POLITÉCNICO
DO PORTO
ESCOLA
SUPERIOR
DE MEDIA ARTES
E DESIGN

P.PORTO

DATABASES
SQL
Data Definition Language

TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO PARA A WEB

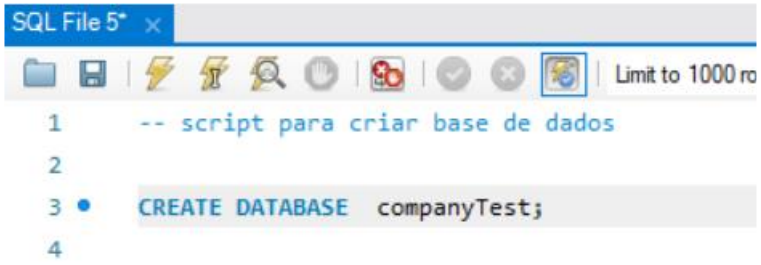
Agenda

- **SQL – Data Definition Language**
  - **CREATE**
  - **USE**
  - **SHOW**
  - **PRIMARY KEY**
  - **FOREIGN KEY**
  - **REFERENCES**
  - **DROP**
  - **ALTER**

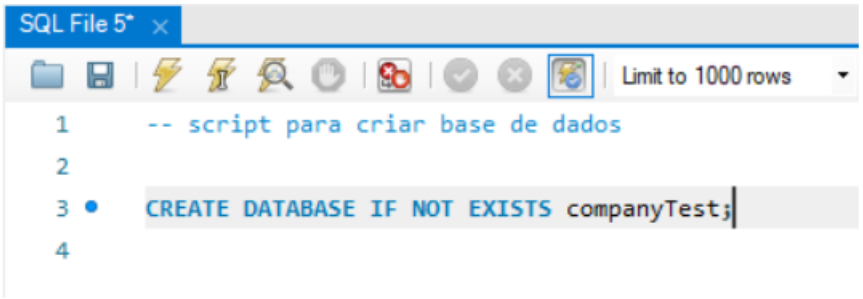
❖ SQL - Data Definition Language

❖ Create a database: CREATE

```
CREATE DATABASE [IF NOT EXISTS] database_name
```

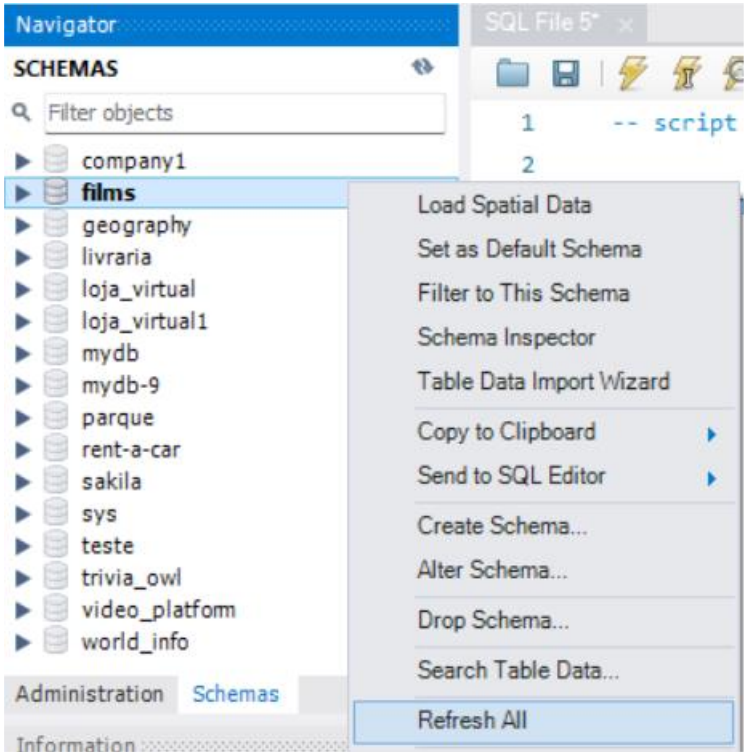


❖ If you have doubts about the existence (or not) of the DB, we can use the suffix IF NOT EXISTS



❖ SQL - Data Definition Language

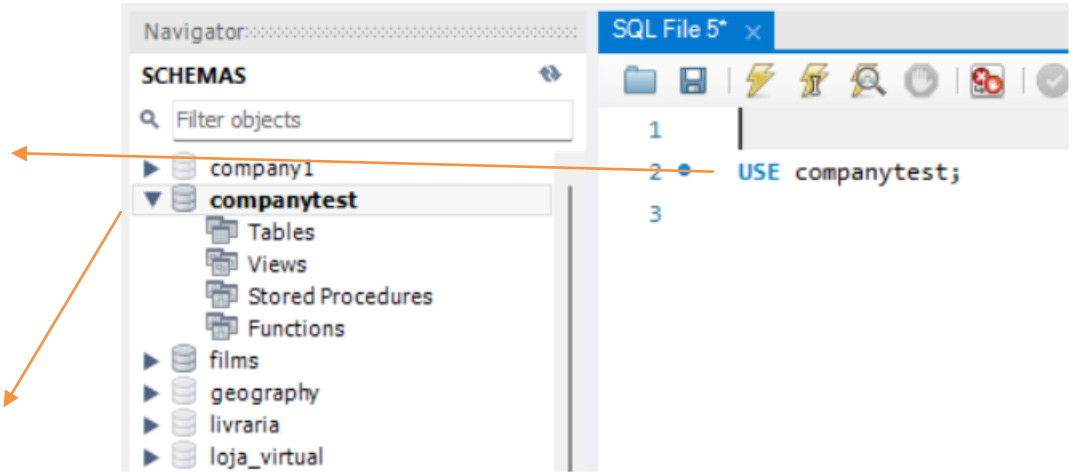
the list of schemas is not updated automatically…



Refresh the database schemas list

❖ SQL - Data Definition Language

❖ Select a database
(using sql language): USE

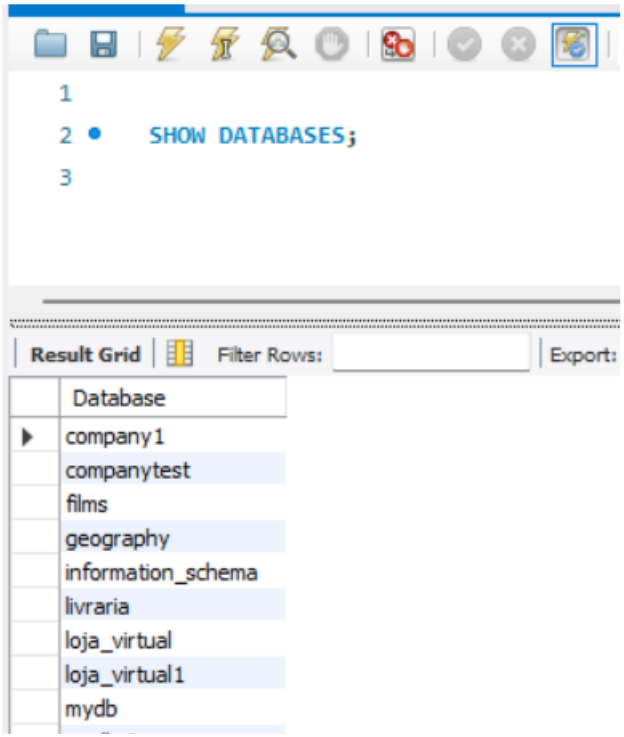


❖ Select a BD in Workbench:

