

## Avaliação \_08 - Banco de Dados

Dupla: Isabelly Barbosa Cordeiro e Thayná Nascimento Albano da Silva

Curso: Informática - P4

Professor: Ricardo Duarte Taveira

### ❖ Proposta de avaliação:

Criar o esquema (código SQL) que cria as tabelas e os relacionamentos do modelo anexo.

Os seguintes atributos devem seguir as seguintes regras:

- 1) id -> é o nome de uma chave primária com auto incremento;
- 2) atributo\_id -> é o nome de uma chave estrangeira.

### ❖ Evidências da prática:

→ Criando banco de dados e tabelas:

```
Microsoft Windows [versão 10.0.22621.2428]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\isabe>mkdir avaliacao08

C:\Users\isabe>cd avaliacao08

C:\Users\isabe>avaliacao08>sqlite3 avaliacao08.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> create table TB_IF (id integer primary key autoincrement, nome_if text, ano int, semestre int);
sqlite> create table TB_CAMPUS (id integer primary key autoincrement, nome text, if_id int, foreign key(if_id) references TB_IF(id));
sqlite> create table TB_CURSOS (id integer primary key autoincrement, nome text, campus_id int, foreign key(campus_id) references TB_CAMPUS(id));
sqlite> create table TB_LABORATORIO (id integer primary key autoincrement, nome text, responsavel_email text, curso_id int, foreign key(curso_id) references TB_CURSOS(id));
sqlite> create table TB_PROFESSOR (id integer primary key autoincrement, nome text, email text, celular text);
sqlite> create table TB_BOLSISTA (id integer primary key autoincrement, nome text, email text, celular text);
sqlite> create table TB_PROJETO (id integer primary key autoincrement, nome text, inicio date, termino date, laboratorio_id int, professor_id int, foreign key(laboratorio_id) references TB_LABORATORIO(id), foreign key (professor_id) references TB_PROFESSOR (id));
sqlite> create table TB_FAIXAHORARIA (id integer primary key autoincrement, E_Turno text CHECK (E_Turno IN ('MANHÃ', 'TARDE', 'NOITE')), E_Faixa_Horaria text CHECK(E_Faixa_Horaria IN ('A_PRIMEIRO', 'B_SEGUNDO', 'C_TERCEIRO', 'D_QUARTO', 'E_QUINTO')));
sqlite> create table TB_HORARIO_PLANEJADO(id integer primary key autoincrement, ano int, semestre int, dia int, bolsista_id int, faixa_horaria_id int, foreign key (bolsista_id) references TB_BOLSISTA(id), foreign key (faixa_horaria_id) references TB_FAIXAHORARIA(id));
sqlite> create table TB_FREQUENCIA(id integer primary key autoincrement, data date, E_Frequencia_Valida text CHECK (E_Frequencia_Valida IN ('SIM', 'NAO')), bolsista_id, projeto_id int, professor_id, horario_planejado_id, foreign key (bolsista_id) references TB_BOLSISTA(id), foreign key (projeto_id) references TB_PROJETO(id), foreign key (professor_id) references TB_PROFESSOR(id), foreign key (horario_planejado_id) references TB_HORARIO_PLANEJADO(id));
```

→ Evidenciando estrutura das tabelas pelo “.schema”:

```
sqlite> .schema
CREATE TABLE TB_IF (id integer primary key autoincrement, nome_if text, ano int, semestre int);
CREATE TABLE sqlite_sequence(name,seq);
CREATE TABLE TB_CAMPUS (id integer primary key autoincrement, nome text, if_id int, foreign key(if_id) references TB_IF(id));
CREATE TABLE TB_CURSOS (id integer primary key autoincrement, nome text, campus_id int, foreign key(campus_id) references TB_CAMPUS(id));
CREATE TABLE TB_LABORATORIO (id integer primary key autoincrement, nome text, responsavel_email text, curso_id int, foreign key(curso_id) references TB_CURSOS(id));
CREATE TABLE TB_PROFESSOR (id integer primary key autoincrement, nome text, email text, celular text);
CREATE TABLE TB_BOLSISTA (id integer primary key autoincrement, nome text, email text, celular text);
CREATE TABLE TB_PROJETO (id integer primary key autoincrement, nome text, inicio date, termino date, laboratorio_id int, professor_id int, foreign key(laboratorio_id) references TB_LABORATORIO(id), foreign key (professor_id) references TB_PROFESSOR (id));
CREATE TABLE TB_FAIXAHORARIA (id integer primary key autoincrement, E_Turno text CHECK (E_Turno IN ('MANHÃ', 'TARDE', 'NOITE')), E_Faixa_Horaria text CHECK(E_Faixa_Horaria IN ('A_PRIMEIRO', 'B_SEGUNDO', 'C_TERCEIRO', 'D_QUARTO', 'E_QUINTO')));
CREATE TABLE TB_HORARIO_PLANEJADO(id integer primary key autoincrement, ano int, semestre int, dia int, bolsista_id int, faixa_horaria_id int, foreign key (bolsista_id) references TB_BOLSISTA(id), foreign key (faixa_horaria_id) references TB_FAIXAHORARIA(id));
CREATE TABLE TB_FREQUENCIA(id integer primary key autoincrement, data date, E_Frequencia_Valida text CHECK (E_Frequencia_Valida IN ('SIM', 'NAO')), bolsista_id, projeto_id int, professor_id, horario_planejado_id, foreign key (bolsista_id) references TB_BOLSISTA(id), foreign key (projeto_id) references TB_PROJETO(id), foreign key (professor_id) references TB_PROFESSOR(id), foreign key (horario_planejado_id) references TB_HORARIO_PLANEJADO(id));
sqlite>
```

## → Modelo Entidade Relacionamento(MER) proposto:

