

**DATABASES**  
**SQL**  
**Data Definition Language**

TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO PARA A  
WEB

# Agenda

## ❖ SQL – Data Definition Language

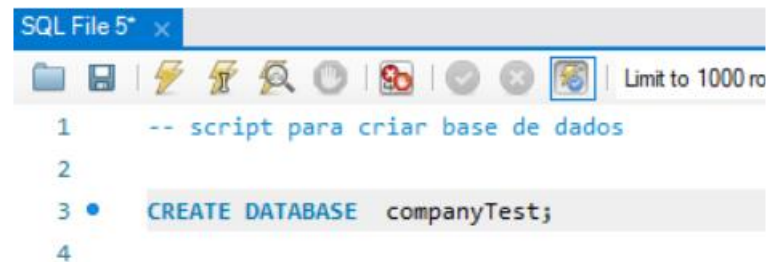
- ❖ CREATE
- ❖ USE
- ❖ SHOW
- ❖ PRIMARY KEY
- ❖ FOREIGN KEY
- ❖ REFERENCES
- ❖ DROP
- ❖ ALTER



## ❖ SQL - Data Definition Language

### ❖ Create a database: **CREATE**

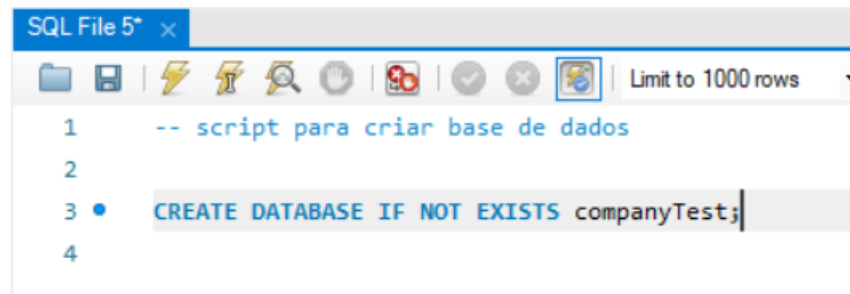
`CREATE DATABASE [IF NOT EXISTS] database_name`



The screenshot shows a SQL editor window titled "SQL File 5\* x". The toolbar includes icons for file operations, execution, and a "Limit to 1000 rows" dropdown. The script content is as follows:

```
1  -- script para criar base de dados
2
3  • CREATE DATABASE companyTest;
4
```

### ❖ If you have doubts about the existence (or not) of the DB, we can use the suffix IF NOT EXISTS



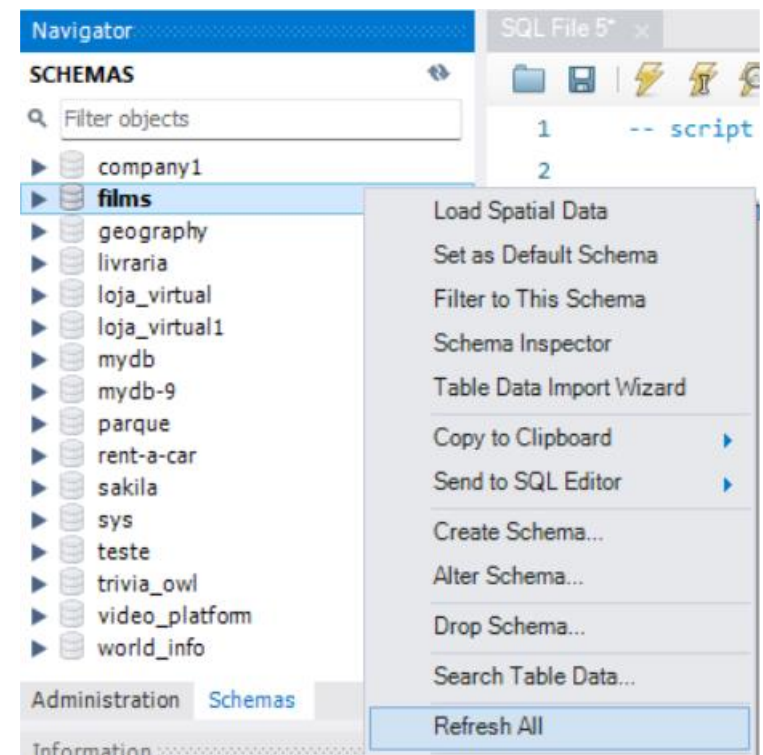
The screenshot shows a SQL editor window titled "SQL File 5\* x". The toolbar includes icons for file operations, execution, and a "Limit to 1000 rows" dropdown. The script content is as follows:

```
1  -- script para criar base de dados
2
3  • CREATE DATABASE IF NOT EXISTS companyTest;
4
```

## ❖ SQL - Data Definition Language

the list of schemas is not updated automatically...