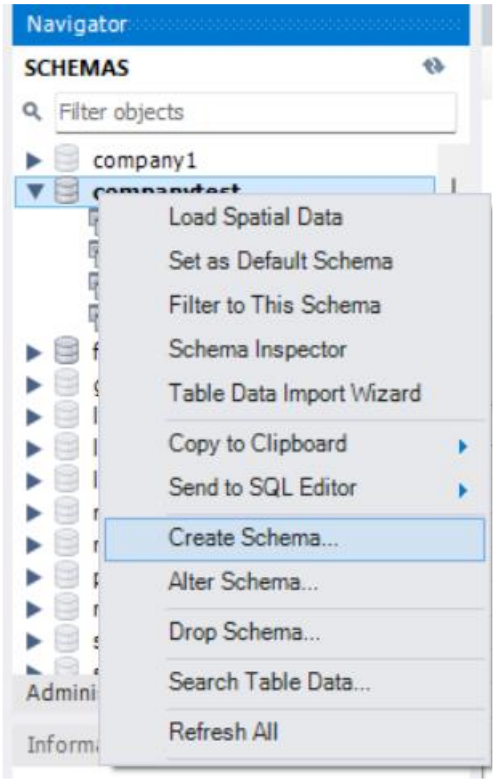
Double click in database name

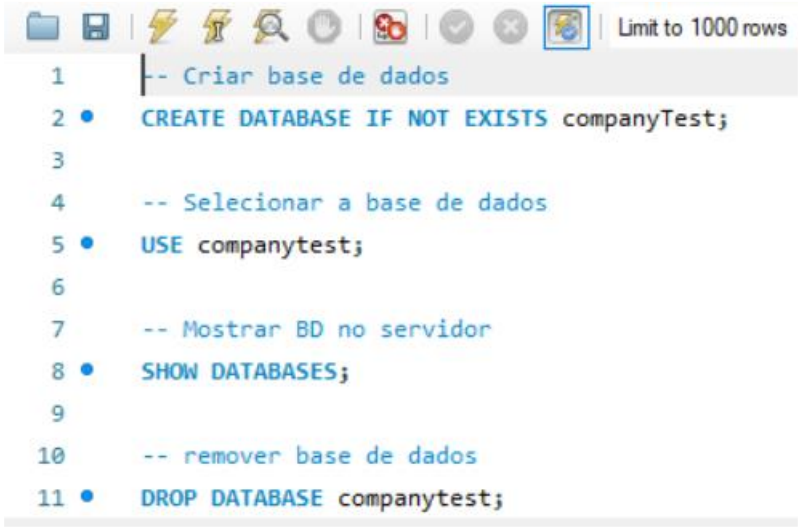
❖ SQL - Data Definition Language

Show all databases on the server: SHOW DATABASES



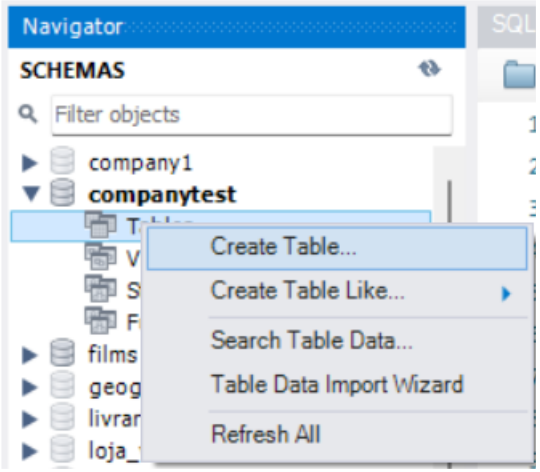
❖ SQL - Data Definition Language





❖ SQL - Data Definition Language

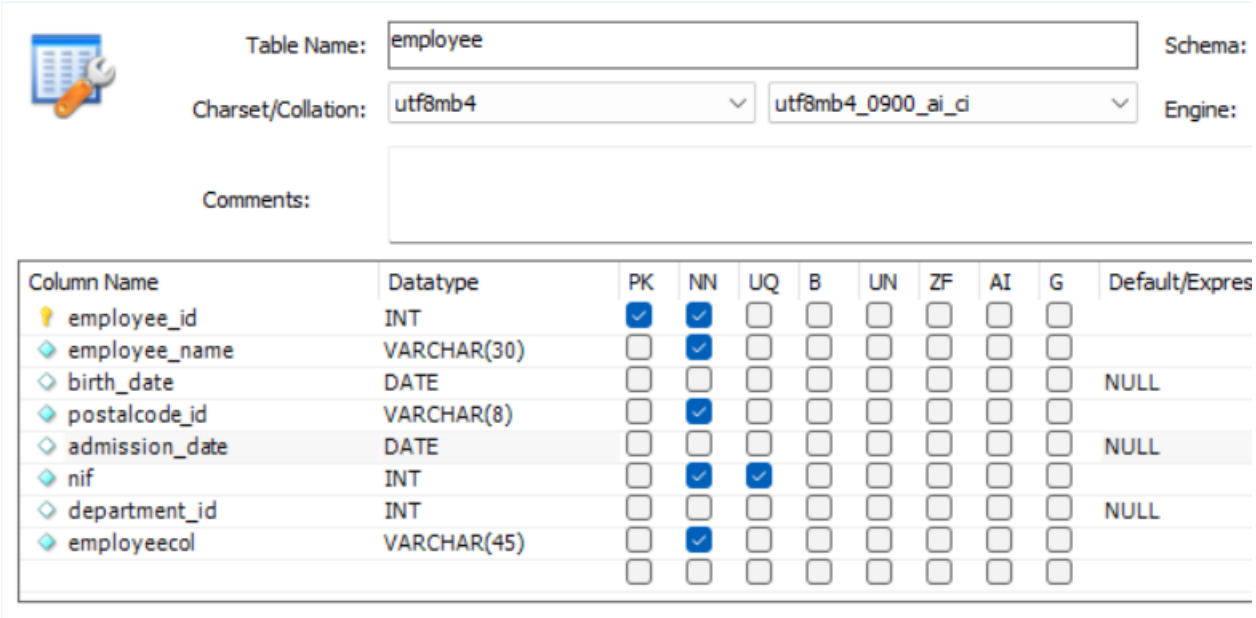
❖ Create a table: CREATE TABLE




```
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]
)
```

- SQL - Data Definition Language

- Create a table: CREATE TABLE



❖ SQL - Data Definition Language

❖ Create a table: CREATE TABLE

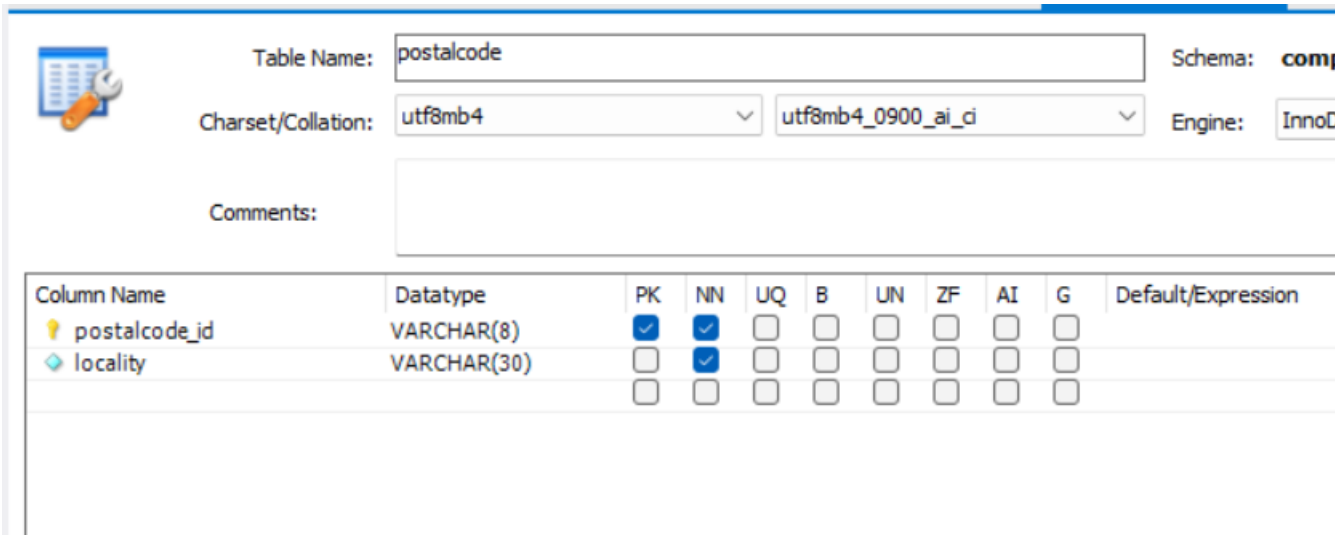
```
1   -- script criar tabela employee
2
3   CREATE TABLE IF NOT EXISTS companytest.employee (
4   employee_id     INT NOT NULL,
5   employee_name   VARCHAR(30) NOT NULL,
6   birth_date      DATE NULL,
7   postalcode_id   VARCHAR(8) NOT NULL,
8   admission_date  DATE NULL,
9   nif             INT NOT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

- SQL - Data Definition Language
- Create a table: CREATE TABLE

In SQL language:

```
-- script criar tabela postalcode
CREATE TABLE IF NOT EXISTS companytest.postalcode (
  postalcode_id VARCHAR(8) NOT NULL,
  locality      VARCHAR(30) NOT NULL,
  PRIMARY KEY (postalcode_id)
  );
```

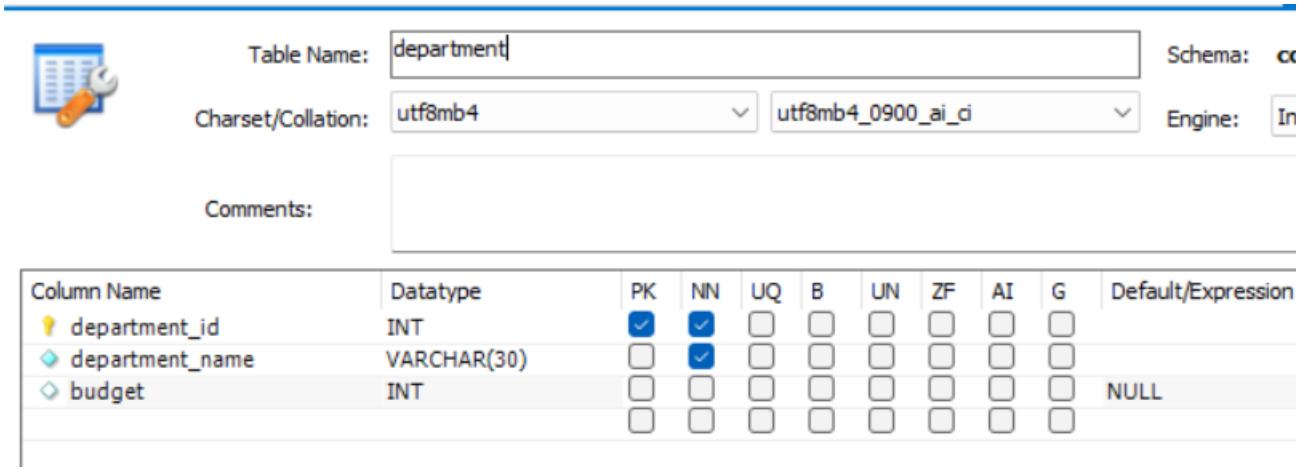
In Workbench:



- SQL Data Definition Language
- Create a Table: CREATE TABLE

PRIMARY KEY

```
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     );
```

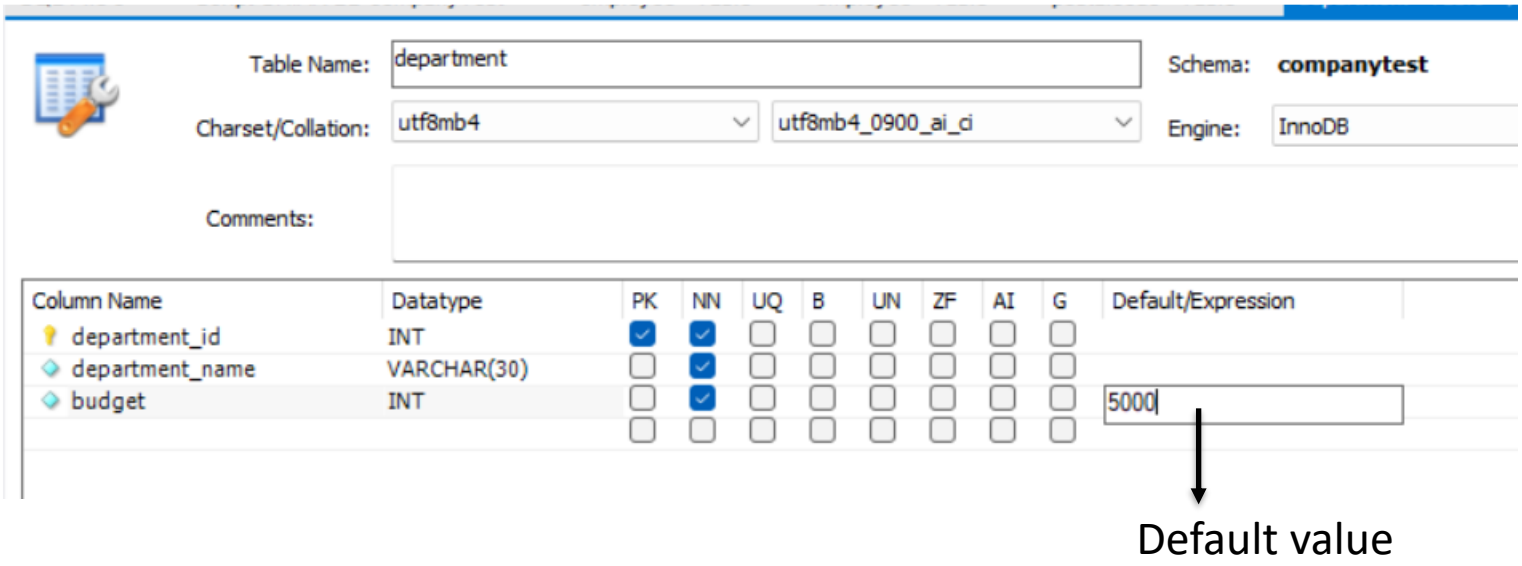


- SQL Data Definition Language
- Create a Table: DEFAULT

```
-- 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.)

