unesp

Colégio Técnico Industrial “Prof. Isaac Portal Roldán”

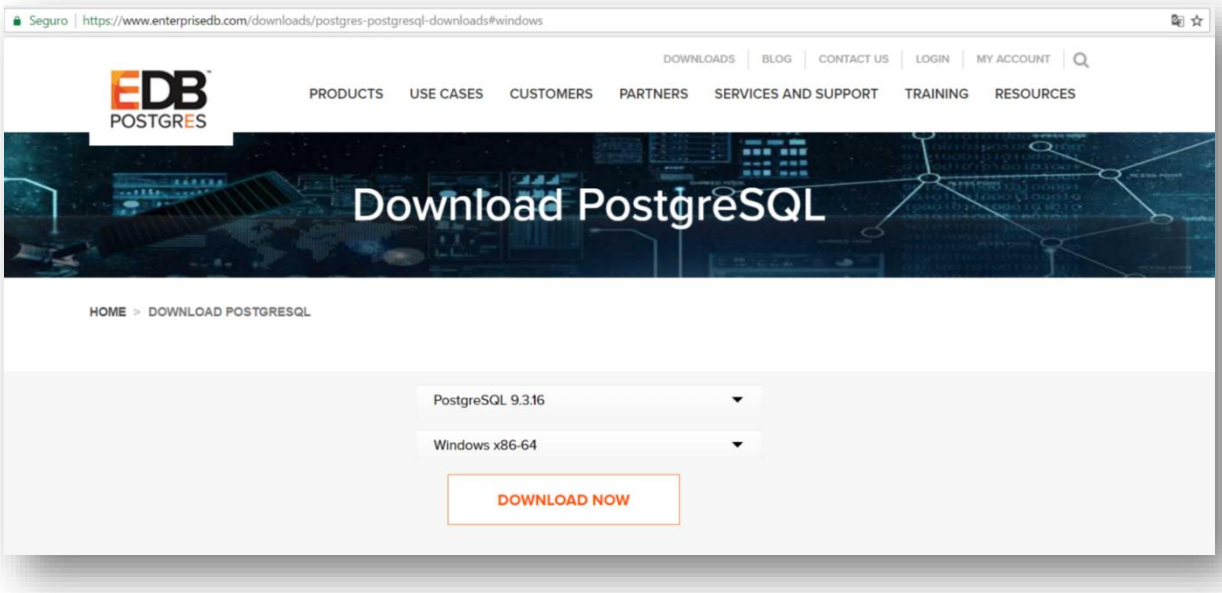
Professora Dra. Kátia L. Zambon e Professor Vitor Simeão
(Adaptação para o PostgreSQL por Prof. Vitor Simeão)

Apostila de Banco de Dados

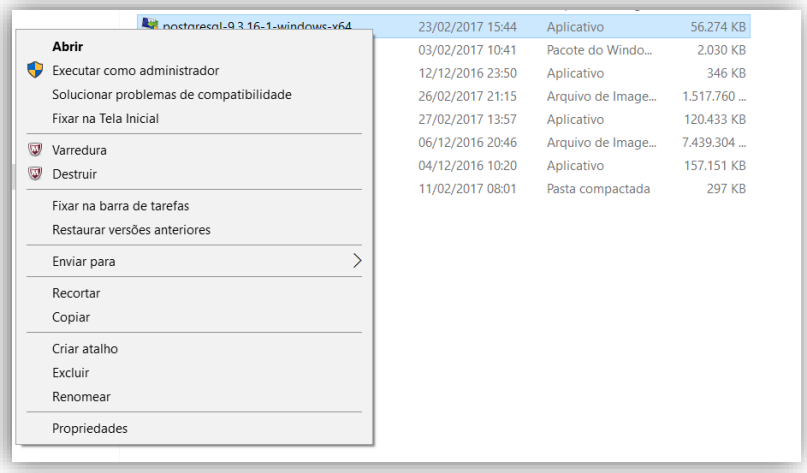
Bauru - 2017

Apresentação

Bem-vindos! Para a realização das atividades, você precisa instalar o PostgreSQL em seu computador. No navegador de internet de sua preferência, digite o endereço https://www.enterprisedb.com/downloads/postgres-postgresql-downloads#windows e escolha a versão 9.3.16 e o sistema operacional Windows x86-32 ou Windows x86-64, conforme o processador do seu equipamento. Depois é só efetuar o download.



