práctico 5-3: Uso de vistas

Resumen del ejercicio:

- Crear una vista dada la definición de su consulta asociada.
- Realizar consultas sobre la vista.

## vista

### Caso práctico 5-3: Uso de vistas

Crear una vista con las siguientes características:

- Nombre: V\_PROJECTS\_EMPLOYEES
- Campos:
  - Código del proyecto
  - Nombre del proyecto
  - Número de empleado
  - Nombre de empleado
  - Apellido de empleado

**Nota:** Se recomienda reutilizar la consulta del ejercicio de join para la creación de la vista.



## vista

### Caso práctico 5-3: Uso de vistas

Consultar todos los datos de la vista. El resultado esperado es:

	PRJT_CODE	PRJT_NAME	EMPL_NUMBER	EMPL_FORNAME	EMPL_SURNAME
►	EXT-001000-01234	Gestión de usuarios	150	Juan	Pérez
	EXT-001000-01234	Gestión de usuarios	160	Luis	González
	INT-001000-03200	Cursos de formación	150	Juan	Pérez

Realizar una consulta sobre la vista, que retorne todos los campos para la condición que el código de proyecto comience con EXT.



# 6 otras operaciones sobre datos

## **otras operaciones sobre datos**

### Caso práctico 6-1: Agrupación, ordenamiento y paginación

Resumen del ejercicio:

- Crear una consulta que utiliza agrupación
- Agregarle ordenamiento
- Agregarle paginación

## otras operaciones sobre datos

### Caso práctico 6-1: Agrupación, ordenamiento y paginación

Crear la tabla de documentos:

```
CREATE TABLE `T_DOCUMENTS` (  
  `DOCS_ID` INT NOT NULL AUTO_INCREMENT,  
  `EMPL_ID` INT NOT NULL,  
  `DOCS_NAME` VARCHAR(100) NOT NULL,  
  `DOCS_TYPE` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`DOCS_ID`));
```

Crear la clave foránea hacia la tabla de empleados:

```
ALTER TABLE T_DOCUMENTS ADD INDEX FK_DOCS_EMPL (EMPL_ID),  
  ADD CONSTRAINT FK_DOCS_EMPL  
    FOREIGN KEY (EMPL_ID) REFERENCES T_EMPLOYEES (EMPL_ID);
```

Crear la constraint para el tipo de documento:

```
ALTER TABLE T_DOCUMENTS  
  ADD CONSTRAINT CT_CK_DOCS_TYPE  
    CHECK (DOCS_TYPE IN ('PDF', 'DOC', 'XLS'));
```

## otras operaciones sobre datos

### Caso práctico 6-1: Agrupación, ordenamiento y paginación

Agregar los siguientes documentos:

```
INSERT INTO `T_DOCUMENTS`  
(`EMPL_ID`, `DOCS_NAME`, `DOCS_TYPE`)  
VALUES  
(1, 'Titulo', 'PDF'),  
(1, 'Curriculum', 'DOC'),  
(1, 'Certificado OCP', 'PDF'),  
(1, 'Matriz conocimientos', 'XLS'),  
(2, 'Grado', 'PDF'),  
(2, 'Curriculum', 'DOC'),  
(2, 'Certificado MS', 'PDF'),  
(3, 'Titulo', 'PDF');
```

## otras operaciones sobre datos

### Caso práctico 6-1: Agrupación, ordenamiento y paginación

Construir y ejecutar una consulta con las siguientes características:

- Campos a obtener:
  - Nombre y apellido del empleado, separados por espacio, alias FULL\_NAME.
  - Número de documentos asociado a cada empleado, alias NUM\_DOCS
- Resultado esperado:

	FULL_NAME	NUM_DOCS
▶	Juan Pérez	4
	Luis González	3
	Pedro García	1

## otras operaciones sobre datos

### Caso práctico 6-1: Agrupación, ordenamiento y paginación

A la consulta anterior, **ordenarla** en forma ascendente por número de documentos:

Resultado esperado:

	FULL NAME	NUM DOCS
►	Pedro García	1
	Luis González	3
	Juan Pérez	4

## otras operaciones sobre datos

### Caso práctico 6-1: Agrupación, ordenamiento y paginación

A la consulta anterior ordenada, extraerle el segundo y tercer registro, utilizando **paginación**:

Resultado esperado:

	FULL NAME	NUM DOCS
►	Luis González	3
	Juan Pérez	4



CORPORATE  
UNIVERSITY