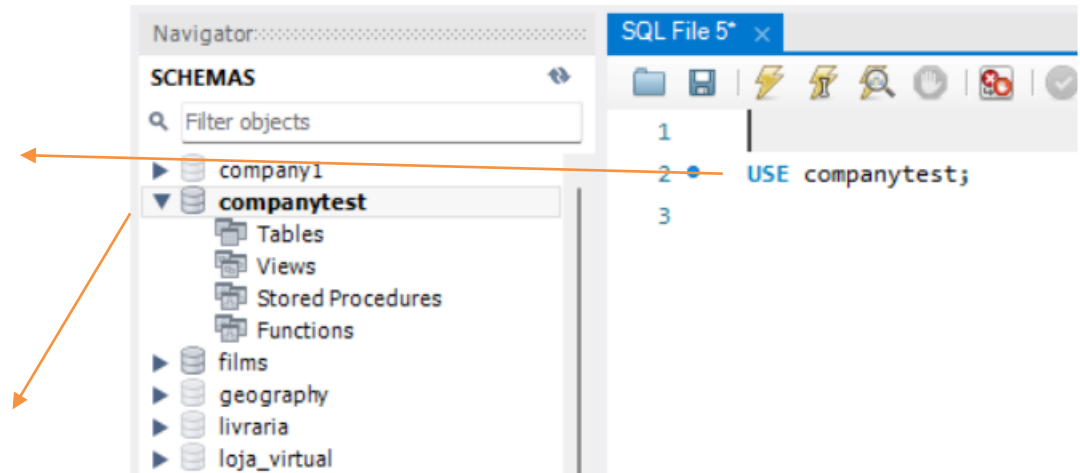
Refresh the database schemas list



## ❖ SQL - Data Definition Language

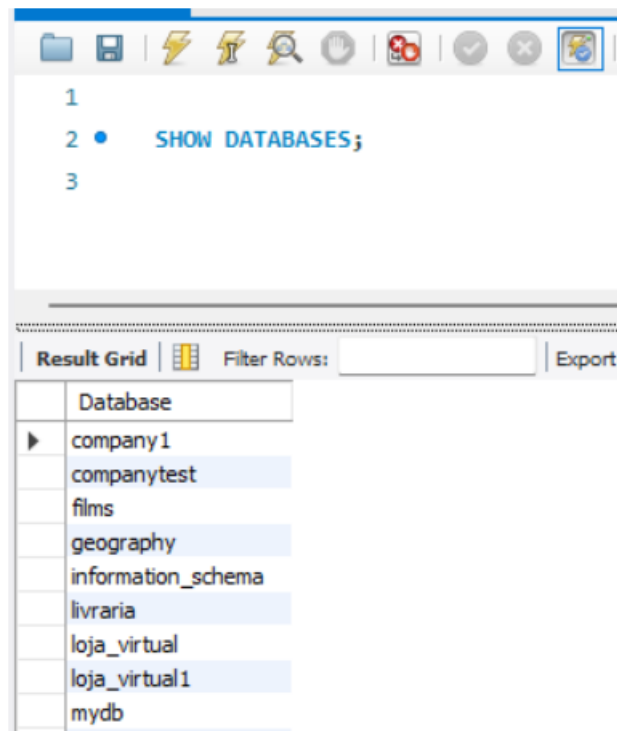
- ❖ Select a database  
(using sql language): **USE**

- ❖ Select a BD in Workbench:  
Double click in database name

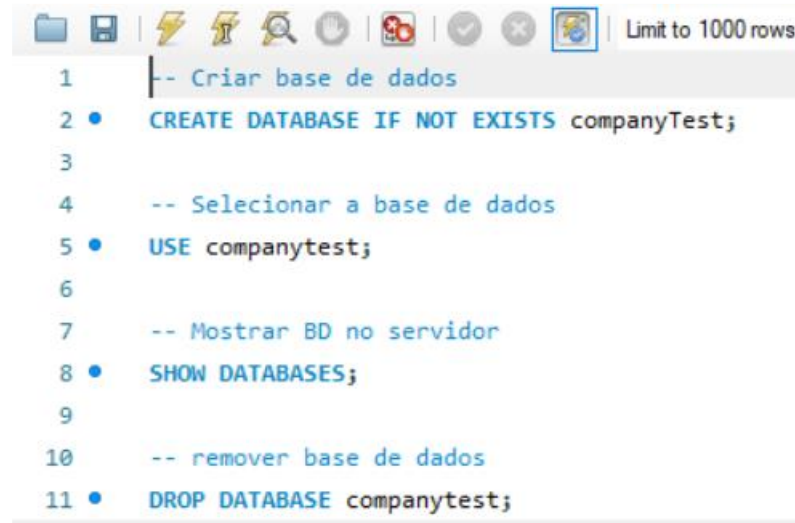
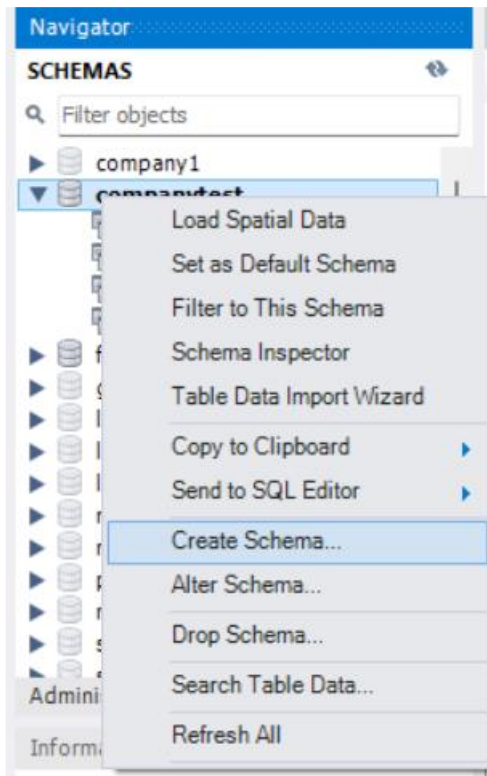