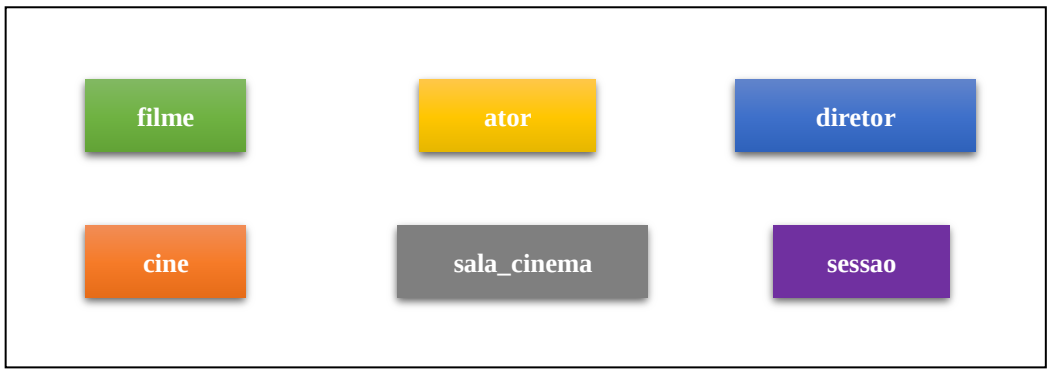
O arquivo postgresql-9.3.16-1-windows-x64.exe (57.0 MB) será copiado para o seu computador, provavelmente na pasta C:\Users\...\Downloads. Selecione o arquivo com o botão direito do mouse e escolha Executar como administrador. Além do PostgreSQL e outros programas, o aplicativo de administração de banco de dados pgAdmin III também será instalado.



ATIVIDADE – CINEMA

A atividade CINEMA é praticamente um passo-a-passo. Utilizando o aplicativo pgAdmin III, você criará um banco de dados para armazenar os dados dos filmes, atores, diretores, cinemas, salas de cinema e sessões.

Logo abaixo, você encontra uma representação gráfica do banco de dados CINEMA e suas tabelas. Cada tabela armazena os dados sobre alguma realidade, por exemplo, a tabela filme armazena algumas características sobre os filmes que são exibidos nas salas de cinema, tais como: o título original do filme, o seu ano de lançamento, o seu gênero, o nome do diretor, etc.
O mesmo ocorre com as demais tabelas.



As tabelas possuem uma estrutura de dados para o armazenamento de tais características.

Criar a tabela **filme**

PK	Nome do campo	Tipo
X	id_filme	INTEGER, NOT NULL
	titulo_original	CHARACTER VARYING, LENGTH 60
	titulo_portugues	CHARACTER VARYING, LENGTH: 60
	ano	NUMERIC, LENGTH: 4, PRECISION: 0
	genero	CHARACTER, LENGTH: 20
	pais	CHARACTER, LENGTH: 20
	diretor	CHARACTER, LENGTH: 40

Utilizando o pgAdmin III armazene os dados na tabela **filme**

id_filme	titulo_original	titulo_portugues	ano	genero	pais	diretor
1	Le Fabuleux Destin d'Amélie Poulain	O Fabuloso Destino de Amélie Poulain	2001	Drama	França	Jean-Pierre Jeunet
2	Tropa de Elite 2	Tropa de Elite 2	2010	Policial	Brasil	José Padilha
3	Wall-E	Wall-E	2008	Animação	EUA	Andrew Stanton
4	The GodFather	O Poderoso Chefão	1972	Policial	EUA	Francis Ford Coppola
5	Central do Brasil	Central do Brasil	1998	Drama	Brasil	Walter Salles
6	Catch Me If You Can	Prenda-me se for capaz	2002	Drama	EUA	Steven Spielberg

Agora que os dados foram armazenados, vamos usar o comando de consulta SELECT. O comando SELECT é um comando da linguagem de consulta estruturada, também conhecida como SQL (Structured Query Language). A linguagem SQL acompanha os sistemas de gerenciamento de banco de dados relacionais.

O comando SELECT recupera os dados armazenados nos bancos de dados. Os dados são selecionados e disponibilizados para a aplicação que fez a solicitação.

A sintaxe geral do comando SELECT é

SELECT ***lista_de_seleção*** **FROM** ***expressão_de_tabela*** **[*****especificação_da_ordenação*****]**

O capítulo 7 do Manual PostgreSQL descreve em detalhes a lista de seleção, a expressão de tabela, e a especificação da ordenação. O tipo mais simples de consulta possui a forma:

```
SELECT * FROM aluno;
```

Supondo existir a tabela **aluno**, este comando traz todas as linhas e todas as colunas da referida tabela.

