

Centro de Ciências Exatas e Engenharias

Cursos de Informática

DDL em Banco de Dados

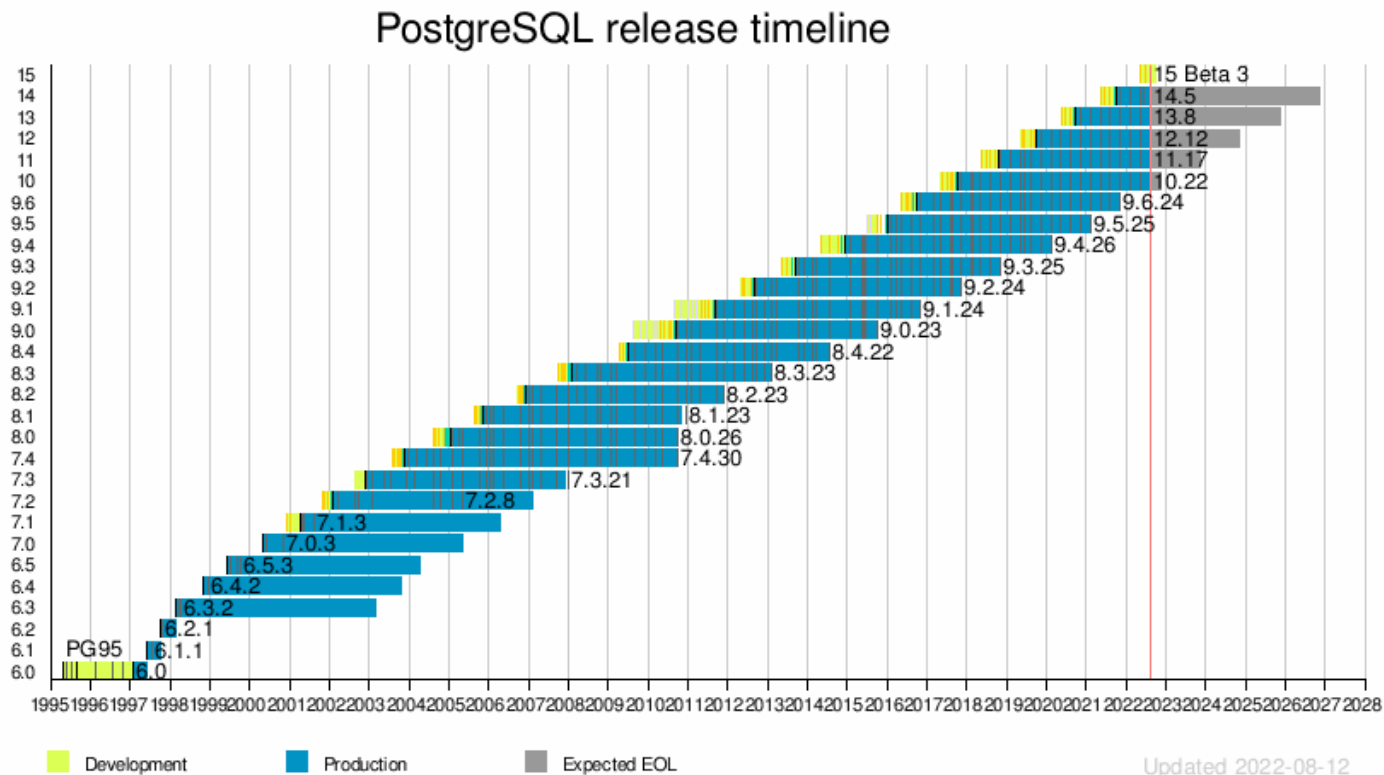
Prof. Daniel Luis Notari

Abril - 2023

SGBD PostgreSQL

- É um sistema gerenciador de banco de dados objeto relacional (SGBDOR), desenvolvido como projeto de código aberto (<http://www.postgresql.org/>).
- Características
 - Consultas complexas
 - Chaves estrangeiras
 - Integridade transacional
 - Controle de concorrência multi-versão
 - Suporte ao modelo híbrido objeto-relacional
 - Gatilhos (*triggers*)
 - Visões
 - Linguagem Procedural em várias linguagens (PL/pgSQL, PL/Python, PL/Java, PL/Perl) para Procedimentos armazenados
 - Estrutura para guardar dados Georeferenciados PostGIS

