

BANCOS DE DADOS

BD-I
SQL (Structured Query Language)
Prof. Moisés Omena

BANCO DE DADOS

Implementando modelo
Físico usando a linguagem
SQL
Prof. Moisés Omena

SQL (Structured Query Language)

- Linguagem Estruturada de Consulta
- Formada pelo conjunto das linguagens:
 - DDL (Data Definition Language)
 - DML (Data Manipulation Language)
 - DQL (Data Query Language):
 - DCL (Data Control Language):
 - DTL (Data Transaction Language):

Notas de Aula - Prof. Moisés Omena

Estrutura da linguagem SQL

SQL (Structured Query Language)

SQL

DDL

DML

DQL

DCL

DTL

- DDL (Data Definition Language): Linguagem de Definição de Dados
- DML (Data Manipulation Language): Linguagem de Manipulação de Dados
- DQL (Data Query Language): Linguagem de Consulta de Dados
- DCL (Data Control Language): Linguagem de Controle de Dados
- DTL (Data Transaction Language): Linguagem de Transação de Dados

Notas de Aula - Prof. Moisés Omena

DDL (Data Definition Language)

- Linguagem de Definição de Dados
 - CREATE: Cria uma estrutura
 - ALTER: Altera uma estrutura
 - DROP: Exclui uma estrutura

Notas de Aula - Prof. Moisés Omena

DML (Data Manipulation Language)

- Linguagem de Manipulação de Dados
 - INSERT: Insere dados
 - UPDATE: Altera dados
 - DELETE: Exclui dados

Notas de Aula - Prof. Moisés Omena

DQL (Data Query Language)

- Linguagem de Consulta de Dados
 - SELECT: Retorna dados
 - Ordenação de dados
 - Agrupamento de dados
 - Funções aritméticas
 - Filtros de seleção

Notas de Aula - Prof. Moisés Omena

DCL (Data Control Language)

- Linguagem de Controle de Dados
 - GRANT: Habilita acesso a dados e operações
 - REVOKE: Revoga acesso a dados e operações

Notas de Aula - Prof. Moisés Omena

DTL (Data Transaction Language)

- Linguagem de Transação de Dados
 - START TRANSACTION: Inicia a transação
 - COMMIT: Concretiza a transação
 - ROLLBACK: Anula a transação

Notas de Aula - Prof. Moisés Omena



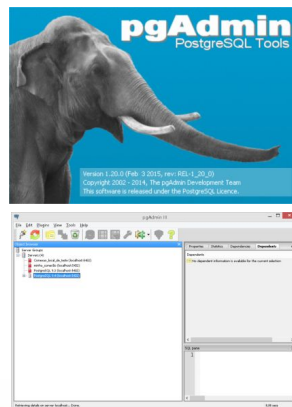
BANCO DE DADOS

Criando nosso DATABASE

Abra o PgAdmin

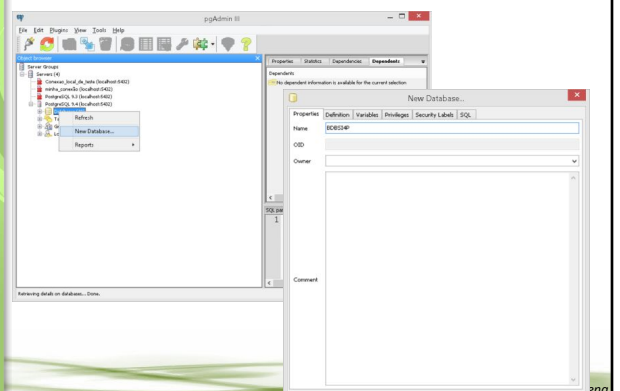


- Faça a conexão local com o banco de dados PostgreSQL

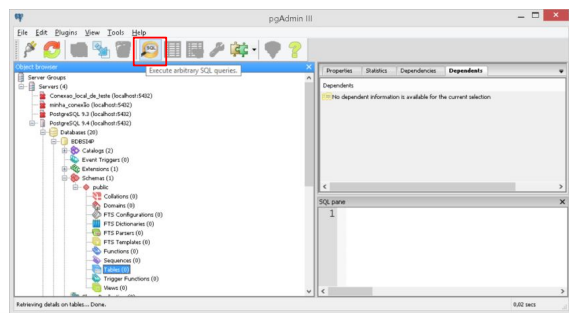


Notas de Aula - Prof. Moisés Omena

Criando banco de dados (BDBSI4P)