Para a atividade, você vai utilizar o comando SELECT, lista de seleção, expressão de tabela, as cláusulas FROM, WHERE, ORDER BY (ASC e DESC), os operadores de relação >, <, >=, <=, <> e =, os operadores lógicos AND, NOT e OR e o operador LIKE.

Considerando que a tabela filme foi criada e os dados armazenados, no pgAdmin, selecione a tabela filme e no menu **Tools**, escolha a opção **Query tool**. Feito isso, digite os comandos e avalie os resultados:

1. SELECT * FROM filme;
2. SELECT titulo_original, titulo_portugues FROM filme;
3. SELECT titulo_original, genero FROM filme WHERE genero = ‘Drama’;
4. SELECT titulo_original, genero FROM filme WHERE genero <> ‘Policial’;
5. SELECT titulo_original, titulo_portugues, ano FROM filme WHERE ano < 2000;
6. SELECT titulo_original, titulo_portugues, ano FROM filme WHERE genero = ‘Drama’ AND ano > 2000;
7. SELECT titulo_portugues, ano, genero FROM filme WHERE genero = ‘Drama’ OR ano > 2000;
8. SELECT * FROM filme ORDER BY titulo_portugues ASC; (o padrão é ASC)
9. SELECT * FROM filme ORDER BY titulo_portugues DESC;
10. SELECT * FROM filme ORDER BY genero, titulo_portugues;
11. SELECT * FROM filme ORDER BY genero, titulo_portugues, ano;
12. SELECT * FROM filme ORDER BY genero, ano, titulo_portugues;
13. SELECT titulo_original FROM filme WHERE titulo_original LIKE ‘%T%’; *(tenham T em qualquer parte do campo). Pode-se incluir: OR* LIKE *‘%t%’*
14. SELECT titulo_original FROM filme WHERE titulo_original LIKE ‘T%’; *(comecem com T)*
15. SELECT titulo_original FROM filme WHERE titulo_original LIKE ‘%T’; *(terminem com T)*

Criar a tabela **cine**

PK	Nome do campo	Tipo
X	id_cine	INTEGER, NOT NULL
	nome_fantasia	CHARACTER VARYING, LENGTH: 60
	endereco	CHARACTER VARYING, LENGTH: 70
	municipio	CHARACTER VARYING, LENGTH: 30
	estado	CHARACTER, LENGTH: 2
	faturamento	NUMERIC, LENGTH: 12, PRECISION: 2

Armazene 10 registros na tabela **cine**, no mínimo 10 registros, sendo:

id_cine	nome_fantasia	endereco	municipio	estado	faturamento
1	Multiplex	R. Henrique Savi, 15-55	Bauru	SP	956000.00
2	Cinemark	Av. 13 de maio, 334	Ribeirão Preto	SP	984000.00
3	Cine’n Fun	Rod. Marechal Rondon, 335	Bauru	SP	1000250.00
4	Cine Star	Av. Beira Mar, s/n	Florianópolis	SC	2653022.00
5	Cine Beluzzo	Av. Fuad Said, 544	Bariri	SP	350000.00
6	Cine Paulista	Av. Paulista, 5690	São Paulo	SP	2987654.00
7	Cine Rodia	R. Luiz Alfredo, 68	Campinas	SP	1233600.00
8	Martins Ltda	R. Amazonas, 9000	Blumenau	SC	1365852.00
9	Diversões & Cine	Av. Brigadeiro Luiz Antonio, 367	São Paulo	SP	896235.00
10	Rede Cine	R. Frei Carlos, 934	Belo Horizonte	MG	876235.00

Dando continuidade, agora com os dados da tabela **cine** armazenados, você vai experimentar as cláusulas GROUP BY, HAVING, BETWEEN e as funções agregadas AVG, COUNT, MAX, MIN, SUM. As funções agregadas são chamadas assim porque apresentam valores resumidos ou agrupados. Por exemplo, a função COUNT exibe a quantidade de linhas de uma tabela. Independentemente da quantidade de linha da tabela, o resultado será apenas uma linha com uma coluna e o valor resultante.

