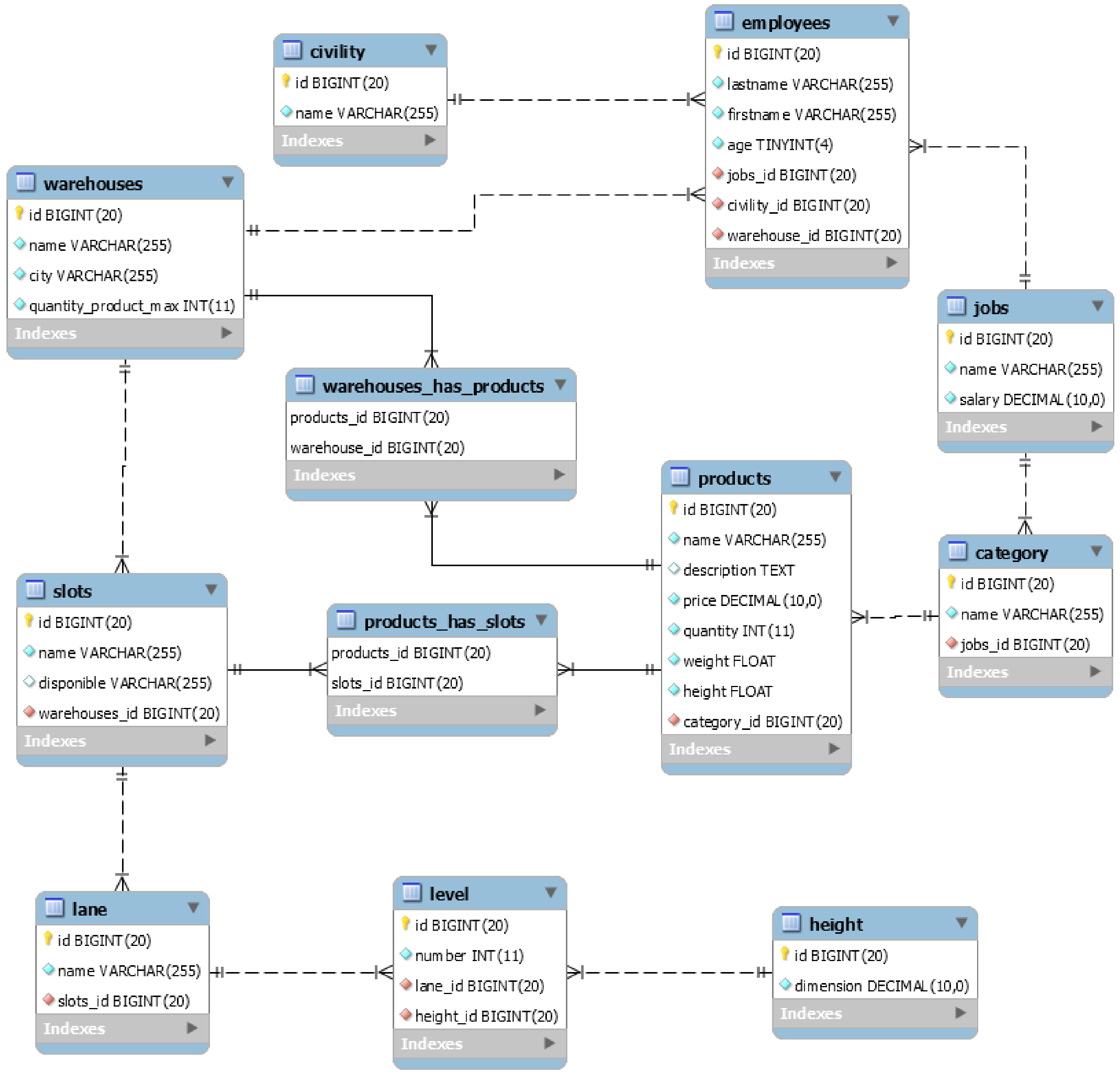
BASE DE DONNÉES

Workbench



civility
- id BIGINT(20)
- name VARCHAR(255)
- Indexes

employees
- id BIGINT(20)
- lastname VARCHAR(255)
- firstname VARCHAR(255)
- age TINYINT(4)
- jobs_id BIGINT(20)
- civility_id BIGINT(20)
- warehouse_id BIGINT(20)
- Indexes