SGBD PostgreSQL



Download em <http://www.postgresql.org/download/>

<https://en.wikipedia.org/wiki/PostgreSQL>

SGBD PostgreSQL

What's a database project without statistics? _____

30+ Years Development	650+ Contributors	52,000+ Commits	55+ Local User Groups
1,500,000+ Lines of C	650+ Events	Millions of Happy Users	∞ Data Stored

<https://www.postgresql.org/about/>

Download em <http://www.postgresql.org/download/>

SQL

SQL (Structured Query Language)

Linguagem de pesquisa declarativa padrão para banco de dados relacional.

Muitas das características originais do SQL foram inspiradas na álgebra relacional.

<http://pt.wikipedia.org/wiki/SQL>

SQL

DDL - Linguagem de Definição de Dados

- Manipulação de esquemas

DML - Linguagem de Manipulação de Dados

- Manipulação de dados
- Consulta de dados

DCL - Linguagem de Controle de Dados

- Permissões de acesso

DTL - Linguagem de Transação de Dados

- Manipulação de transações

SQL

The SQL standard has gone through a number of revisions:

Year	Name	Alias	Comments
1986	SQL-86	SQL-87	First formalized by ANSI
1989	SQL-89	FIPS 127-1	Minor revision that added integrity constraints adopted as FIPS 127-1
1992	SQL-92	SQL2, FIPS 127-2	Major revision (ISO 9075), <i>Entry Level</i> SQL-92 adopted as FIPS 127-2
1999	SQL:1999	SQL3	Added regular expression matching, recursive queries (e.g. transitive closure), triggers , support for procedural and control-of-flow statements, nonscalar types (arrays), and some object-oriented features (e.g. structured types), support for embedding SQL in Java (SQL/OLB) and vice versa (SQL/JRT)
2003	SQL:2003		Introduced XML -related features (SQL/XML), window functions , standardized sequences, and columns with autogenerated values (including identity columns)
2006	SQL:2006		ISO/IEC 9075-14:2006 defines ways that SQL can be used with XML. It defines ways of importing and storing XML data in an SQL database, manipulating it within the database, and publishing both XML and conventional SQL data in XML form. In addition, it lets applications integrate queries into their SQL code with XQuery , the XML Query Language published by the World Wide Web Consortium (W3C), to concurrently access ordinary SQL-data and XML documents. ^[33]
2008	SQL:2008		Legalizes ORDER BY outside cursor definitions. Adds INSTEAD OF triggers, TRUNCATE statement, ^[34] FETCH clause
2011	SQL:2011		Adds temporal data (PERIOD FOR) ^[35] (more information at Temporal database#History). Enhancements for window functions and FETCH clause. ^[36]
2016	SQL:2016		Adds row pattern matching, polymorphic table functions, JSON
2019	SQL:2019		Adds Part 15, multidimensional arrays (MDarray type and operators)

ANSI (American National Standards Institute)
ISO (International Standards Organization)

<https://en.wikipedia.org/wiki/SQL>

SQL como DDL

Criação de banco de dados

- Comando CREATE DATABASE

Criação de esquemas

- Comando CREATE SCHEMA

SQL como DDL

- ---

Comandos:
 - Definição de tabelas – CREATE: Cria uma nova tabela na base de dados, especificando nome, atributos e restrições.
 - Alteração de tabelas - ALTER: Altera definições de uma tabela.
 - Remoção de tabelas – DROP: Remove uma tabela, quando suas definições não são mais necessárias.

SQL como DDL

- **CREATE TABLE**
- Colunas são especificadas primeiro, sob a forma:
 <nomeCol> <domínio> <restrição>
- Depois Chaves, integridade referencial e restrições de integridade

```
CREATE TABLE <nome_da_tabela>
( <nome_da_col1>    <tipo_da_col1> NOT NULL,
  <nome_da_col2>    <tipo_da_col2> NOT NULL,
  PRIMARY KEY <lista_de_nomes_de_col>,
  FOREIGN KEY <nomes_de_col> REFERENCES <nome_tab_ref>
    (<nome_da_col_ref>)
);
```

SQL como DDL

Tipos de dados para a definição de colunas (SQL Padrão)

- CHAR(tamanho)
- CHARACTER(tamanho)
- INT
- INTEGER
- SMALLINT
- NUMERIC(precisão,escala)
- DECIMAL(precisão,escala)
- DEC(precisão,escala)
- FLOAT(precisão)
- REAL
- DOUBLE PRECISION

SQL 2

- VARCHAR(tamanho)
- CHAR VARYING(tamanho)
- CHARACTER VARYING(tamanho)
- NCHAR(tamanho)
- NATIONAL CHAR(tamanho)
- NATIONAL CHARACTER(tamanho)
- VARYING(tamanho)
- BIT(tamanho)
- BIT VARYING(tamanho)
- DATE
- TIME(precisão)
- TIMESTAMP(precisão)
- INTERVAL

SQL como DDL

**Especificação
de chaves:**

Primária:

[CONSTRAINT nome] PRIMARY KEY(<nomeColuna>)

Estrangeira:

**[CONSTRAINT nome] FOREIGN KEY(<nomeCol>) REFERENCES
<NomeTabRefer>**

Alternativa:

UNIQUE KEY(nomeCol), (SQL2)

SQL como DDL

Restrições:

- NOT NULL: Restrição aplicadas a colunas cujos valores não podem ser nulos na definição de uma coluna

Valores Default

- Adiciona-se a cláusula DEFAULT <valor> logo após a restrição na definição de uma coluna

SQL como DDL

- **Integridade referencial:**
 - utilizada para representar uma relação entre tabelas com o vínculo entre chaves primárias e estrangeiras.
- Cuidados
 - Quando colunas são excluídas ou alteradas
 - Quando o valor do atributo da chave estrangeira é modificado na tabela referenciada
- Ações disparadas quando ocorrem violações:

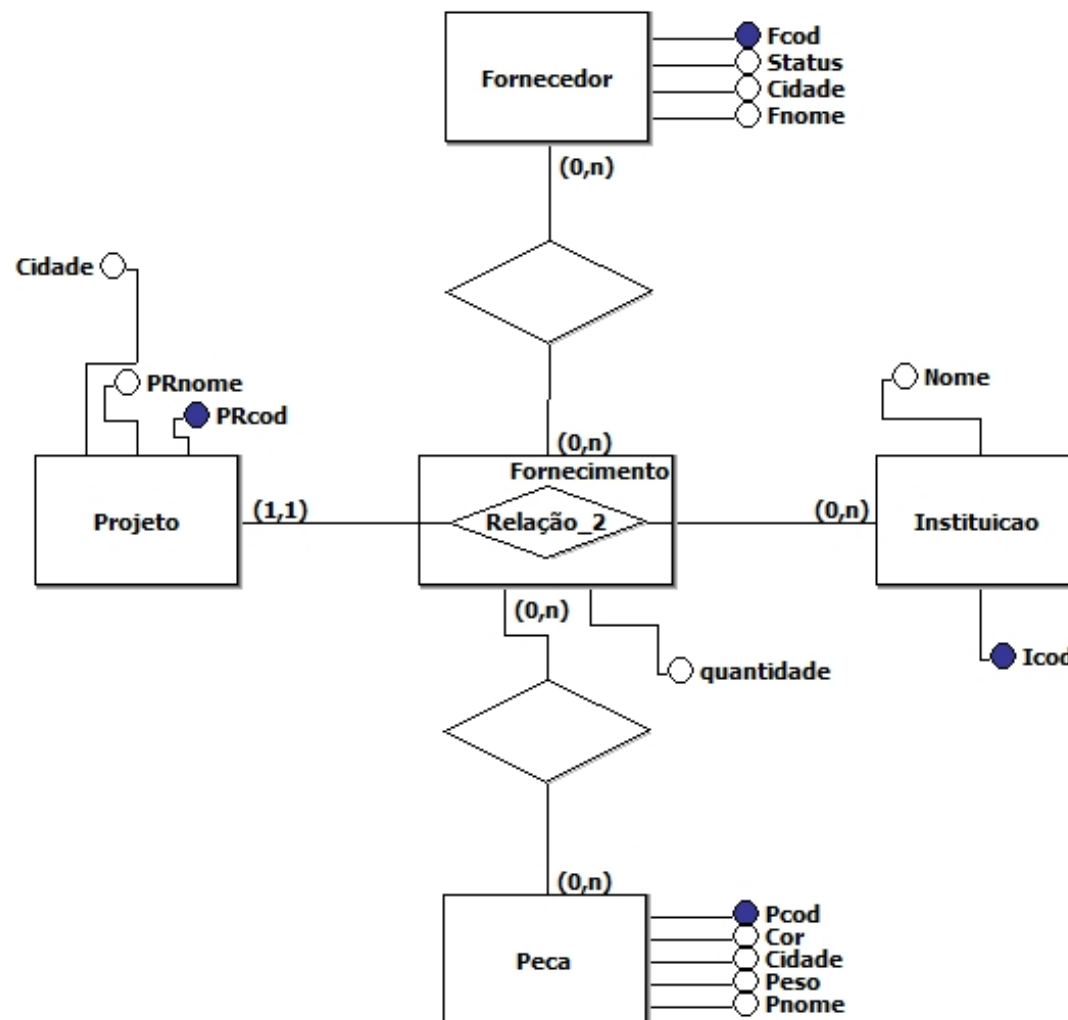
SET NULL

CASCADE

ON DELETE

ON UPDATE

Modelo Físico: Exemplo



Modelo Físico: Exemplo

-
- **Tradução do Diagrama ER para o modelo lógico**
 - Fornecedor (Fcod, Fnome, Status, Cidade)
 - Peça (Pcod, Pnome, Cor, Peso, Cidade)
 - Projeto (PRcod, lcod, PRnome, Cidade)
 - Fornecimento (#Fcod, #Pcod, #lcod, #Prcod, quantidade)
 - Instituição (lcod, nome)

Modelo Físico: Exemplo

- Tradução do modelo lógico para o modelo físico

CREATE TABLE fornecedor(

fcod integer NOT NULL,

fnome varchar(40) NOT NULL,

status char(1) NOT NULL DEFAULT 'S',

cidade varchar(40) NOT NULL,

PRIMARY KEY (fcod));

CREATE TABLE peca (

pcod integer NOT NULL,

pnome varchar(40) NOT NULL,

cor char(12) NOT NULL,

peso float NOT NULL,

cidade varchar(40) NOT NULL,

PRIMARY KEY (pcod));

Modelo Físico: Exemplo

- Tradução do modelo lógico para o modelo físico

CREATE TABLE instituicao (

 icod integer NOT NULL,
 nome varchar(40) NOT NULL,
 CONSTRAINT pk_instituicao
PRIMARY KEY (icod));

CREATE TABLE projeto (

 prcod integer NOT NULL,
 icod integer NOT NULL,
 prnome varchar(40) NOT NULL,
 cidade varchar(40) NOT NULL,
PRIMARY KEY (prcod, icod),
CONSTRAINT fk_projeto FOREIGN KEY
(icod) REFERENCES instituicao(icod));

Modelo Físico: Exemplo

- **Tradução do modelo lógico para o modelo físico**

```
CREATE TABLE fornecimento (  
    fcod integer NOT NULL,  
    pcod integer NOT NULL,  
    prcod integer NOT NULL,  
    icod integer NOT NULL,  
    quantidade integer NOT NULL,  
    CONSTRAINT pk_fornecimento PRIMARY KEY (fcod, pcod, prcod, icod),  
    CONSTRAINT fk_fornecimento1 FOREIGN KEY (fcod) REFERENCES fornecedor (fcod),  
    CONSTRAINT fk_fornecimento2 FOREIGN KEY (pcod) REFERENCES peca (pcod),  
    CONSTRAINT fk_fornecimento3 FOREIGN KEY (prcod,icod) REFERENCES  
    projeto(prcod, icod) ON DELETE CASCADE  
);
```

SQL como DDL

Remoção de tabelas

Elimina completamente a tabela, estando vazia ou não

```
DROP TABLE <nome_da_tabela>;
```

Exemplo:

- DROP TABLE Empregado;

Não há como recuperar a tabela removida

SQL como DDL

Remoção de tabelas

SQL2 inclui cláusulas:

- CASCADE: Se existirem outros objetos que dependam da tabela excluída serão excluídos também
- RESTRICT: Exclui a tabela somente se não existirem objetos do banco de dados que dependam da tabela

Exemplo:

- DROP TABLE Empregado CASCADE;

SQL como DDL

Alteração de tabelas

ALTER TABLE <nome_da_tabela> <alteração>;

Adicionar uma coluna:

- ALTER TABLE Empregado

ADD Telefone VARCHAR(30),

ADD sexo CHAR(1) DEFAULT 'F';

Remover uma coluna:

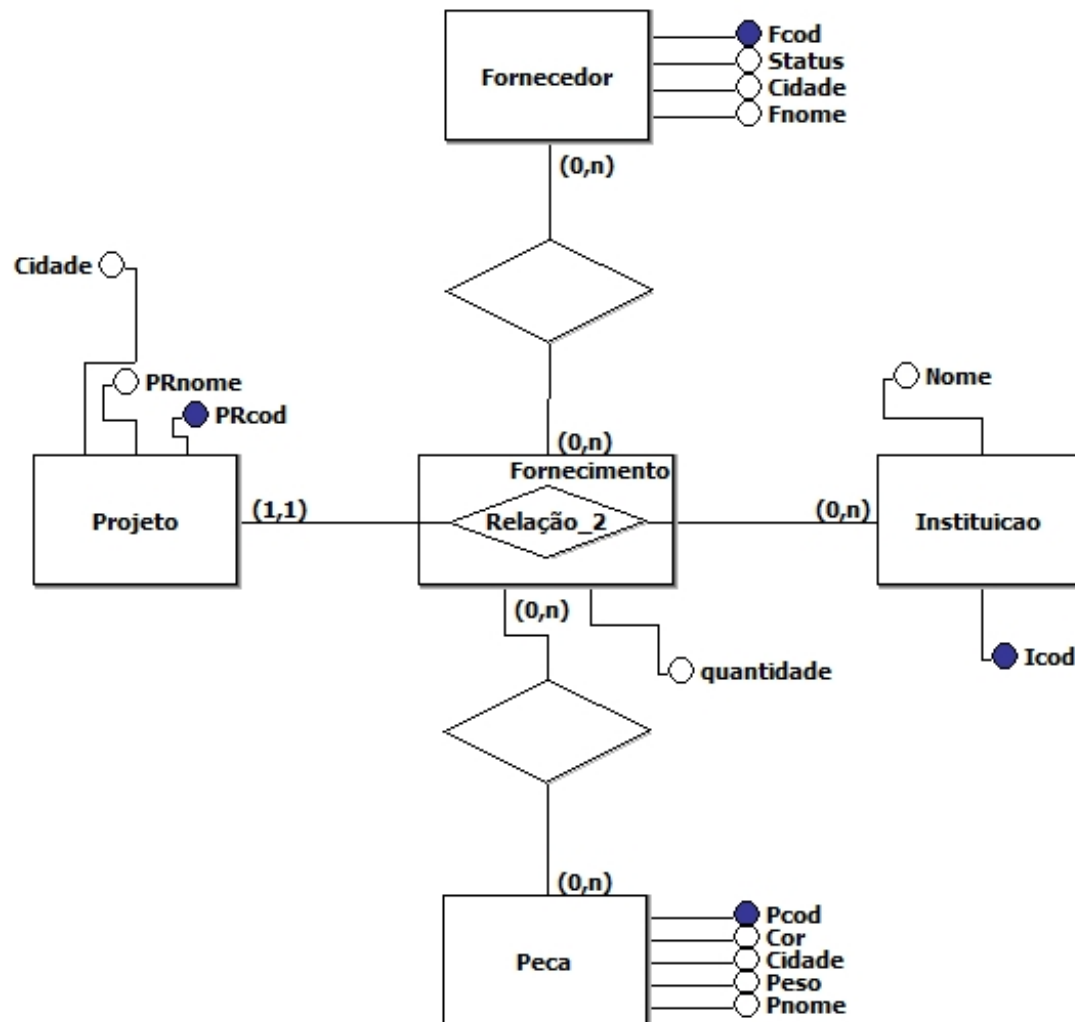
- ALTER TABLE Empregado

DROP nome;

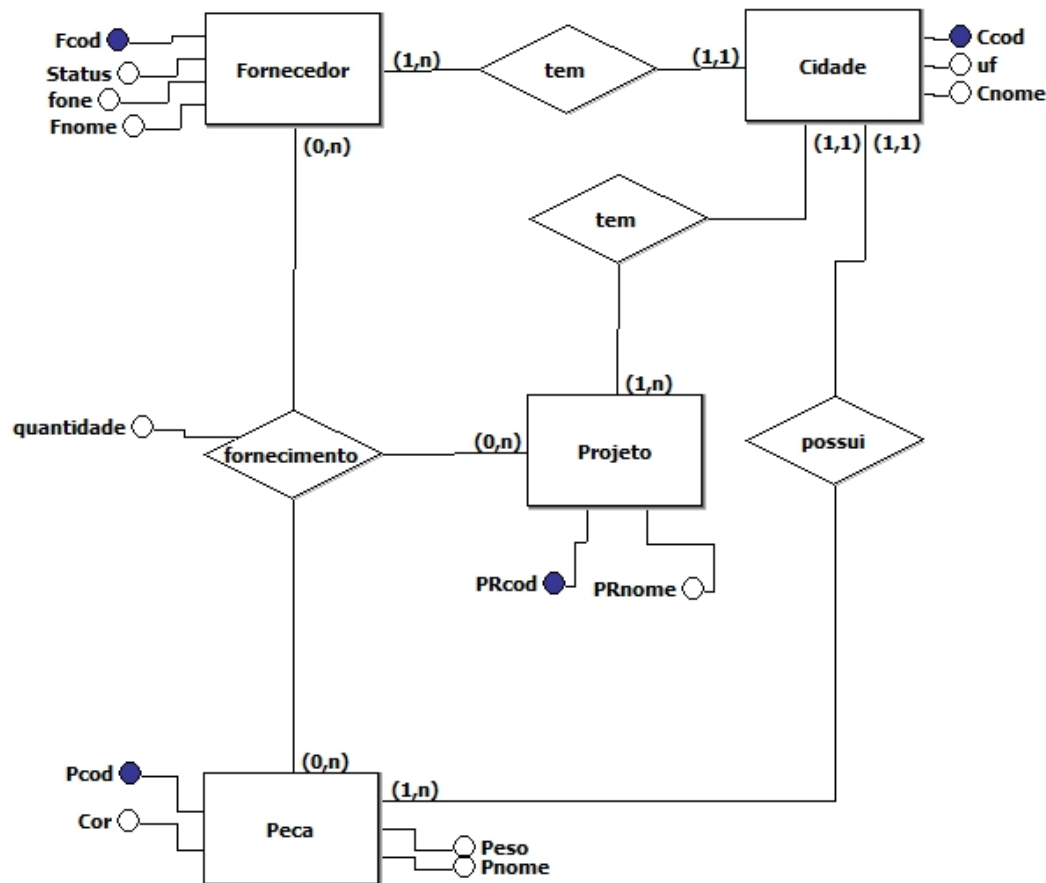
SQL como DDL

-
- **Alteração de tabelas**
 - Excluir restrições de chave primária:
 - ALTER TABLE Empregado
DROP CONSTRAINT empregado_pkey;
 - Exclusão de uma chave estrangeira:
 - ALTER TABLE Empregado
DROP CONSTRAINT empregado_fkey;
 - Adicionar chave estrangeira:
 - ALTER TABLE Empregado
ADD COLUMN Codesupervisor int,
ADD PRIMARY KEY (Codesupervisor),
ADD CONSTRAINT empregado_fkey FOREIGN
KEY(Codesupervisor) REFERENCES Empregado(ecod);

Modelo Físico: exemplo original



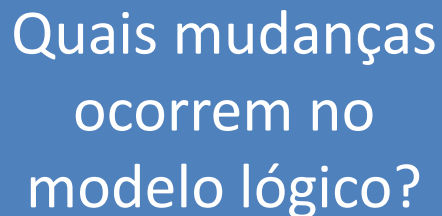
Modelo Físico: exemplo normalizado



Quais
mudanças
ocorrem no
DER?

Modelo Físico: Exemplo

- **Tradução do Diagrama ER para o modelo lógico**
- Fornecedor (Fcod, Fnome, Status, #Ccod, fone)
- Cidade (Ccod, Cnome, uf)
- Peça (Pcod, Pnome, Cor, Peso, #Ccod)
- Projeto (PRcod, PRnome, #Ccod)
- Fornecimento (#Fcod, #Pcod, #PRcod, Quantidade)



Quais mudanças
ocorrem no
modelo lógico?

Modelo Físico: modificado

- Tradução do modelo lógico para o modelo físico

DROP TABLE instituicao cascade;

CREATE TABLE cidade (

ccod integer NOT NULL,

cnome varchar (40) NOT NULL,

uf character(2) NOT NULL,

PRIMARY KEY (ccod));

ALTER TABLE fornecedor

drop cidade,

add ccod integer not null,

add fone integer not null,

add constraint fk_fornecedor

foreign key (ccod) references cidade(ccod);

Modelo Físico: modificado

- Tradução do modelo lógico para o modelo físico

ALTER TABLE peca

drop cidade,

add ccod integer not null,

add constraint fk_peca1 foreign key (ccod) references
cidade(ccod);

ALTER TABLE fornecimento

drop icod;

Modelo Físico: modificado

- Tradução do modelo lógico para o modelo físico

ALTER TABLE projeto

drop cidade,

drop icod,

add ccod integer not null,

add constraint pk_projeto1 foreign key (ccod) references

cidade(ccod),

add primary key(prcod);

ALTER TABLE fornecimento

add primary key(fcod, pcod, prcod),

add constraint fk_fornecimento4 foreign key (prcod) references projeto(prcod);

SQL como DML

- **Inserção de dados em tabelas**

INSERT INTO <nome_tabela> [(<colunas>)]
VALUES (<lista de valores>)

- Exemplo:

- insert into cidade (ccod, cnome, uf) values (1, 'csl', 'rs');
- insert into fornecedor (fcod, fnome, status, ccod, fone) values (1, 'agropecuaria clarice', 'n', 1, 123);

Integridade
referencial

SQL como DML

- **Alterar dados em uma tabela**

UPDATE <nome_tabela>

SET <atribuição>

WHERE <condição>

- Exemplo:

UPDATE cidade

SET cnome = 'caxias do sul'

WHERE ccod = 1;

SQL como DML

- **Deletar dados em uma tabela**

```
DELETE FROM <nome_tabela>  
[ WHERE <condição> ]
```

- Exemplo:
 - DELETE FROM fornecedor
 - DELETE FROM cidade
WHERE ccod = 1;