## ❖ SQL - Data Definition Language

Show all databases on the server: **SHOW DATABASES**

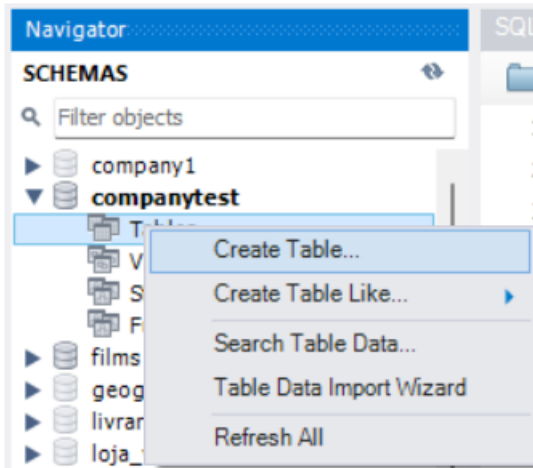


## ❖ SQL - Data Definition Language



## ❖ SQL - Data Definition Language

### ❖ Create a table: **CREATE TABLE**



```
CREATE [IF NOT EXISTS] TABLE name_tabel (  
    attribute type [properties],  
    attribute type [properties],  
    [PRIMARY KEY (attribute)]  
    [FOREIGN KEY (attribute) REFERENCES table (attribute)]  
        [ON DELETE  
        ON UPDATE]  
)
```



❖ Create a table: **CREATE TABLE**

[illegible]

## ❖ SQL - Data Definition Language

### ❖ Create a table: **CREATE TABLE**

```
1  -- script criar tabela employee
2
3  CREATE TABLE IF NOT EXISTS companytest.employee (
4      employee_id      INT NOT NULL,
5      employee_name     VARCHAR(30) NOT NULL,
6      birth_date        DATE NULL,
7      postalcode_id     VARCHAR(8) NOT NULL,
8      admission_date    DATE NULL,
9      nif               INT NOT NULL,
10     department_id     INT NOT NULL,
11     PRIMARY KEY       (employee_id)
12
13 );
```

[illegible]

```
2 -- script criar tabela department
3 CREATE TABLE IF NOT EXISTS companytest.department (
4     department_id      INT NOT NULL,
5     department_name     VARCHAR(30) NOT NULL,
6     budget              INT NULL,
7     PRIMARY KEY (department_id)
8 );
```

[illegible]

## ❖ SQL Data Definition Language

### ❖ Create a Table: **DEFAULT**

```
-- script criar tabela department
CREATE TABLE IF NOT EXISTS companytest.department (
  department_id INT NOT NULL,
  department_name VARCHAR(30) NOT NULL,
  budget INT NOT NULL DEFAULT 5000,
  PRIMARY KEY (department_id)
);
```

**DEFAULT** may be associated with:

- ☐ Literal Values
- ☐ NULL
- ☐ System Values (system date, e.g.)

Table Name: department Schema: companytest

Charset/Collation: utf8mb4 utf8mb4\_0900\_ai\_ci Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
department_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
department_name	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
budget	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5000

↓  
Default value

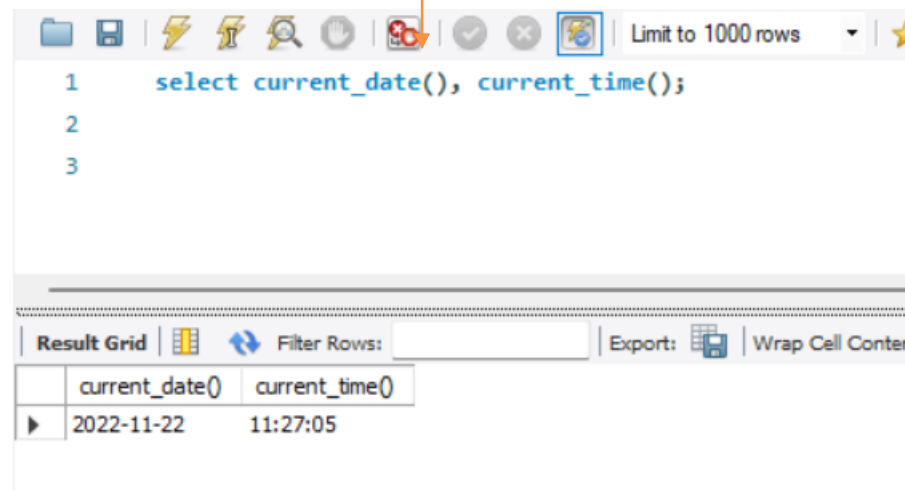
## ❖ SQL Data Definition Language

### ❖ Create a Table: **DEFAULT**

```
-- script criar tabela department
CREATE TABLE IF NOT EXISTS companytest.department (
  department_id    INT NOT NULL,
  department_name  VARCHAR(30) NOT NULL,
  budget          INT NOT NULL DEFAULT 5000,
  PRIMARY KEY (department_id)
);
```

**DEFAULT** may be associated with:

- ☐ Literal Values
- ☐ NULL
- ☐ System Values (system date, e.g.)



## ❖ SQL Data Definition Language

### ❖ Create a Table: **CONSTRAINTS**

#### **CONSTRAINTS**

- ☐ Rules to which the values of one or more attributes must respect.
- ☐ Ensure that DB data respects defined constraints.