warehouses
- id BIGINT(20)
- name VARCHAR(255)
- city VARCHAR(255)
- quantity_product_max INT(11)
- Indexes

jobs
- id BIGINT(20)
- name VARCHAR(255)
- salary DECIMAL(10,0)
- Indexes

warehouses_has_products
- products_id BIGINT(20)
- warehouse_id BIGINT(20)
- Indexes

products
- id BIGINT(20)
- name VARCHAR(255)
- description TEXT
- price DECIMAL(10,0)
- quantity INT(11)
- weight FLOAT
- height FLOAT
- category_id BIGINT(20)
- Indexes

category
- id BIGINT(20)
- name VARCHAR(255)
- jobs_id BIGINT(20)
- Indexes

slots
- id BIGINT(20)
- name VARCHAR(255)
- disponible VARCHAR(255)
- warehouses_id BIGINT(20)
- Indexes

products_has_slots
- products_id BIGINT(20)
- slots_id BIGINT(20)
- Indexes

lane
- id BIGINT(20)
- name VARCHAR(255)
- slots_id BIGINT(20)
- Indexes

level
- id BIGINT(20)
- number INT(11)
- lane_id BIGINT(20)
- height_id BIGINT(20)
- Indexes

height
- id BIGINT(20)
- dimension DECIMAL(10,0)