Exercício

Abrir o PgAdmin IV

Criar um banco de dados com o seu nome

Selecione o esquema “public” no seu banco de dados e clique no botão SQL

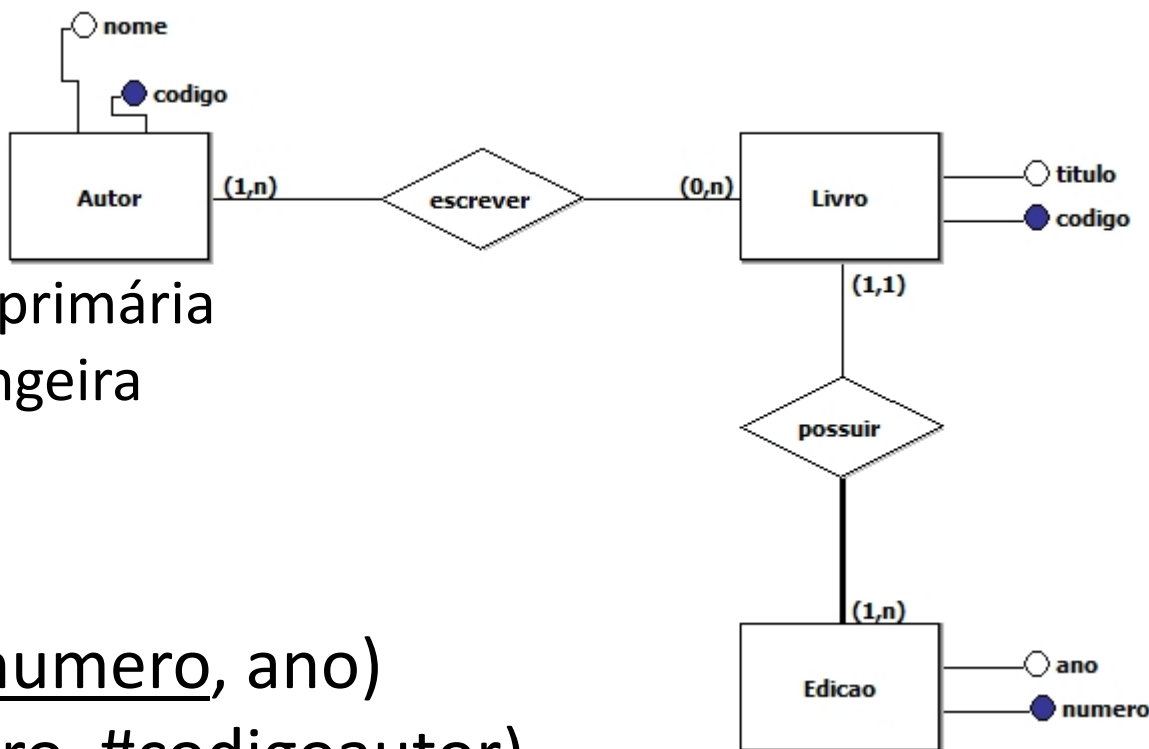
Teste a criação e alteração dos esquemas

Insira Dados

Baixe o script do AVA

Exercício (salvar webfolio)

- Para o modelo lógico apresentado a seguir, faça a criação do modelo físico e faça a inserção de dados. Grupos de dois alunos.



Legenda:

- campo sublinhado : chave primária
- campo com #: chave estrangeira
- Autor** (codigo, nome)
- Livros** (codigo, titulo)
- Edicao** (#codigoLivro, numero, ano)
- LivroAutor** (#codigoLivro, #codigoautor)

Exercício (salvar webfolio)

- Refaça o exercício (modelo físico e inserção de dados), agora para a tradução use somente *ids* ao invés de chaves compostas. Grupos de dois alunos.
- Orientações:
 1. Crie um atributo `id_<nome_tabela>` para cada tabela e defina-o como chave primária.
 2. Os atributos definidos anteriormente como chave primária devem ser definidos como chave única.
 3. Não utilize definição de chave estrangeira.
 4. Pesquise como se define um identificador auto-incrementável no SGBD PostgreSQL