Enviando instruções ao banco de dados desejado



Notas de Aula - Prof. Moisés Omena

DDL (Data Definition Language)

- **CREATE TABLE:** Cria uma tabela

```
CREATE TABLE <table_name>  
(<column_name> <datatype> {null | Not null} [...]);
```

Definições:

table_name → refere-se ao nome que será dado a tabela

column_name → nome para cada coluna a ser criada para a tabela

datatype → refere-se ao tipo do dado (numeric, char, date) que será atribuído a coluna

Notas de Aula - Prof. Moisés Omena

Tipos de Campos (<datatype>)

- **TINYINT:**
 - é um número inteiro com ou sem sinal. Com sinal a margem de valores válida é de -128 até 127. Sem sinal, a margem de valores é de 0 até 255
- **BIT OU BOOL:**
 - um número inteiro que pode ser 0 ou 1.
- **INT**
 - Um inteiro de tamanho normal. A faixa com sinal é de -2147483648 a 2147483647. A faixa sem sinal é de 0 a 4294967295.
- **BIGINT**
 - Um inteiro grande. A faixa com sinal é de -9223372036854775808 a 9223372036854775807. A faixa sem sinal é de 0 a 18446744073709551615.
- **DOUBLE**
 - Um número de ponto flutuante de tamanho normal (dupla-precisão). Valores entre -1.7976931348623157E+308 até -2.2250738585072014E-308.
- **DECIMAL**
 - Um número de ponto flutuante não empacotado. Se comporta como um campo CHAR: "não empacotado" significa que o número é armazenado como uma string, usando um caractere para cada dígito do valor.

Notas de Aula - Prof. Moisés Omena

Tipos de dados ou <datatype>

- Tipos de dados são formatos de armazenamentos dos dados/informações em colunas de uma tabela.
 - **Integer:**
 - De -2147483648 até 2147483647
 - **BigInt**
 - Um inteiro grande. A faixa com sinal é de -9223372036854775808 a 9223372036854775807. A faixa sem sinal é de 0 a 18446744073709551615.
 - **Numeric**
 - números com valores decimais de maneira precisa: numeric (precisão,escala) ou seja numeric (número de dígitos, dígitos após ponto).

Notas de Aula - Prof. Moisés Omena

Tipos de dados

- Tipos de dados são formatos de armazenamentos dos dados/informações em colunas de uma tabela.
 - **Float**
 - flutuante de precisão simples. Os valores válidos vão desde -175494351E-38 até 3.402823466E+38
 - **Serial**
 - Tipo numérico ,inteiro, e comumente utilizado para gerar valores sequenciais automaticamente, estando normalmente relacionado as colunas chave primária com auto-incremento pela sua própria característica.

Notas de Aula - Prof. Moisés Omena

- **DATE**
 - Uma data. A faixa suportada é entre '1000-01-01' e '9999-12-31'. MySQL mostra valores DATE no formato 'AAAA-MM-DD', mas permite a você a atribuir valores a campos DATE utilizando tanto strings quanto números
- **DATETIME**
 - Um combinação de hora e data. A faixa suportada é entre '1000-01-01 00:00:00' e '9999-12-31 23:59:59'. MySQL mostra valores DATETIME no formato 'AAAA-MM-DD HH:MM:SS', mas permite a você que atribuir valores a campos DATETIME utilizando strings ou números
- **TIMESTAMP[(M)]**
 - Um timestamp. A faixa é entre '1970-01-01 00:00:00' e algum momento no ano 2037.

Notas de Aula - Prof. Moisés Omena

Blob (Binary Large Object) e Text

- TINYBLOB, TINYTEXT
 - Um campo BLOB ou TEXT com tamanho máximo de 255 ($2^8 - 1$) caracteres
- BLOB, TEXT
 - Um campo BLOB ou TEXT com tamanho máximo de 65535 ($2^{16} - 1$) caracteres
- MEDIUMBLOB, MEDIUMTEXT
 - Um campo BLOB ou TEXT com tamanho máximo de 16777215 ($2^{24} - 1$) caracteres
- LONGBLOB, LONGTEXT
 - Um campo BLOB ou TEXT com tamanho máximo de 4294967295 ou 4G ($2^{32} - 1$) caracteres