- Indexes

Scripts SQL

Commandes

Types

INDEX

CONSTRAINT

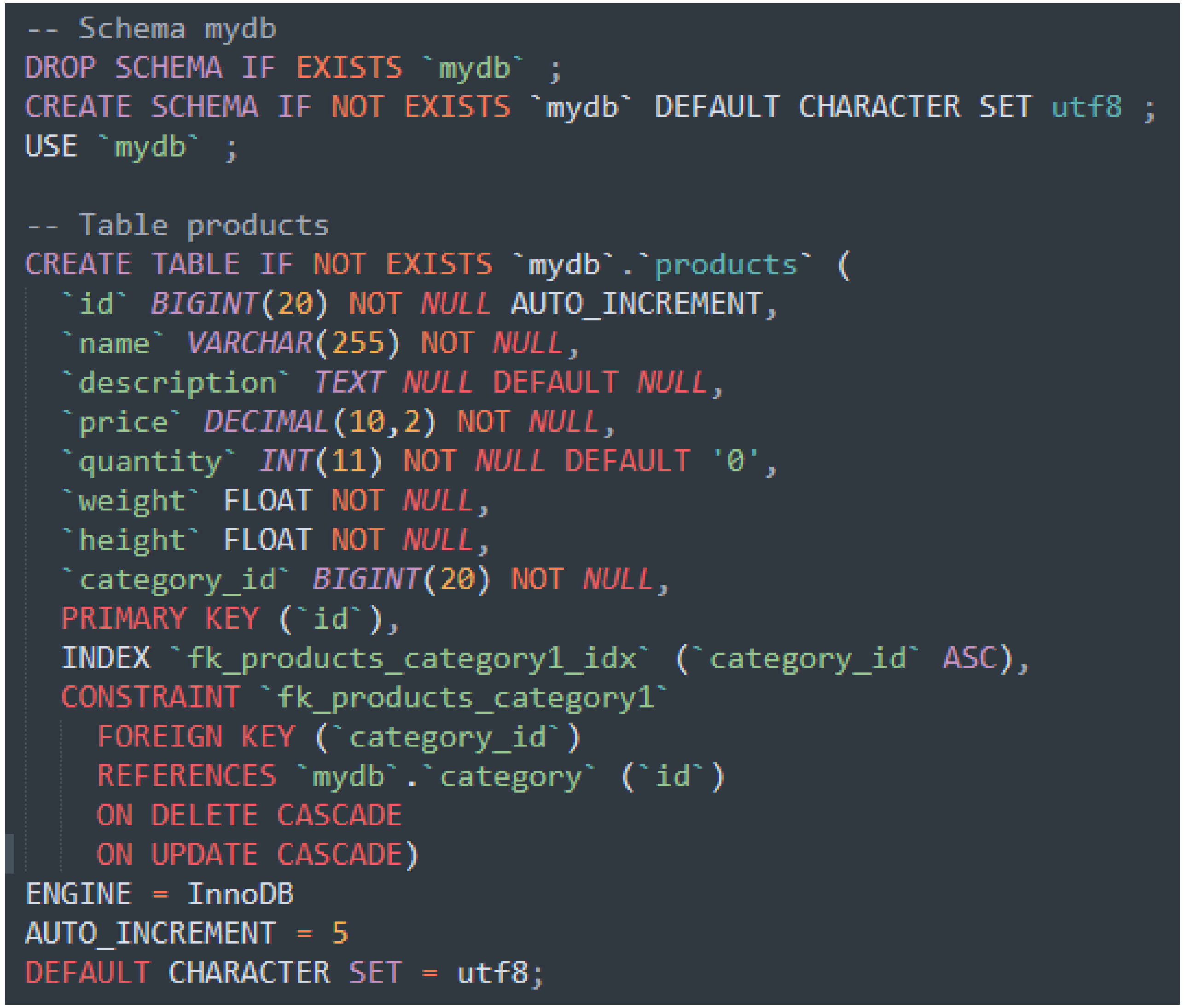
REFERENCES

ON DELETE

ON UPDATE

ENGINE

AUTO_INCREMENT



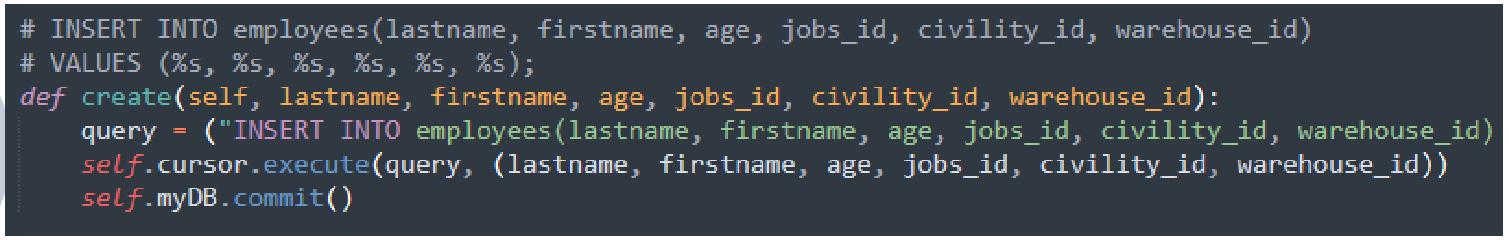
```sql
-- Schema mydb
DROP SCHEMA IF EXISTS `mydb` ;
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-- Table products
CREATE TABLE IF NOT EXISTS `mydb`.`products` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `description` TEXT NULL DEFAULT NULL,
  `price` DECIMAL(10,2) NOT NULL,
  `quantity` INT(11) NOT NULL DEFAULT '0',
  `weight` FLOAT NOT NULL,
  `height` FLOAT NOT NULL,
  `category_id` BIGINT(20) NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_products_category1_idx` (`category_id` ASC),
  CONSTRAINT `fk_products_category1`
    FOREIGN KEY (`category_id`)
    REFERENCES `mydb`.`category` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 5
DEFAULT CHARACTER SET = utf8;
```