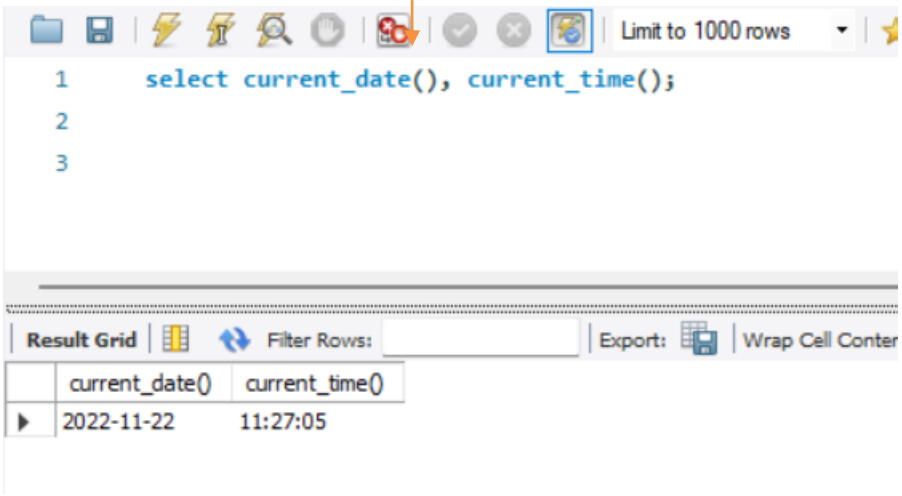


- SQL Data Definition Language
- Create a Table: DEFAULT

```
-- 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

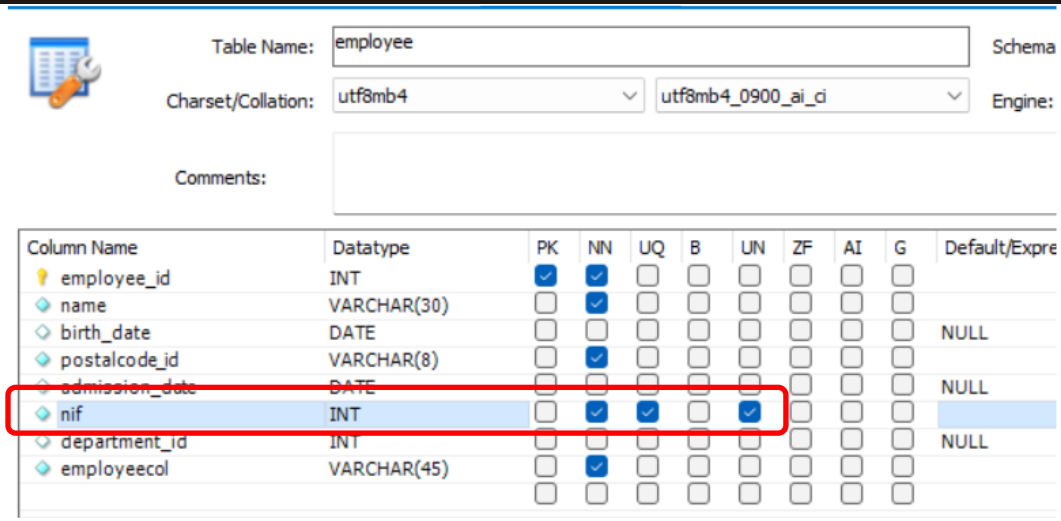
- ❑ NOT NULL is the most used constraint (already seen in previous examples)
- ❑ CHECK allows you to perform data validation on an attribute

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

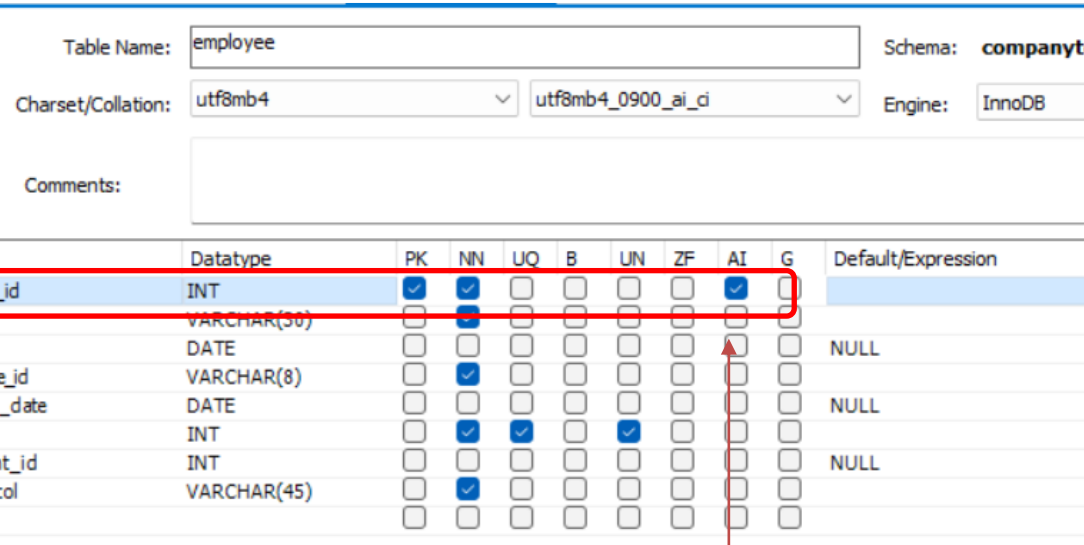
```
CREATE TABLE IF NOT EXISTS companytest.employee (
employee_id     INT NOT NULL,
employee_name   VARCHAR(30) NOT NULL,
birth_date      DATE NULL,
postalcode_id   VARCHAR(8) NOT NULL,
admission_date  DATE NULL,
nif             INT NOT NULL UNIQUE,
department_id   INT NOT NULL,
PRIMARY KEY     (employee_id)