For example:

- ☐ Gender attribute (is exists) could only assume 'M' or 'F';
- ☐ Salary must be between 500 and 5000.
- ☐ Attribute with unique values
- ☐ Foreign keys to implement relations between entities

## ❖ SQL Data Definition Language

### ❖ Create a table: **CREATE TABLE**

#### **CONSTRAINTS**

- ❑ **NOT NULL** is the most used constraint (already seen in previous examples)
- ❑ **CHECK** allows you to perform data validation on an attribute

Some examples...

```
-- script criar tabela department
CREATE TABLE IF NOT EXISTS companytest.department (
  department_id    INT NOT NULL,
  department_name  VARCHAR(30) NOT NULL,
  budget           INT NOT NULL CHECK (budget >0),
  PRIMARY KEY (department_id)
);
```

```
-- script criar tabela department
CREATE TABLE IF NOT EXISTS companytest.department (
  department_id    INT NOT NULL,
  department_name  VARCHAR(30) NOT NULL,
  budget           INT NOT NULL CHECK (budget BETWEEN 0 AND 5000),
  PRIMARY KEY (department_id)
);
```

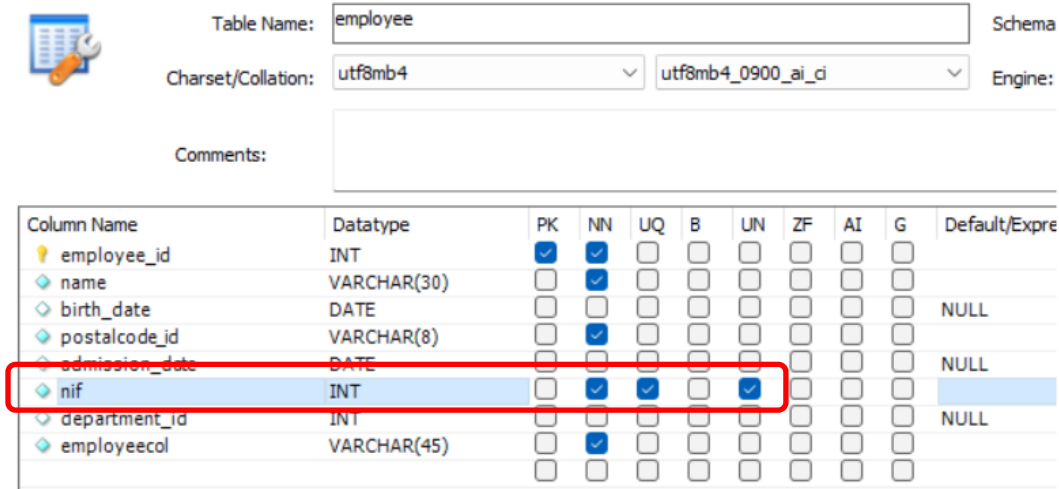


## ❖ SQL Data Definition Language

## ❖ Create a table: **CREATE TABLE**

### CONSTRAINTS

☐ **UNIQUE** indicates that the value of this column is **unique**, can not be repeated.  
For example: NIF



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expre
employee_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
birth_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
postalcode_id	VARCHAR(8)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
admission_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
nif	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
department_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
employeeecol	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```
• CREATE TABLE IF NOT EXISTS companytest.employee (  
  employee_id INT NOT NULL,  
  employee_name VARCHAR(30) NOT NULL,  
  birth_date DATE NULL,  
  postalcode_id VARCHAR(8) NOT NULL,  
  admission_date DATE NULL,  
  nif INT NOT NULL UNIQUE,  
  department_id INT NOT NULL,  
  PRIMARY KEY (employee_id)  
);
```

## ❖ SQL Data Definition Language

## ❖ Create a table: **CREATE TABLE**









### **CONSTRAINTS**

#### ☐ **AUTO\_INCREMENT**

Table Name:  Schema: **companyt**

Charset/Collation:   Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 employee_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 name	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 birth_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 postalcode_id	VARCHAR(8)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 admission_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 nif	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 department_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 employeeecol	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```
1  -- script criar tabela employee
2
3  CREATE TABLE IF NOT EXISTS companytest.employee (
4  employee_id INT NOT NULL AUTO_INCREMENT,
5  employee_name VARCHAR(30) NOT NULL,
6  birth_date DATE NULL,
7  postalcode_id VARCHAR(8) NOT NULL,
8  admission_date DATE NULL,
9  nif INT NOT NULL UNIQUE,
10 department_id INT NOT NULL,
```

## ❖ SQL Data Definition Language

### ❖ Create a table: **CREATE TABLE**

#### **CONSTRAINTS**

#### **❑ FOREIGN KEY**

Entity with which  
it relates

Table Name:  Schema: **companytest**

Charset/Collation:   Engine:

Comments:

Foreign Key Name	Referenced Table
department_id	'companytest`.`department`'

Column	Referenced Column
<input type="checkbox"/> employee_id	
<input type="checkbox"/> employee_name	
<input type="checkbox"/> birth_date	
<input type="checkbox"/> postalcode_id	
<input type="checkbox"/> admission_date	
<input type="checkbox"/> nif	
<input checked="" type="checkbox"/> department_id	department_id

Columns Indexes **Foreign Keys** Triggers Partitioning Options

## ❖ SQL Data Definition Language

### ❖ Create a table: CREATE TABLE

#### CONSTRAINTS

#### ❑ FOREIGN KEY



```
1  -- script criar tabela employee
2
3  CREATE TABLE IF NOT EXISTS companytest.employee1 (
4      employee_id      INT NOT NULL AUTO_INCREMENT,
5      employee_name     VARCHAR(30) NOT NULL,
6      birth_date        DATE NULL,
7      postalcode_id     VARCHAR(8) NOT NULL,
8      admission_date    DATE NULL,
9      nif               INT NOT NULL UNIQUE,
10     department_id     INT NOT NULL,
11     PRIMARY KEY       (employee_id),
12     FOREIGN KEY        (department_id) REFERENCES companytest.department (department_id)
13
14 );
15
```

## ❖ SQL Data Definition Language

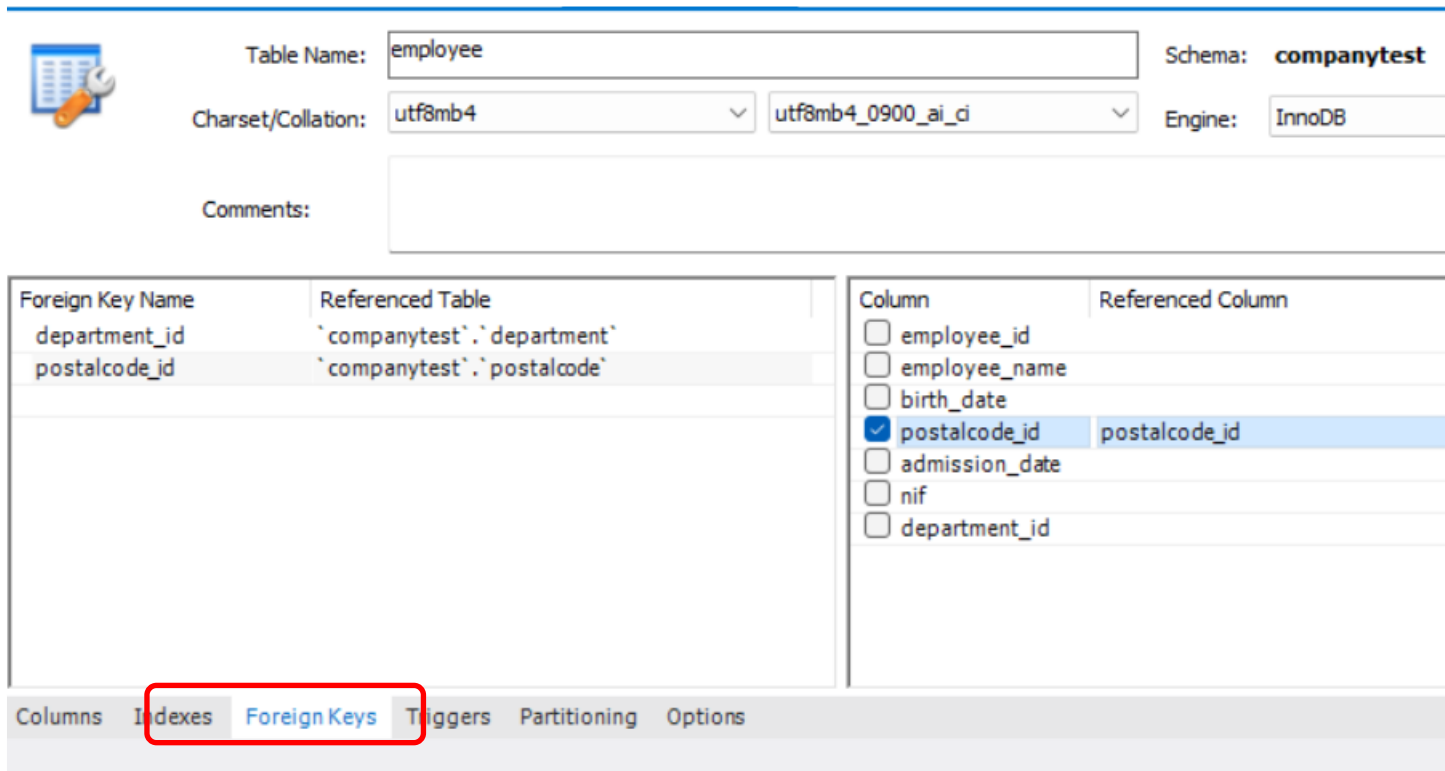


Table Name:  Schema: **companytest**

Charset/Collation:   Engine:

Comments:

Foreign Key Name	Referenced Table
department_id	`companytest`.`department`
postalcode_id	`companytest`.`postalcode`

Column	Referenced Column
<input type="checkbox"/> employee_id	
<input type="checkbox"/> employee_name	
<input type="checkbox"/> birth_date	
<input checked="" type="checkbox"/> postalcode_id	postalcode_id
<input type="checkbox"/> admission_date	
<input type="checkbox"/> nif	
<input type="checkbox"/> department_id	

Columns | **Indexes** | **Foreign Keys** | Triggers | Partitioning | Options

## ❖ SQL Data Definition Language

### ❖ Create a table: CREATE TABLE

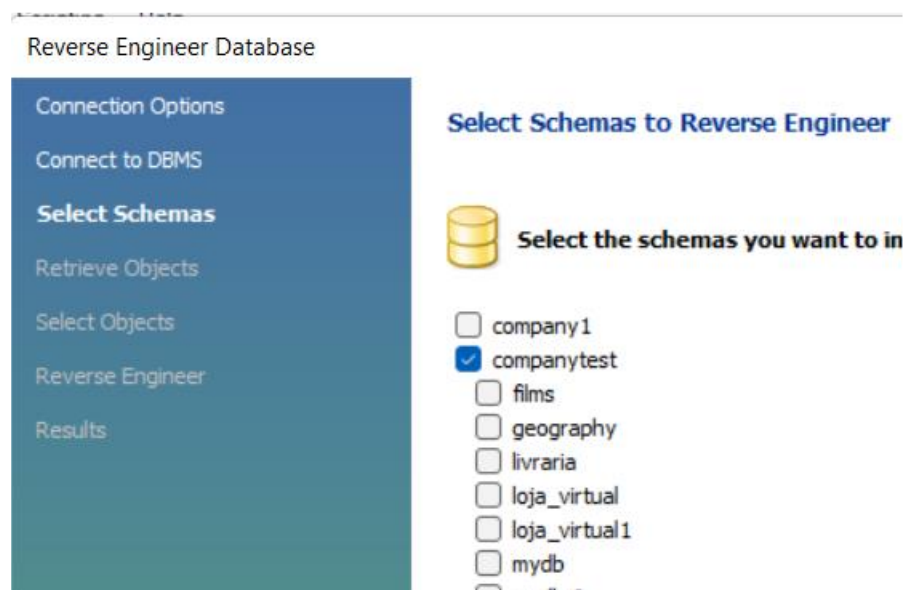
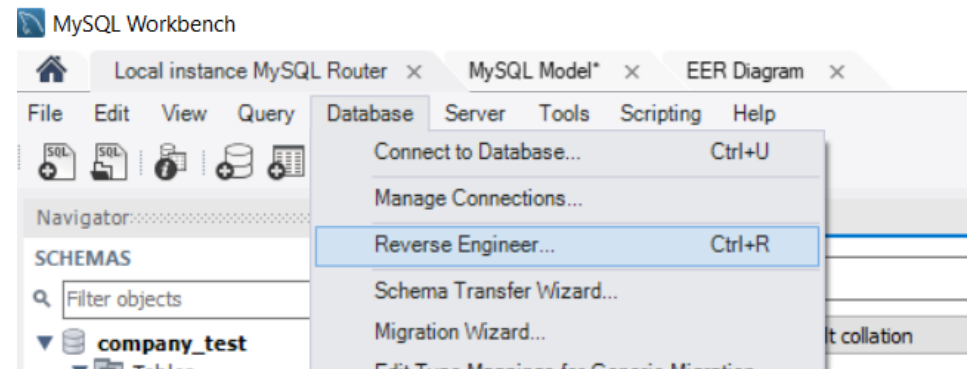
#### CONSTRAINTS

#### ❑ FOREIGN KEY

```
1  -- script criar tabela employee
2
3  ● ○ CREATE TABLE IF NOT EXISTS companytest.employee (
4      employee_id      INT NOT NULL AUTO_INCREMENT,
5      employee_name     VARCHAR(30) NOT NULL,
6      birth_date        DATE NULL,
7      postalcode_id     VARCHAR(8) NOT NULL,
8      admission_date    DATE NULL,
9      nif               INT NOT NULL UNIQUE,
10     department_id     INT NOT NULL,
11     PRIMARY KEY       (employee_id),
12     FOREIGN KEY        (department_id) REFERENCES companytest.department (department_id),
13     FOREIGN KEY        (postalcode_id) REFERENCES companytest.postalcode(postalcode_id)
14
15 );
```

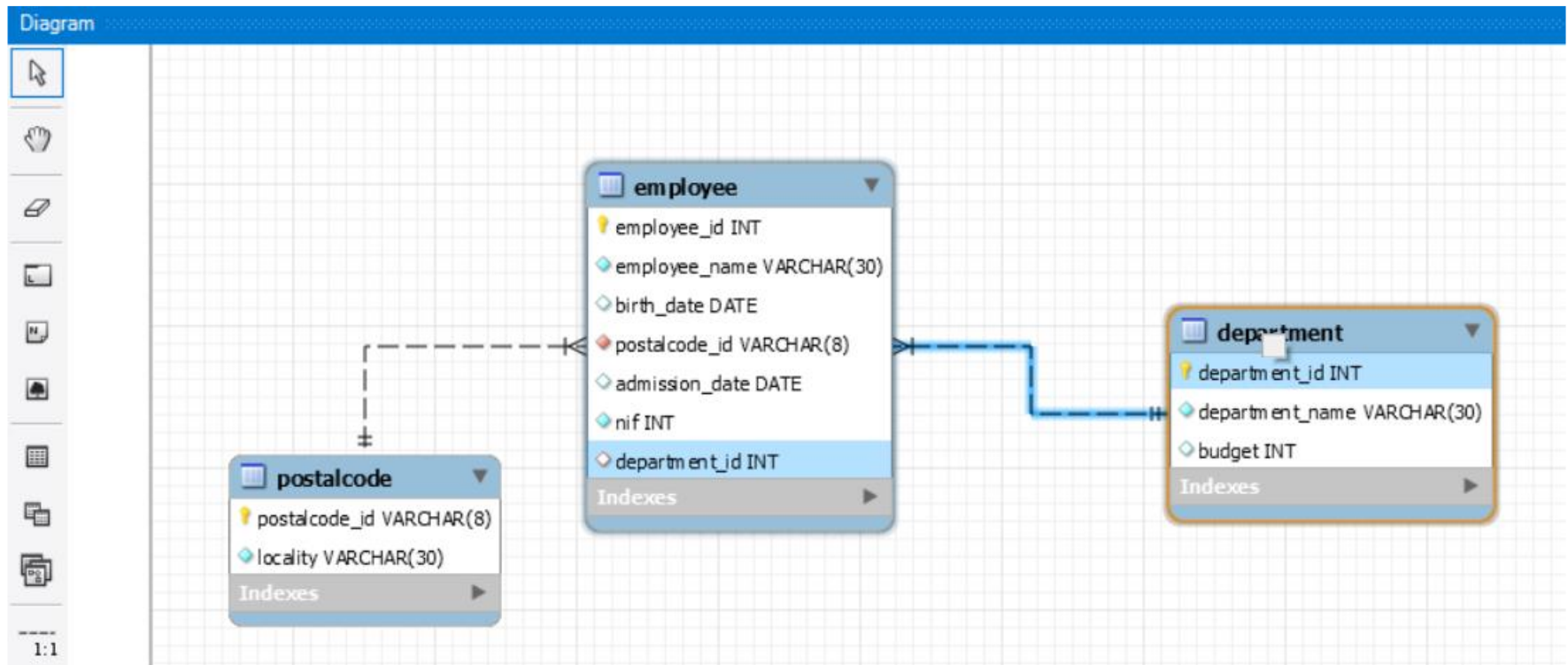
## ❖ FOREIGN KEY

To get / verify the database schema...



## ❖ FOREIGN KEY

To get / verify the database schema...





## ❖ FOREIGN KEY ... REFERENCES

### ❖ ON DELETE / ON UPDATE

- ❑ Let's you enforce referential constraints between tables.
- ❑ Defines how changes in a table are propagated in the tables with which it relates.

It could include:

Table Name:  Schema: **companytest**

Charset/Collation:   Engine:

Comments:

Foreign Key Name	Referenced Table	Column	Referenced Column
department_id	`companytest`.`department`	<input type="checkbox"/> employee_id	
postalcode_id	`companytest`.`postalcode`	<input type="checkbox"/> employee_name	
		<input type="checkbox"/> birth_date	
		<input checked="" type="checkbox"/> postalcode_id	postalcode_id
		<input type="checkbox"/> admission_date	
		<input type="checkbox"/> nif	
		<input type="checkbox"/> department_id	

Foreign Key Options

On Update:

On Delete:

Foreign Key Comm:

Foreign Key Options:

Foreign Key Comm:

## ❖ FOREIGN KEY ... REFERENCES

### ❖ ON DELETE / ON UPDATE

- ☐ Let's you enforce referential constraints between tables.
- ☐ Defines how changes in a table are propagated in the tables with which it relates.

It could include :

- ❖ **NO ACTION:** Changes or deletes the record of the table without implications for the related entity, unless you make a change to a value that does not exist in the related entity. NO ACTION is deferred in time, tested after statement.
- ❖ **RESTRICT (by default):** Rejects the delete or update operation for the parent table. Specifying RESTRICT is the same as omitting the ON DELETE or ON UPDATE clause. Acts immediately.

## ❖ FOREIGN KEY ... REFERENCES

### ❖ ON DELETE / ON UPDATE

- ❖ **SET NULL**: Deletes or changes the row in the table and sets the foreign key column in the related table to Null.
- ❖ **CASCADE**: Changes or deletes the record from the table and propagates those changes to the related entity

The screenshot shows the configuration for the 'employee' table in the 'company\_test' schema. The table has columns: employee\_id, name, birth\_date, salary, postalcode\_id, admission\_date, nif, and department\_id. A foreign key is defined on the 'department\_id' column, referencing the 'department' table in the 'company\_test' schema. The 'Foreign Key Options' dropdown menu is open, showing the following options: NO ACTION, RESTRICT, CASCADE, SET NULL, and NO ACTION. The 'SET NULL' option is highlighted.

Foreign Key Name	Referenced Table
department_id	'company_test'. 'department'
postalcode_id	'company_test'. 'postalcode'

Column	Referenced Column
<input type="checkbox"/> employee_id	
<input type="checkbox"/> name	
<input type="checkbox"/> birth_date	
<input type="checkbox"/> salary	
<input type="checkbox"/> postalcode_id	
<input type="checkbox"/> admission_date	
<input type="checkbox"/> nif	
<input checked="" type="checkbox"/> department_id	department_id

Foreign Key Options

On Update: NO ACTION

On Delete: RESTRICT, CASCADE, SET NULL, NO ACTION

☐ Skip in SQL generation

## ❖ DROP

- ❑ To remove objects in a database: database, table or other objects



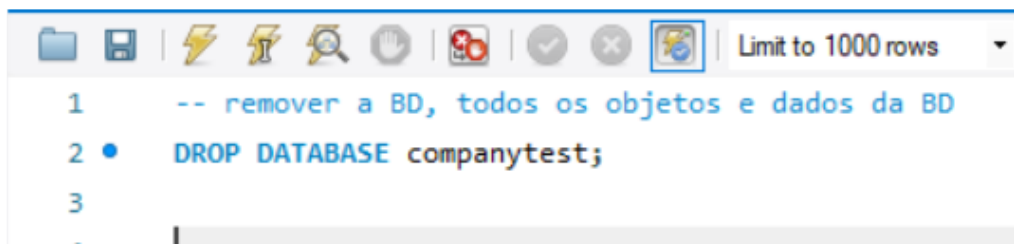
- ❑ The DROP command can not be reversed! Once executed, all database data is removed!
- ❑ It is not possible to reverse the operation with a ROLLBACK, as it happens with UPDATE, for example.
- ❑ Removing database is removing all database tables.

## ❖ DROP

```
DROP DATABASE database_name;
```



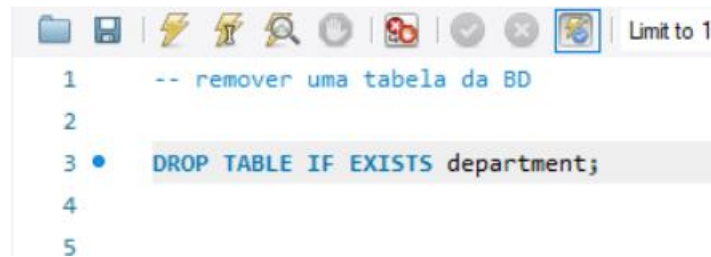
- ☐ The DROP command can not be reversed! Once executed, all database data is removed!
- ☐ It is not possible to reverse the operation with a ROLLBACK, as it happens with UPDATE.
- ☐ Removing database is removing all database tables.



```
1  -- remover a BD, todos os objetos e dados da BD
2  • DROP DATABASE companytest;
3
4  |
```

## ❖ Remove a table: **DROP**

```
DROP TABLE [IF EXISTS] table_name;
```



The screenshot shows a SQL editor window with a toolbar at the top containing icons for file operations, execution, and search. The main text area displays a SQL command on line 3: `DROP TABLE IF EXISTS department;`. The command is highlighted with a light blue background. Line 1 contains a comment: `-- remover uma tabela da BD`. The line numbers 1 through 5 are visible on the left side of the editor.

```
1      -- remover uma tabela da BD
2
3  •   DROP TABLE IF EXISTS department;
4
5
```

## ❖ **ALTER:** Changing the structure or characteristics of an object in the database

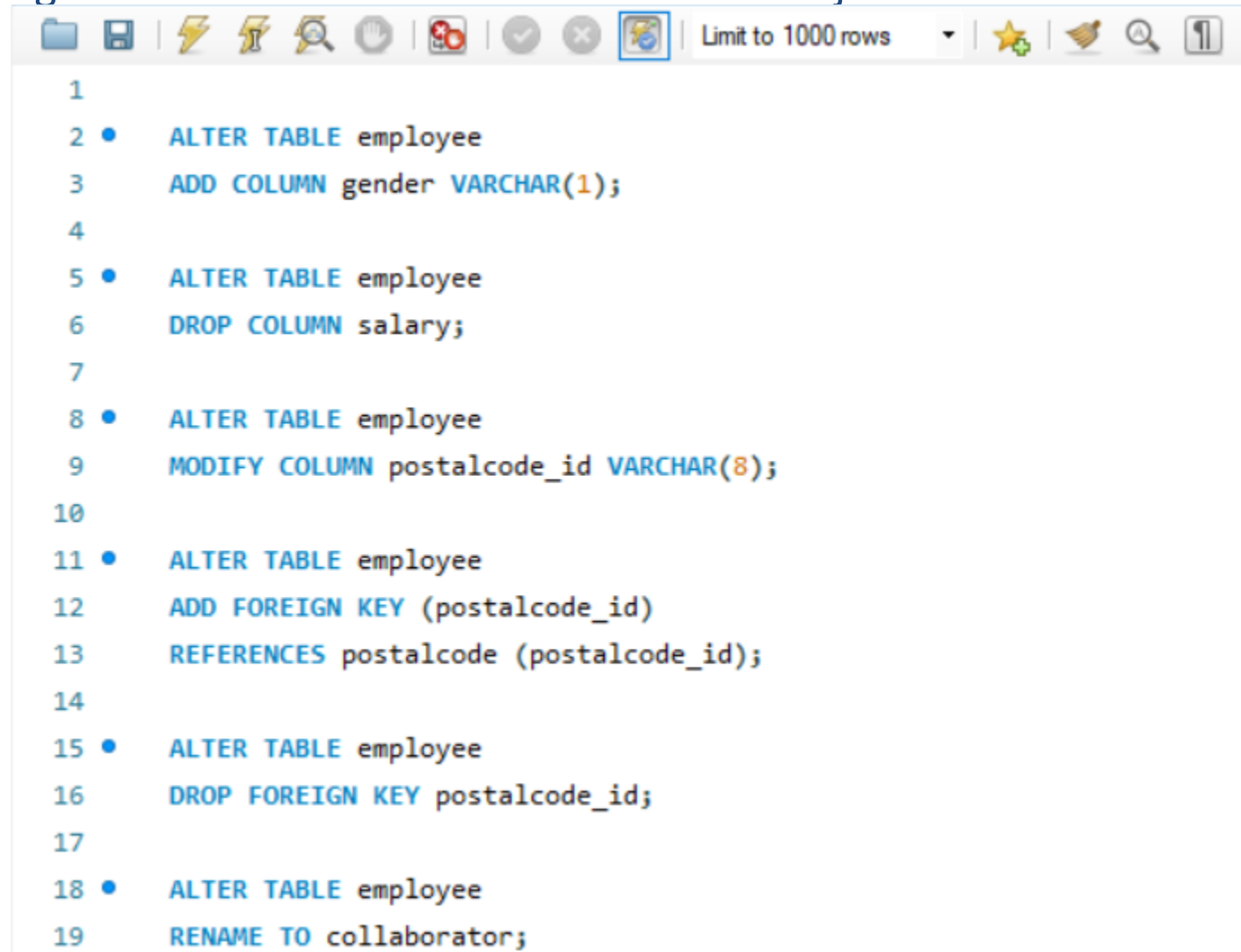
### ❖ Change of database or table structure, such as:

- ☐ Renaming the DB name or a table name
- ☐ Add, remove, or edit an attribute from a table
- ☐ Change data types or properties
- ☐ Add / Remove Indexes, References, or Foreign Keys

# INTRODUÇÃO

❖ **ALTER:** Changing the structure or characteristics of an object in the database

Some examples...



```
1
2 • ALTER TABLE employee
3   ADD COLUMN gender VARCHAR(1);
4
5 • ALTER TABLE employee
6   DROP COLUMN salary;
7
8 • ALTER TABLE employee
9   MODIFY COLUMN postalcode_id VARCHAR(8);
10
11 • ALTER TABLE employee
12   ADD FOREIGN KEY (postalcode_id)
13   REFERENCES postalcode (postalcode_id);
14
15 • ALTER TABLE employee
16   DROP FOREIGN KEY postalcode_id;
17
18 • ALTER TABLE employee
19   RENAME TO collaborator;
```



# INTRODUÇÃO

❖ **ALTER:** Changing the structure or characteristics of an object in the database

Some examples...

- `ALTER TABLE employee  
RENAME COLUMN salary TO payment;`
- `ALTER TABLE employee  
ADD COLUMN salary INT(5) AFTER birth_date;`
- `ALTER TABLE employee  
ADD COLUMN salary INT(5) DEFAULT 1200;`
- `ALTER TABLE employee  
ADD COLUMN salary INT(5) CHECK (salary BETWEEN 1000 AND 5000) ;`



the number of digits of an integer will be deprecated  
in the next versions of Mysql