Notas de Aula - Prof. Moisés Omena

Tipos de dados

- Char

Possibilita guardar um conjunto de caracteres definidos pelo usuário. Por exemplo char(10), guardaria 10 caracteres. Entretanto se você não utilizar os 10 caracteres eles serão preenchidos com espaços em branco, ocorrendo assim desperdício de espaço.
- Varchar

Ao contrário do formato Char, varchar lida de modo dinâmico com a quantidade de caracteres, podendo-se especificar um determinado número, (ex: varchar(150)), mas que ao serem inseridos 20 caracteres nenhum preenchimento automático de caracteres espaço ocorrerá. Por tal característica o tipo varchar é amplamente utilizado.

Notas de Aula - Prof. Moisés Omena

Os Tipos CHAR e VARCHAR

- Os tipos CHAR e VARCHAR são parecidos, mas diferem no modo como são armazenados e recuperados.
- CHAR é fixado pelo tamanho declarado na criação da tabela.
 - Qualquer valor entre 1 e 255 caracteres
 - Utiliza todo espaço declarado, preenchendo automaticamente o que ficaram em branco.
- VARCHAR são strings de tamanho variável.
 - qualquer tamanho entre 1 e 255, assim como para campo CHAR.
 - No entanto, diferente de CHAR, valores VARCHAR são armazenados usando apenas quantos caracteres forem necessários, mais 1 byte para gravar o tamanho.
 - Se você atribuir um valor para uma coluna CHAR ou VARCHAR que exceda o tamanho máximo da coluna, o valor é truncado para este tamanho.

Notas de Aula - Prof. Moisés Omena

Tipos de dados

- Money

Destinado a receber valores monetários. Possui duas casas decimais e aceita valores grandes no intervalo de -92.233.720.386.547.758,08 até 92.233.720.386.547.758,07.
- Text

Não implementa nenhum limite para inserção de caracteres. Utilizado para campos onde serão inclusos textos longos.

Notas de Aula - Prof. Moisés Omena

Tipos de dados

- Date

Representação de datas. De 4713 A.C até 5.874.897 D.C.
- Time

Formato para horas do dia.
- Timestamp

Formato que inclui tanto data quanto horas
- Boolean

Representa valores verdadeiros ou falsos (também permite utilização de NULL)

Notas de Aula - Prof. Moisés Omena

DDL (Data Definition Language)

- CREATE TABLE: Cria uma tabela

```
CREATE TABLE <table_name>  
(<column_name> <datatype> {null | Not null} [...]);
```

Execução de código:

```
CREATE TABLE ALUNO (  
  CODIGO INT NOT NULL,  
  NOME VARCHAR(45),  
  DATA_NASCIMENTO DATE);
```

Notas de Aula - Prof. Moisés Omena

DML (Data Manipulation Language)

- **SELECT:** Selecionar dados

```
SELECT * FROM <table_name>
```

Definições:

table_name → refere-se ao nome que será dado a tabela
***** → representa todas as colunas que a tabela possui

OBS: A instrução **SELECT** será melhor detalhada mais a frente, o objetivo de apresentá-la aqui é o de possibilitar analisar se os campos determinados foram criados.

Notas de Aula - Prof. Moisés Omena

Exercícios

- Crie a estrutura da tabela abaixo (testaremos os dados mais a frente quando tratarmos inserção)

```
CREATE TABLE tipos_armazenamento (  
    campo0 INTEGER,  
    campo1 INT,  
    campo2 BIGINT,  
    campo3 NUMERIC(9,3),  
    campo4 SERIAL,  
    campo5 MONEY,  
    campo6 CHAR(5),  
    campo7 VARCHAR(50),  
    campo8 TEXT,  
    campo9 FLOAT,  
    campo10 DATE,  
    campo11 TIME,  
    campo12 TIMESTAMP,  
    campo13 BOOLEAN  
);
```

Notas de Aula - Prof. Moisés Omena

Observações

- Apesar de existirem vários tipos de dados os tipos mais largamente utilizados são:

- **INTEGER**
- **FLOAT**
- **CHAR()**
- **VARCHAR()**
- **DATE**
- **TIMESTAMP**

Notas de Aula - Prof. Moisés Omena

DDL (Data Definition Language)

