

# Projeto Físico de Banco de Dados

## Atividade Prática

Desenvolver, a partir de seus projetos lógicos, os projetos físicos correspondentes, conforme exemplo e instruções abaixo. Use um editor de textos e teste os seus projetos em um software gerenciador de banco de dados.

## Exemplo:

### Projeto Lógico

cargo (codigoCargo,  
descricao,  
idade)

empregado (matricula,  
nome,  
idade,  
codigoCargo)  
codigoCargo referencia codigoCargo em cargo

projeto (codigoProjeto,  
descricao,  
orcamento)

projetoEmpregado (  
codigoProjeto,  
matricula,  
dtInicio)  
codigoProjeto referencia codigoProjeto em projeto  
matricula referencia matricula em empregado  
)

## Projeto Físico

### Roteiro para o mapeamento a partir do projeto lógico

Para criação do projeto físico, ACRESCENTE ao projeto lógico, conforme o exemplo, o seguinte:

1. A instrução CREATE TABLE antes do nome da tabela
2. O tipo de dado após o nome de cada campo (VARCHAR(tamanho) para cadeias de caracteres, INTEGER para números inteiros, DOUBLE para números reais e DATE para datas)
3. A instrução PRIMARY KEY (nome da chave primária) depois da lista de campos
4. A instrução FOREIGN KEY (nome da chave estrangeira) REFERENCES nome da tabela referenciada (nome da chave primária da tabela referenciada)
5. Observe abaixo que as definições de campos e de chaves primárias e estrangeiras são todas separadas por vírgulas e envoltas por um conjunto de parênteses que se inicia após o nome da tabela e termina após toda a instrução de criação da tabela. No projeto físico o rigor da sintaxe deve ser observado, visto que são instruções que serão executadas pelo SGBD.

```
CREATE DATABASE testeApagar;  
USE testeApagar;
```

```
CREATE TABLE cargo (  
  codigoCargo INTEGER,  
  descricao VARCHAR(50),  
  idade DOUBLE,  
  PRIMARY KEY (codigoCargo)  
) ENGINE = InnoDB;
```

```
CREATE TABLE empregado (  
  matricula INTEGER,  
  nome VARCHAR(45),  
  idade INTEGER,  
  codigoCargo INTEGER,  
  PRIMARY KEY (matricula),  
  FOREIGN KEY (codigoCargo)  
    REFERENCES cargo (codigoCargo)  
) ENGINE = InnoDB;
```

```
CREATE TABLE projeto (  
  codigoProjeto INTEGER,  
  descricao VARCHAR(45),  
  orcamento DOUBLE,  
  PRIMARY KEY (codigoProjeto)  
) ENGINE = InnoDB;
```

```
CREATE TABLE projetoEmpregado (  
  codigoProjeto INTEGER,  
  matricula INTEGER,  
  dtInicio DATE,  
  PRIMARY KEY (codigoProjeto, matricula),  
  FOREIGN KEY (codigoProjeto) REFERENCES projeto (codigoProjeto),  
  FOREIGN KEY (matricula) REFERENCES empregado (matricula)  
) ENGINE = InnoDB;
```

# Roteiro para Implementação e Teste do Projeto Físico

## No MySQL Query Browser:

1. Crie um novo schema: `CREATE DATABASE ibd`
2. Torne o schema criado o default: `USE ibd`
3. Crie uma aba de script (script tab) e execute cada um dos comandos de seu projeto lógico. Observe a ordem de dependência quando referenciar outra tabela
4. Se desejar, monte um arquivo de script com vários comandos e carregue-o por meio do item de menu open script

## No prompt de comando:

1. Abra o prompt de comando e vá até o diretório bin do mysql
2. `Mysql --user=root --port=3308`
3. `CREATE DATABASE ibd`
4. `USE ibd`
5. Insira os comandos `CREATE TABLE` gerados no projeto físico
6. Use a instrução `SHOW TABLES` para verificar as tabelas criadas
7. Use a instrução `SHOW COLUMNS FROM tabela;` para visualizar a estrutura da tabela criada
8. Use a instrução `SHOW INDEX FROM tabela` para visualizar os índices criados (para as `PRIMARY KEY` e `FOREIGN KEY`)
9. Para executar um arquivo gerado e um editor de textos, use a instrução `SOURCE`. Ao colocar várias instruções em um a arquivo, separe-as com ponto-e-vírgula

## Esquema geral resumido da instrução CREATE TABLE

```
CREATE TABLE [IF NOT EXISTS] tbl_name
  (create_definition,...)
  [table_option ...]
```

```
create_definition:
  column_definition
| PRIMARY KEY (index_col_name,...)
| FOREIGN KEY (index_col_name,...) [reference_definition]
```

```
column_definition:
  col_name data_type [NOT NULL | NULL] [DEFAULT default_value]
  [AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
  [COMMENT 'string'] [reference_definition]
```

```
data_type:
  INTEGER[(length)] [UNSIGNED] [ZEROFILL]
| DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DATE
| VARCHAR(length)
```

```
reference_definition:
  REFERENCES tbl_name [(index_col_name,...)]
  [MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]
  [ON DELETE reference_option]
  [ON UPDATE reference_option]
```

```
reference_option:
  RESTRICT | CASCADE | SET NULL | NO ACTION
```

```
table_option:
  {ENGINE|TYPE} [=] engine_name
| AUTO_INCREMENT [=] value
```

## Esquema geral resumido da instrução ALTER TABLE

```
ALTER TABLE tbl_name
  alter_specification [, alter_specification] ...
```

```
alter_specification:
  table_option ...
| ADD [COLUMN] column_definition [FIRST | AFTER col_name ]
| ADD [COLUMN] (column_definition,...)
| ADD PRIMARY KEY (index_col_name,...)
| ADD UNIQUE (index_col_name,...)
| ADD FOREIGN KEY (index_col_name,...) [reference_definition]
| CHANGE [COLUMN] old_col_name column_definition [FIRST | AFTER col_name]
| MODIFY [COLUMN] column_definition [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP FOREIGN KEY fk_symbol
| RENAME [TO] new_tbl_name
```