

## EXEMPLO DE ESQUEMA DE BANCO DE DADOS

```
CREATE TABLE IF NOT EXISTS `mydb`.`Cliente` (  
  `cod_cli` INT NOT NULL,  
  `nome_cli` VARCHAR(45) NOT NULL,  
  `endereco` VARCHAR(45) NULL DEFAULT 'null',  
  `cidade` VARCHAR(45) NULL DEFAULT 'null',  
  `cep` CHAR(8) NULL DEFAULT 'null',  
  `uf` CHAR(4) NULL DEFAULT 'null',  
  PRIMARY KEY (`cod_cli`))  
ENGINE = InnoDB;
```

-----

-- Table `mydb`.`Vendedor`

-----

```
CREATE TABLE IF NOT EXISTS `mydb`.`Vendedor` (  
  `cod_vend` INT NOT NULL,  
  `nome_vend` VARCHAR(45) NOT NULL,  
  `sal_fixo` DECIMAL(7,2) NOT NULL,  
  `faixa_comiss` CHAR(1) NOT NULL,  
  PRIMARY KEY (`cod_vend`))  
ENGINE = InnoDB;
```

-----

-- Table `mydb`.`Produto`

-----

```
CREATE TABLE IF NOT EXISTS `mydb`.`Produto` (  
  `cod_prod` INT NOT NULL,  
  `unid_prod` VARCHAR(5) NOT NULL,  
  `desc_prod` VARCHAR(45) NOT NULL,  
  `val_unit` DECIMAL(5,2) NOT NULL,  
  PRIMARY KEY (`cod_prod`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `mydb`.`Pedido`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Pedido` (  
  `num_ped` INT NOT NULL,  
  `prazo_ent` INT NOT NULL,  
  `Cliente_cod_cli` INT NOT NULL,  
  `Vendedor_cod_vend` INT NOT NULL,  
  PRIMARY KEY (`num_ped`),  
  INDEX `fk_Pedido_Cliente1_idx` (`Cliente_cod_cli` ASC),  
  INDEX `fk_Pedido_Vendedor1_idx` (`Vendedor_cod_vend` ASC),  
  CONSTRAINT `fk_Pedido_Cliente1`  
    FOREIGN KEY (`Cliente_cod_cli`)  
      REFERENCES `mydb`.`Cliente` (`cod_cli`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Pedido_Vendedor1`  
    FOREIGN KEY (`Vendedor_cod_vend`)  
      REFERENCES `mydb`.`Vendedor` (`cod_vend`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `mydb`.`Pedido_has_Produto`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Pedido_has_Produto` (  
  `Pedido_num_ped` INT NOT NULL,  
  `Produto_cod_prod` INT NOT NULL,  
  `qtd_ped` INT NOT NULL,  
  PRIMARY KEY (`Pedido_num_ped`, `Produto_cod_prod`),  
  INDEX `fk_Pedido_has_Produto_Produto1_idx` (`Produto_cod_prod` ASC),
```

```

INDEX `fk_Pedido_has_Produto_Pedido1_idx` (`Pedido_num_ped` ASC),
CONSTRAINT `fk_Pedido_has_Produto_Pedido1`
  FOREIGN KEY (`Pedido_num_ped`)
  REFERENCES `mydb`.`Pedido` (`num_ped`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Pedido_has_Produto_Produto1`
  FOREIGN KEY (`Produto_cod_prod`)
  REFERENCES `mydb`.`Produto` (`cod_prod`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

## EXERCÍCIOS

Escreva as instruções DDL da SQL para declarar o esquema de banco de dados relacional a seguir.

```

empregado (id,cpf,nome,endereco,sexo,salario,supervisor_id,departamento_id)
departamento (id, nome, gerente_id)
departamento_localizacao (departamento_id, localizacao)
projeto (id, nome, localizacao, departamento_id)
alocacao (projeto_id, empregado_id, numHoras)
empregado_dependente (empregado_id, nome, sexo, data_nascimento, parentesco)

```

### REGRAS DE NEGÓCIO:

1. O empregado pode ficar eventualmente sem supervisor ou não estar lotado em nenhum departamento.
2. O CPF do empregado deve ser único.
3. O departamento pode ficar eventualmente sem gerente.
4. A localização só deve existir se estiver associada a algum departamento, por isso, caso um departamento seja apagado, suas localizações também devem ser apagadas.
5. O projeto sempre precisa está associado a um departamento, por isso, um departamento só pode ser apagado se não tiver nenhum projeto associado a ele.
6. Quando um empregado é apagado, suas alocações em projetos também devem ser apagadas.

7. Um projeto só pode ser apagado quando não houver mais empregados alocados nele.
8. Quando um empregado é apagado, seus dependentes também devem ser apagados.