- **ALTER TABLE:** Alterando uma tabela

```
ALTER TABLE <table_name>
```

```
ADD (<column_name> <datatype> {null | Not null} [...]);
```

definições:

table_name → refere-se ao nome da tabela que sofrerá a alteração
ADD COLUMN → refere-se a modificação que será adicionada a tabela
DROP COLUMN → Refere-se a modificação de exclusão da tabela
RENAME COLUMN → Refere-se ao tipo de modificação que será executada na tabela
ALTER COLUMN → Refere-se ao tipo de modificação que será executada na tabela

Notas de Aula - Prof. Moisés Omena

DDL (Data Definition Language)

- **ALTER TABLE:** Alterando uma tabela

```
ALTER TABLE <table_name>
```

```
ADD (<column_name> <datatype> {null | Not null} [...]);
```

Execução de código:

```
ALTER TABLE aluno ADD nome VARCHAR(50);  
ALTER TABLE aluno DROP nome;  
ALTER TABLE aluno RENAME COLUMN nome TO nome_completo;  
ALTER TABLE aluno ALTER COLUMN nome_completo TYPE varchar(80);  
ALTER TABLE aluno ALTER COLUMN codigo SET NOT NULL;  
ALTER TABLE aluno ALTER COLUMN codigo DROP NOT NULL;
```

Notas de Aula - Prof. Moisés Omena

DDL (Data Definition Language)

- **DROP TABLE:** Apaga uma tabela

```
DROP TABLE [IF EXISTS] <table_name>
```

definições:

table_name → refere-se ao nome da tabela que sofrerá a alteração
IF EXISTS → verifica se a tabela existe para não causar um erro de execução.

Execução de código:

```
DROP TABLE ALUNO  
OU  
DROP TABLE IF EXISTS ALUNO
```

Notas de Aula - Prof. Moisés Omena

Exercícios

Após realização dos exercícios envie as instruções que foram utilizadas em um arquivo txt para o professor por meio do Ava-IFES
(OBS: não use acentos nos nomes de campos)

1. Crie as seguintes tabelas e seus respectivos campos no banco de dados BDBSI4P:
 - a. PROJETO: numero, nome e localização;
 - b. EMPREGADO: rg, nome,cpf,depto,rg_supervisor,salario e dat_init_sal
 - c. DEPARTAMENTO: numero, nome e rg_gerente
 - d. DEPENDENTE: codigo, rg_responsável,nome_dependente,nascimento, relacao e sexo
 - e. EMPREGADO_PROJETO: codigo,rg_empregado,numero_projeto e horas
 - f. DEPARTAMENTO_PROJETO: codigo, numero_depto e numero_projeto
 - g. HISTORICO_SALARIO: rg, dat_ini_sal,dat_fim_sal e salario

Notas de Aula - Prof. Moisés Omena

Exercícios

(OBS: não use acentos nos nomes de campos)

1. Altere a tabela PROJETO
 - a. Exclua o campo localizacao
 - b. Adicione o campo localidade
 - c. Adicione o campo logico
2. Altere a tabela EMPREGADO
 - a. Crie o campo funcao
 - b. Altere o campo nome para nome_completo
 - c. Apague o campo "dat_ini_sal"
3. Altere a tabela DEPARTAMENTO
 - a. Crie o campo descricao_departamento
 - b. Crie o campo data_criacao
 - c. Altere o nome para nome_departamento
 - d. Apague o campo data_criacao
4. Cria uma tabela denominada ALUNO com os campos código e nome
5. Altere em um único comando os campos codigo para tipo float e nome para varchar(30)

Notas de Aula - Prof. Moisés Omena

Exercícios

6. Apague a tabela Aluno
7. Altere a tabela historico_salario
 - a. Exclua o campo dat_ini_sal
 - b. Adicione o campo salario_total_anual
 - c. Altere o campo salario para salario_mensal
8. Altere a tabela empregado_projeto
 - a. Crie o campo função
 - b. Crie o campo data_inicio
 - c. Crie o campo data_fim
9. Altere a tabela Departamento_projeto
 - a. Crie o campo demanda_de_funcionarios
 - b. Crie o campo horas_destinadas
 - c. Renomear o campo codigo para número

Notas de Aula - Prof. Moisés Omena

Trabalho em Andamento!

- Aplique os conhecimentos obtidos nesta aula para criar as tabelas do seu banco de dados!
- Enviar arquivo com código para o GIT.

Notas de Aula - Prof. Moisés Omena