Code

CRUD

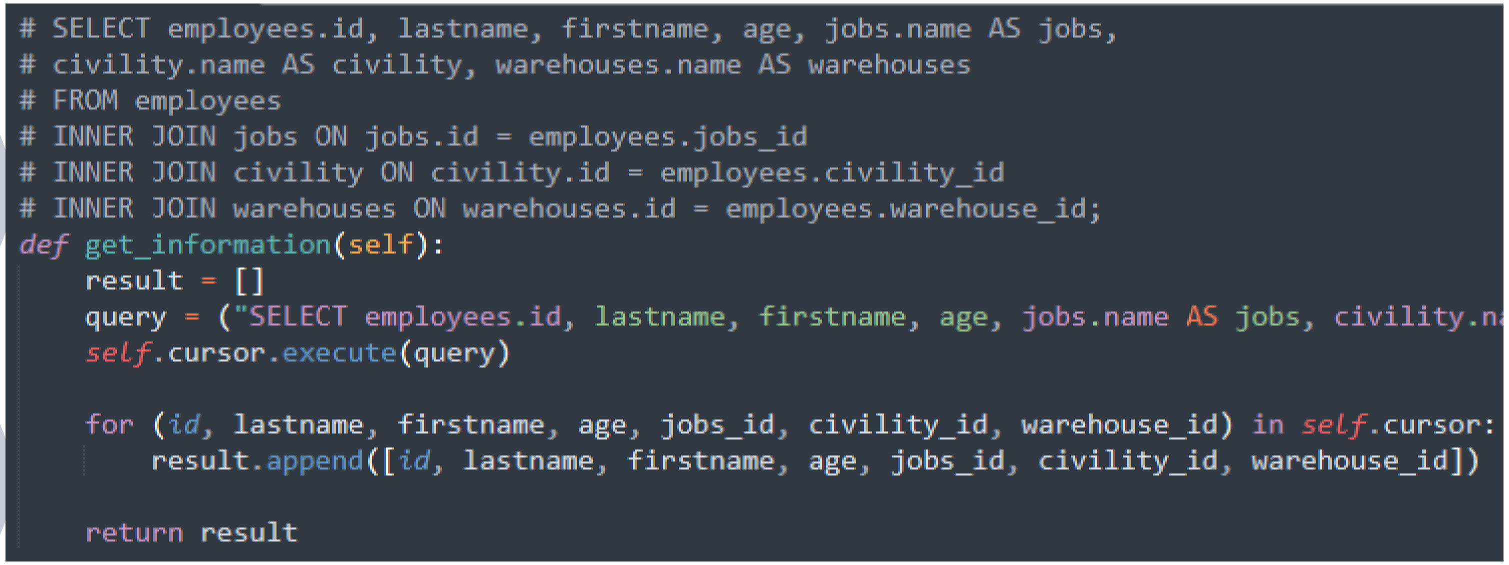
Create Read Update Delete

CREATE



```python
# INSERT INTO employees(lastname, firstname, age, jobs_id, civility_id, warehouse_id)
# VALUES (%s, %s, %s, %s, %s, %s);
def create(self, lastname, firstname, age, jobs_id, civility_id, warehouse_id):
    query = ("INSERT INTO employees(lastname, firstname, age, jobs_id, civility_id, warehouse_id)
    self.cursor.execute(query, (lastname, firstname, age, jobs_id, civility_id, warehouse_id))
    self.myDB.commit()
```

READ



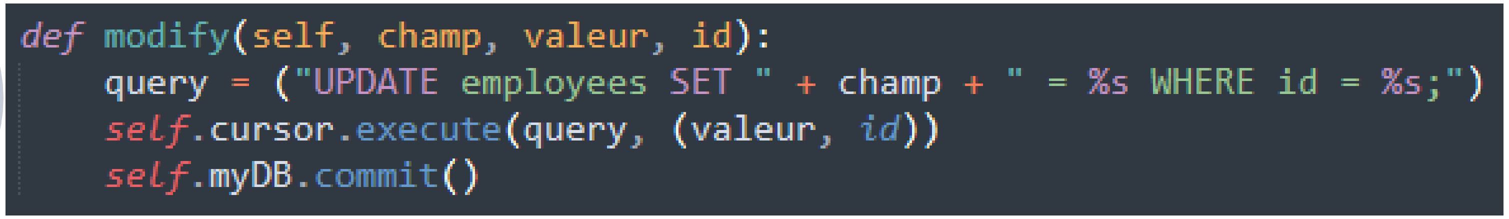
```python
# SELECT employees.id, lastname, firstname, age, jobs.name AS jobs,
# civility.name AS civility, warehouses.name AS warehouses
# FROM employees
# INNER JOIN jobs ON jobs.id = employees.jobs_id
# INNER JOIN civility ON civility.id = employees.civility_id
# INNER JOIN warehouses ON warehouses.id = employees.warehouse_id;
def get_information(self):
    result = []
    query = ("SELECT employees.id, lastname, firstname, age, jobs.name AS jobs, civility.na
    self.cursor.execute(query)

    for (id, lastname, firstname, age, jobs_id, civility_id, warehouse_id) in self.cursor:
        result.append([id, lastname, firstname, age, jobs_id, civility_id, warehouse_id])

    return result
```