);
```

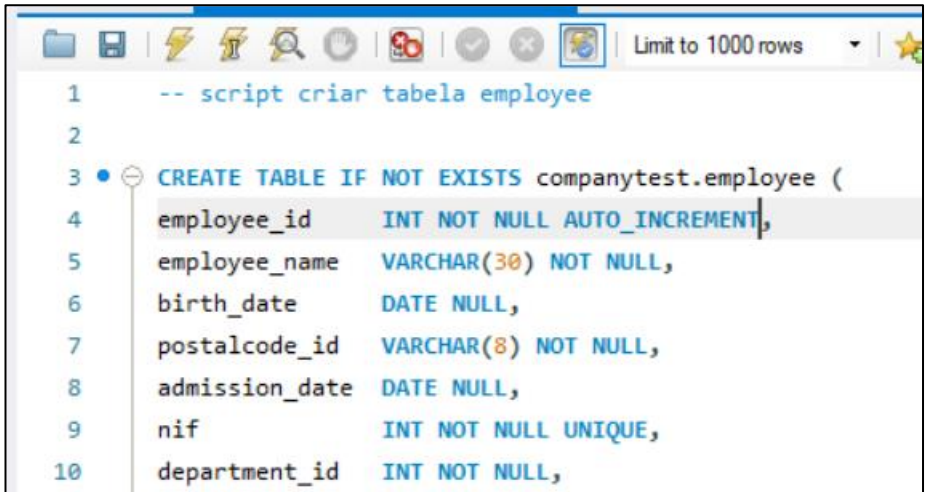
- SQL Data Definition Language

- Create a table: CREATE TABLE

CONSTRAINTS

- AUTO_INCREMENT



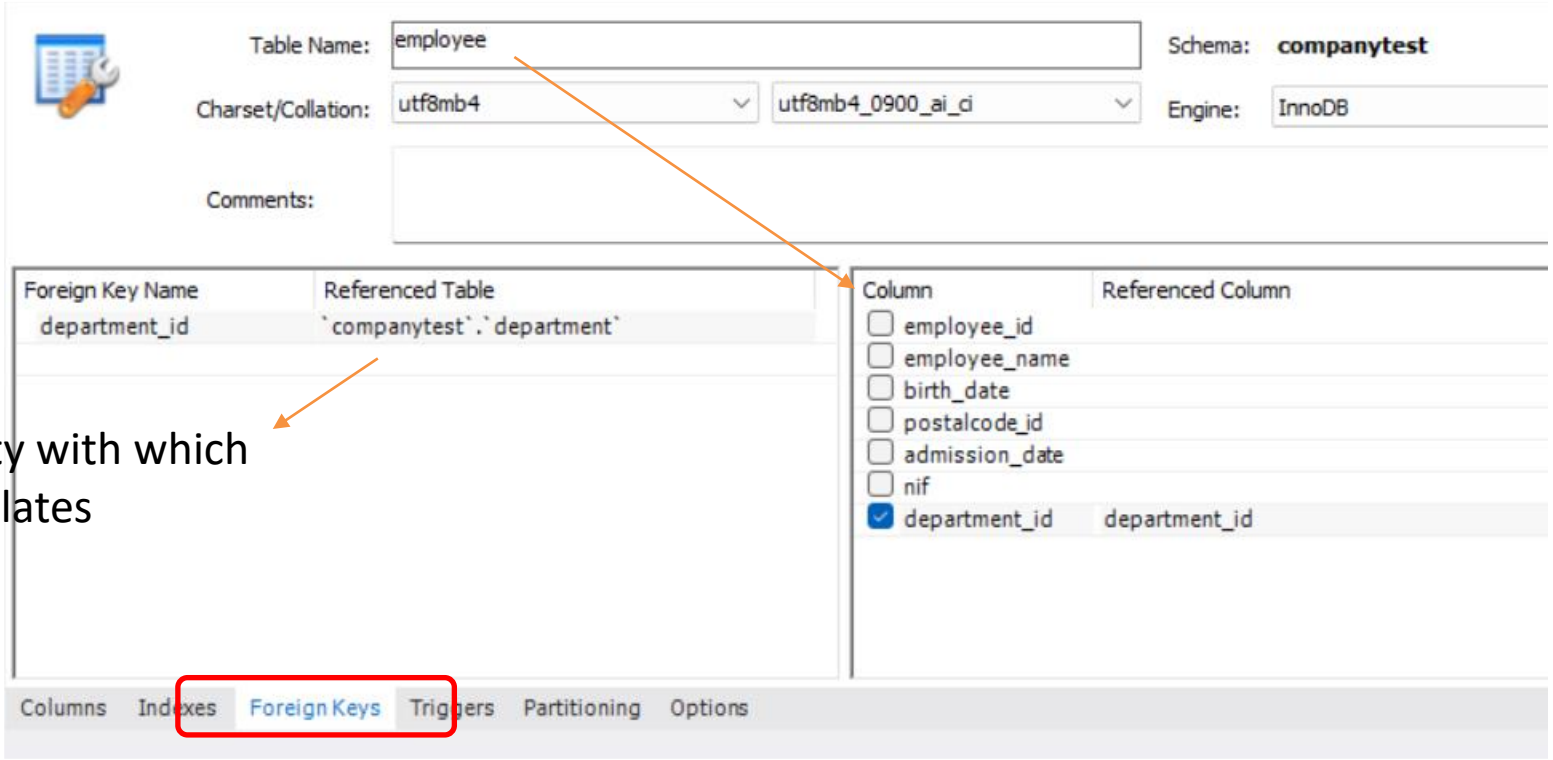


- SQL Data Definition Language

- Create a table: CREATE TABLE

CONSTRAINTS

- FOREIGN KEY



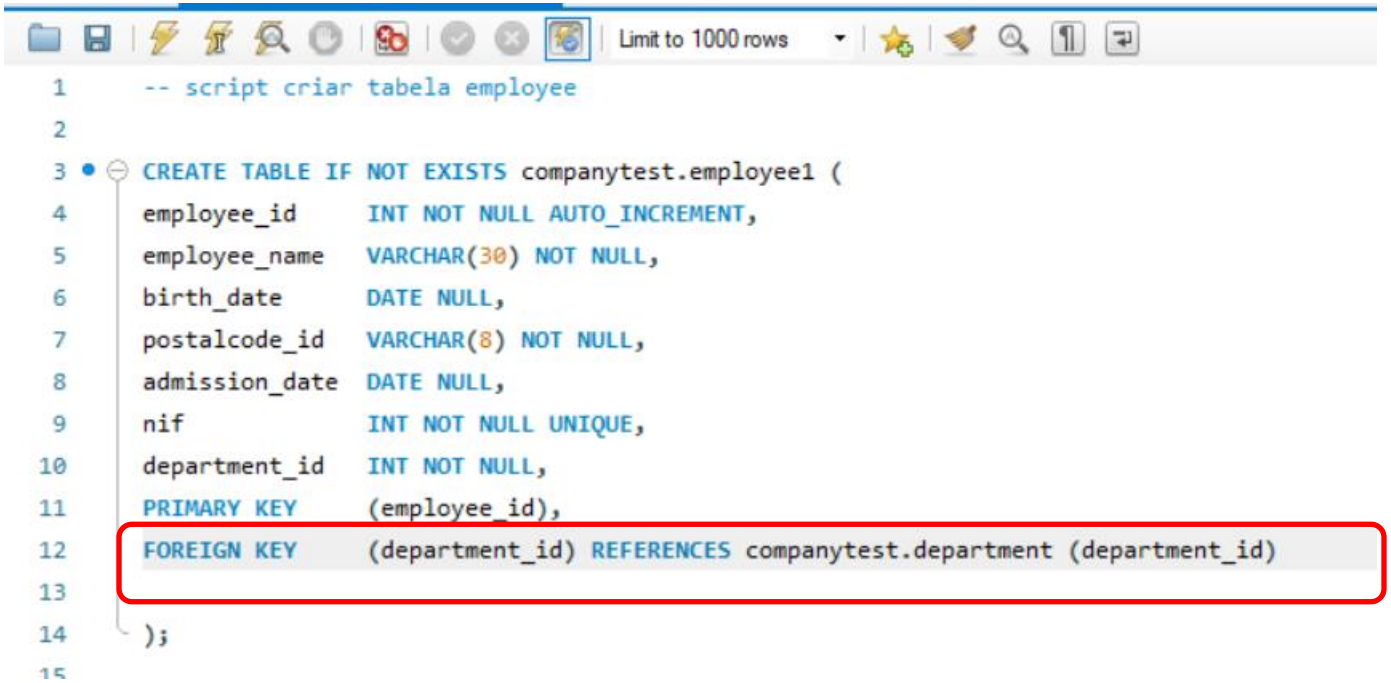
Entity with which it relates

