

Lista de Exercícios IV

John Vitor da Silva Cunha, 11821BCC005

Laís Saloum Deghaide, 11821BCC001

Vinícius Calixto Rocha, 11911BCC039

IV.

Modelo Relacional e Mapeamento EER-Relacional

- 1) Uma entidade é a representação de um objeto do mundo real ou um conceito, que são descritos no banco de dados.

Já a relação é a associação entre duas ou mais entidades que deve ser definido quando um tipo de entidade se refere a outro.

A relação pode ser pensada como uma tabela de valores em que cada linha na tabela representa uma coleção de valores de dados relacionados. No modelo relacional, cada linha na tabela representa um fato que corresponde a uma entidade ou relacionamento do mundo real. O nome da tabela e os nomes das colunas são usados para ajudar na interpretação do significado dos valores em cada linha. Uma linha pode ser chamada tupla, um cabeçalho de coluna é conhecido como atributo, e a tabela é chamada relação.

- 2) Uma instância de uma entidade é a ocorrência de um dado no banco de dados. Uma linha da tabela é chamada de tupla e cada tupla na relação representa uma instância de entidade em particular.

- 3) Um atributo corresponde a alguma propriedade de interesse que ajuda a descrever uma entidade. No modelo ER, o atributo é representado por uma elipse. No modelo R, é representado por meio de um esquema de relação onde o R é a relação e A_i é o nome de um atributo: $R(A_1, A_2, A_3, \dots, A_n)$. Já na tabela, o atributo é representado por uma coluna. Os nomes das colunas, nomes dos atributos) especificam como interpretar os valores de dados em cada linha, com base na coluna em que cada valor está. Todos os valores em uma coluna são do mesmo tipo de dado.

- 4) Chave estrangeira, ou Foreign Key (FK), é a chave que permite a referência a registros procedentes de outras tabelas. Ou seja, é o campo ou conjunto de campos que compõem a chave primária de uma outra tabela. A utilização da chave estrangeira possibilita a implementação da integridade de dados diretamente no banco de dados, conhecida como integridade referencial.

Exemplo:

PESSOA(idpes, nome, datanasc, sexo)

CARRO(idcar, idpes, modelo, datafabricação)

PK(PESSOA) = idpes

PK(CARRO) = idcar

$FK_{idpes}(\text{CARRO}) = PK(\text{PESSOA})$

SQL/DDDL

- 1) Podemos dizer que metadados são “dados sobre os dados”, ou seja, são as descrições dos construtores do esquema e suas restrições. Através deles, conseguimos efetuar a representação e identificação dos dados.

Exemplo de comando que cria metadados:

```
CREATE TABLE empregado (  
    nome varchar (15) not null,  
    ssn char(9) not null,  
    datanasc date,  
    endereço varchar(30),  
    sexo char CHECK (sexo IN ('M', 'F')),  
    salario DECIMAL(10, 2),  
    superssn char(9),  
    dno int not null,  
    primary key (ssn),  
);
```

Exemplo de comando que cria dados:

```
INSERT INTO empregado  
VALUES ('James', '888665555', DATE '1937-11-10', '450 Stone, Houston,  
TX', 'M', 55000, null, 1);
```

- 2) As chaves primárias são declaradas desta forma no SQL/DDDL:
nome_da_chave tipo_da_chave PRIMARY KEY;

Exemplo: idpes integer PRIMARY KEY AUTOINCREMENT;

Já as chaves alternativas são declaradas da seguinte forma: nome_do_valor
tipo_do_valor CONSTRAINT nome_do_valor_sk UNIQUE;

Exemplo: nomepes varchar(15) CONSTRAINT nomepes_sk UNIQUE;

- 3) CREATE TABLE curso (
 idcur SERIAL CONSTRAINT idcur_pk PRIMARY KEY,
 nomecur VARCHAR(30) CONSTRAINT nomecur_sk UNIQUE
);

```
CREATE TABLE aluno (  
    idalu SERIAL CONSTRAINT idalu_pk PRIMARY KEY,  
    nomealu VARCHAR(50) NOT NULL,  
    tipoalu VARCHAR(15) NOT NULL,
```

```
idcurso INT NOT NULL  
);
```

```
CREATE TABLE disciplina (  
  iddis SERIAL CONSTRAINT idalu_pk PRIMARY KEY,  
  nomedis VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE pre_requisito (  
  iddis INT NOT NULL CONSTRAINT iddis_pre_fk REFERENCES  
  disciplina,  
  idpre SERIAL,  
  CONSTRAINT idpre_pk PRIMARY KEY (iddis, idpre)  
);
```

```
CREATE TABLE professor (  
  idprof SERIAL CONSTRAINT idprof_pk PRIMARY KEY,  
  nomepro VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE turma (  
  idtur SERIAL CONSTRAINT idtur_pk PRIMARY KEY,  
  iddis INT NOT NULL CONSTRAINT iddis_pre_fk REFERENCES  
  disciplina,  
  ano DATE INT NOT NULL,  
  semestre INT NOT NULL,  
  codtur VARCHAR(10) INT NOT NULL,  
  CONSTRAINT (iddis, ano, semestre, codtur) UNIQUE  
);
```

```
CREATE TABLE turpro (  
  idtur INT NOT NULL CONSTRAINT idtur_turpro_fk REFERENCES turma,  
  idprof INT NOT NULL CONSTRAINT idprof_turpro_fk REFERENCES  
  professor,  
  CONSTRAINT turpro PRIMARY KEY (idtur, idprof)  
);
```

```
CREATE TABLE historico (  
  idalu INT NOT NULL CONSTRAINT idalu_hist_fk REFERENCES turma,  
  idturma INT NOT NULL CONSTRAINT idtur_hist_fk REFERENCES  
  turma,  
  nota DECIMAL (3, 2)
```

```
CONSTRAINT historico PRIMARY KEY (idalu, idturma)  
);
```