Code

CRUD

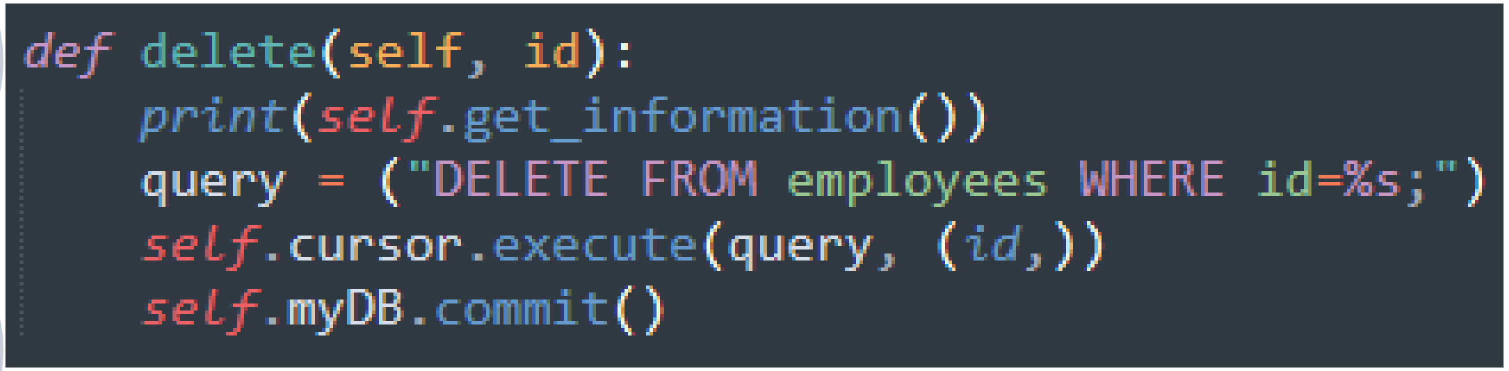
Create Read Update Delete

UPDATE



```python
def modify(self, champ, valeur, id):
    query = ("UPDATE employees SET " + champ + " = %s WHERE id = %s;")
    self.cursor.execute(query, (valeur, id))
    self.myDB.commit()
```

DELETE

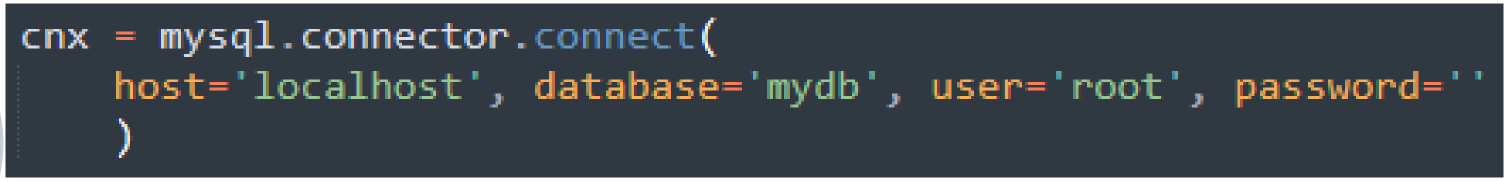


```python
def delete(self, id):
    print(self.get_information())
    query = ("DELETE FROM employees WHERE id=%s;")
    self.cursor.execute(query, (id,))
    self.myDB.commit()
```

Code

Module

mysql.connector



```
cnx = mysql.connector.connect(
    host='localhost', database='mydb', user='root', password=''
    )
```