- SQL Data Definition Language

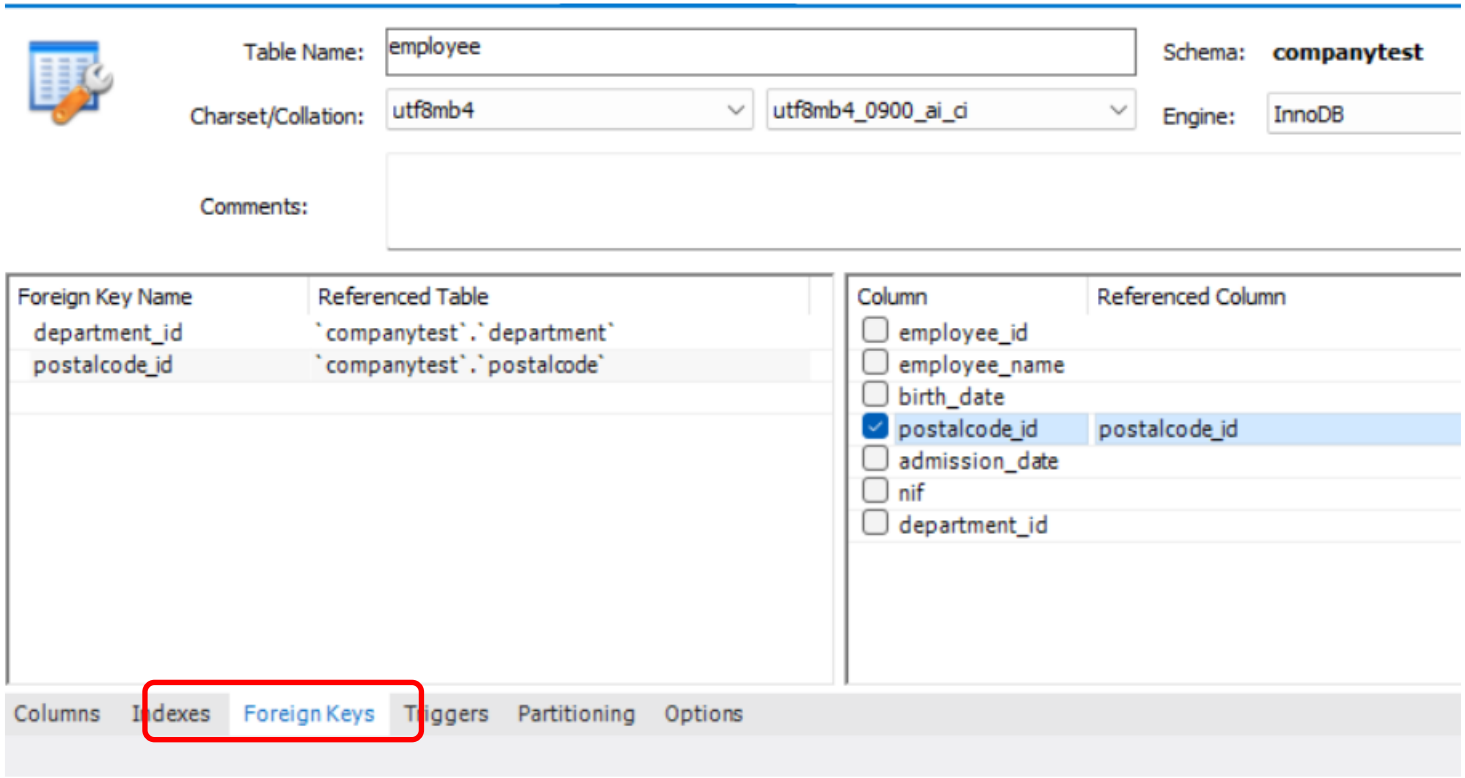
- Create a table: CREATE TABLE

CONSTRAINTS

- FOREIGN KEY



❖ SQL Data Definition Language



❖ SQL Data Definition Language

❖ Create a table: CREATE TABLE

CONSTRAINTS

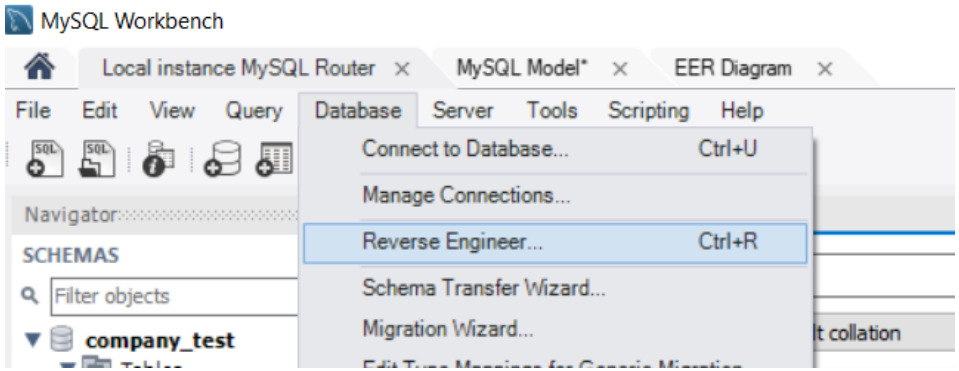
❑ FOREIGN KEY

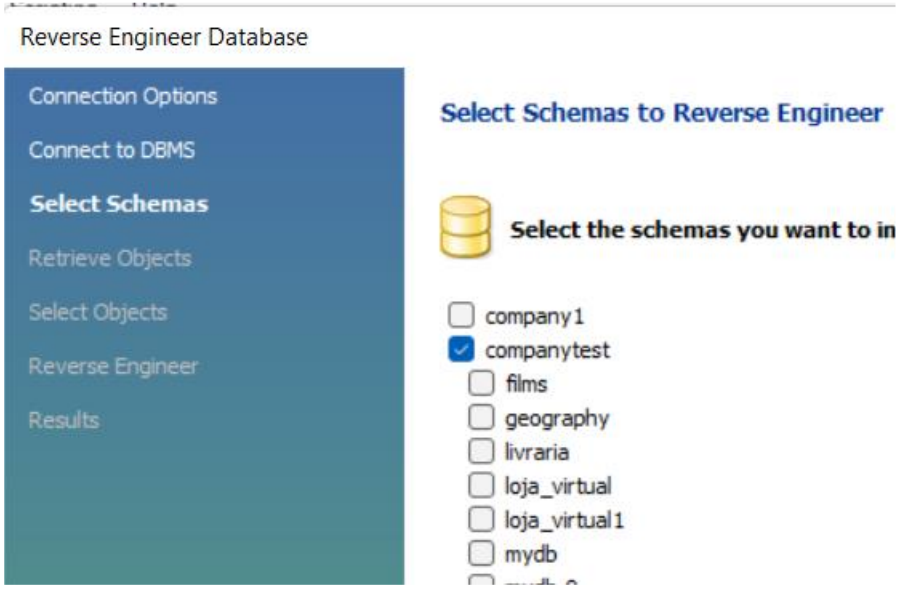
...

```
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

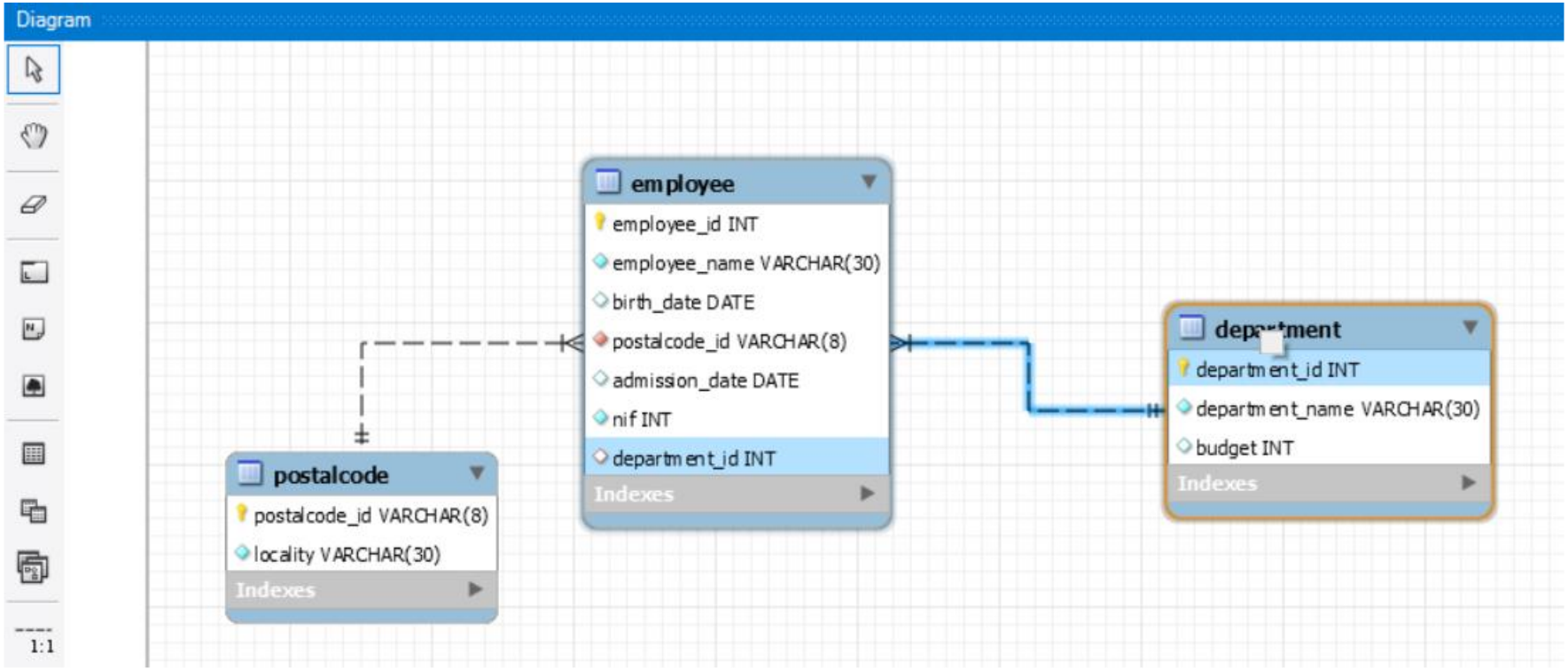
To get / verify the database schema…





❖ FOREIGN KEY

To get / verify the database schema…

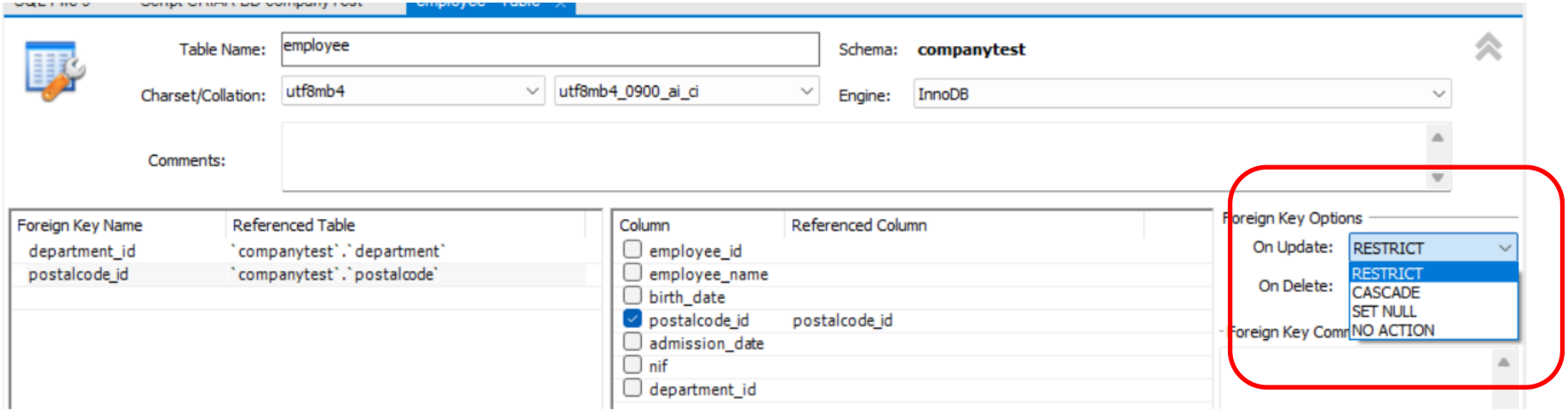


❖ FOREIGN KEY … REFERENCES

❖ 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:



❖ 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.

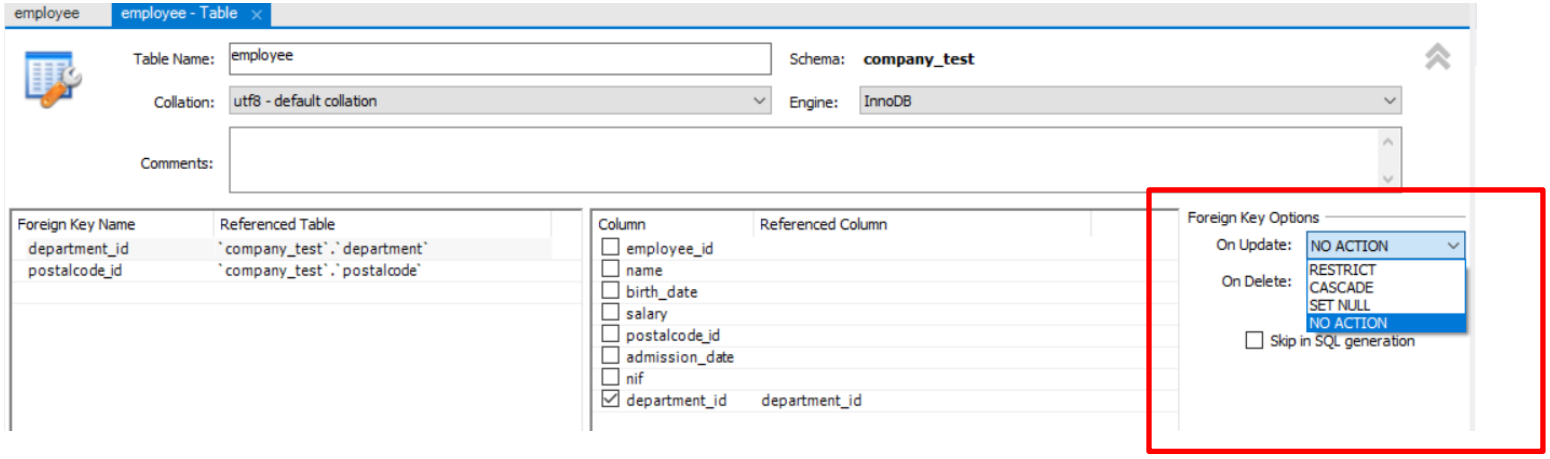
- ❖ RESTRICT (by default): Rejects the delete or update operation for the parent table. Specifying RESTRICT is the same as omitting the ON DELETE or ON UPDATE clause. Acts immediately.

❖ 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



DROP

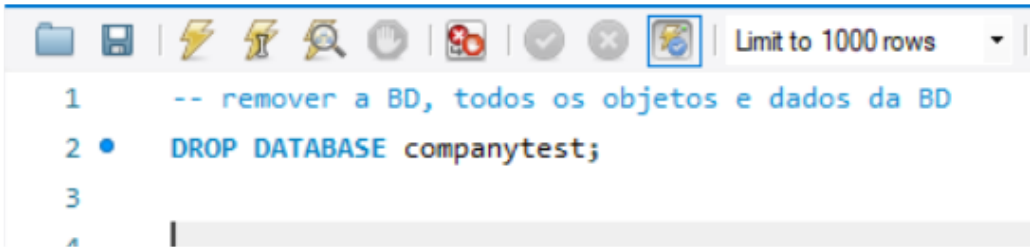
- To remove objects in a database: database, table or other objetts

- 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

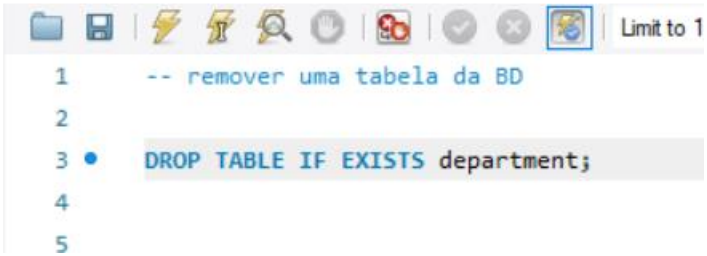
```
DROP DATABASE database_name;
```

- ❑ 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.



❖ Remove a table: DROP

DROP TABLE [IF EXISTS] *table_name;*



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

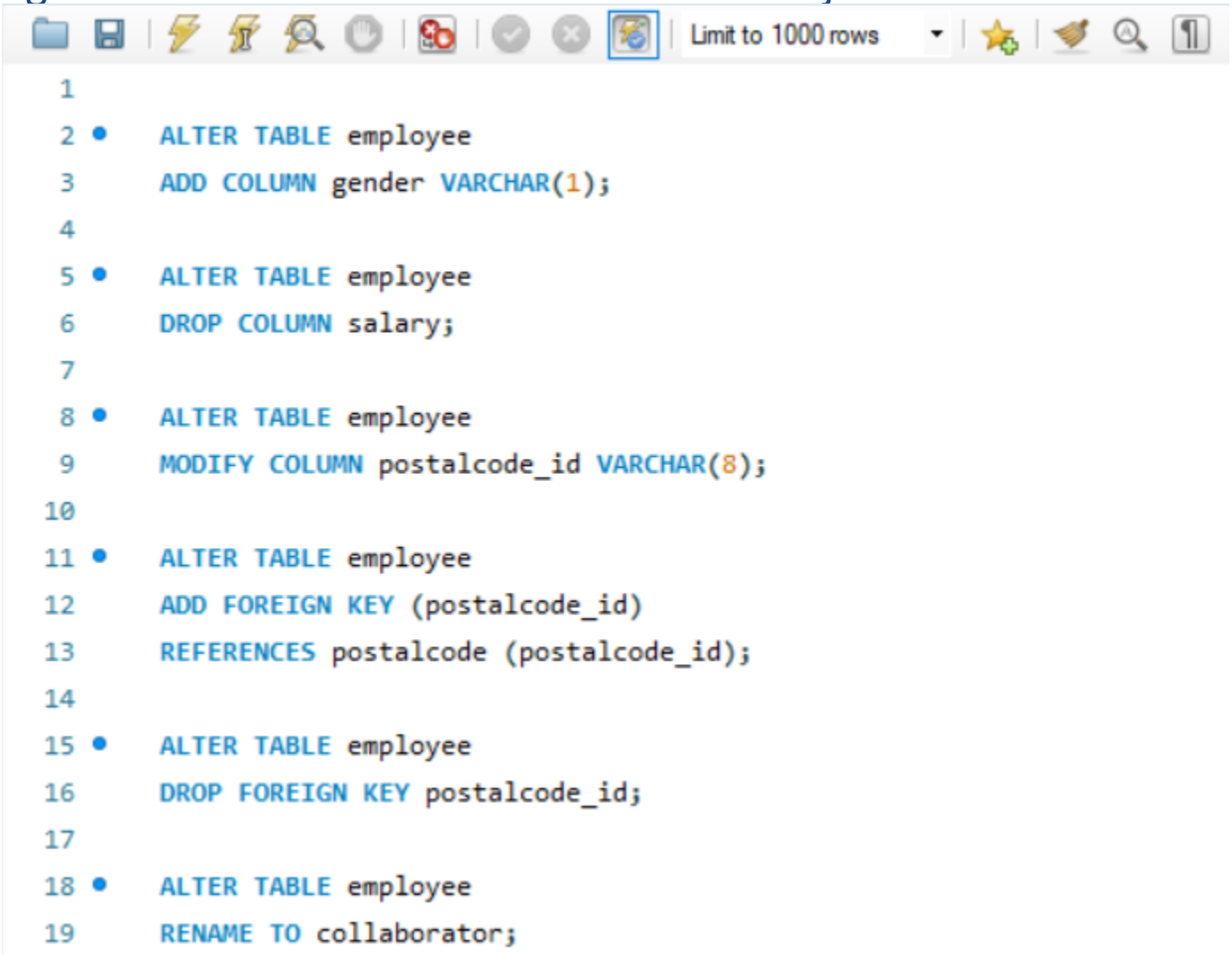
- Change of database or table structure, such as:

  - 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…



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