Digite os comandos e avalie os resultados:

1. Encontre todos os cinemas que são da cidade de Bauru

   SELECT * FROM cine WHERE municipio = 'BAURU';

2. Encontre todos os cinemas que são do estado de São Paulo e ordene por municipio

   SELECT * FROM cine WHERE estado = 'SP' ORDER BY municipio;

3. Encontre todos os cinemas que são do estado de Santa Catarina e tenham faturamento maior que 2.000.000

   SELECT * FROM cine WHERE estado = 'SC' AND faturamento > 20000000;

4. Selecione todos os cinemas que não são do estado de São Paulo e que tenham faturamento maior que 500.000,00. Apresente somente os atributos id_cine, estado, faturamento;

   SELECT id_cine, estado, faturamento FROM cine WHERE estado <> ‘SP’ AND faturamento > 500000;

5. Faça uma seleção que contenha os atributos nome_fantasia, municipio, estado de todos os cinemas ordenados pelo estado, município e faturamento

   SELECT nome_fantasia, municipio, estado FROM cine ORDER BY estado, municipio, faturamento;

6. Encontre todos os registros que estão situados em uma avenida (‘Av.’)

SELECT * FROM cine WHERE ENDERECO LIKE '%AV.%';

7. Encontre todos os registros que tenha no seu nome fantasia a palavra “Cine”

SELECT * FROM cine WHERE nome_fantasia LIKE '%cine%';

8. Mostre todos os registros ordenados por estado, município e nome fantasia.

SELECT * FROM cine ORDER BY estado, municipio, nome_fantasia;

9. Encontre todos os registros que tenham faturamento maior que 1000000.00 e menor que 1300000.00 inclusive

- **SELECT * FROM cine WHERE faturamento >= 1000000 AND faturamento <= 1300000;**
- **SELECT * FROM cine WHERE faturamento BETWEEN 1000000 AND 1300000; (na verdade ele está fazendo: faturamento >= 1000000 AND faturamento <= 1300000);**

10. Total do Faturamento de todos os cinemas

SELECT SUM(faturamento) FROM cine;

Obs.: o resultado é um único registro com o total: 13.302.848

11. Total do faturamento por Estado

SELECT estado, SUM(faturamento) FROM cine GROUP BY estado;

estado	SUM	
MG	876.235	(1)
SC	3.003.022	(2)
SP	9.423.591	(7)

12. Contar os municípios do estado de São Paulo

SELECT COUNT(municipio) FROM cine WHERE estado = 'SP';

<resultado = 7>

SELECT COUNT(DISTINCT municipio) FROM cine WHERE estado = 'SP';

<resultado = 5>

SELECT COUNT(*) AS total_cinemas_sp FROM cine WHERE estado = ' SP';

<resultado = 7 com o nome do campo>

Obs.: o AS é opcional

total_cinemas_sp funciona também com “Total Cinemas de São Paulo” – com as aspas pode colocar até caractere especial no nome do campo definido com AS.

13. Contar os municípios de cada estado

SELECT estado, COUNT(municipio) FROM cine GROUP BY estado;

14. Média do Faturamento por estado

SELECT estado, AVG(faturamento) FROM cine GROUP BY estado;

Obs.: o resultado são 3 registros com a média de faturamento de cada um dos estados

estado	AVG
MG	876.235
SC	1.501.511
SP	1.346.227,86

15. Soma do Faturamento somente do estado de São Paulo (definindo qual)

SELECT SUM(faturamento) FROM cine WHERE estado = ’SP’;

16. Média do Faturamento somente do estado de São Paulo (definindo qual)

SELECT AVG(faturamento) FROM cine WHERE estado = ’SP’;

17. Soma do Faturamento de cada município apresentando o nome do município

SELECT municipio, SUM(faturamento) FROM cine GROUP BY municipio;

18. Média do Faturamento somente do município de Bauru

- **SELECT AVG(faturamento) FROM cine WHERE municipio = 'BAURU';**
- **SELECT municipio, AVG(faturamento) FROM cine GROUP BY municipio HAVING municipio = 'BAURU';**

ou

- **SELECT municipio, AVG(faturamento) FROM cine WHERE municipio = 'BAURU' GROUP BY municipio;**

19. Soma do faturamento agrupado por município cuja soma seja maior que 500.000

- **SELECT municipio, SUM(faturamento) FROM cine GROUP BY municipio HAVING SUM(faturamento) > 500000;**

ou

- **SELECT municipio, SUM(faturamento) FROM cine WHERE SUM(faturamento) > 500000 GROUP BY municipio;**

20. Maior faturamento de cada estado

SELECT estado, MAX(faturamento) FROM cine GROUP BY estado;

21. Menor faturamento de cada estado

SELECT estado, MIN(faturamento) FROM cine GROUP BY estado;

Criar a tabela **ator**

PK	Nome do campo	Tipo
x	id_ator	INTEGER, NOT NULL
	nome	CHARACTER VARYING, LENGTH: 60
	nome_artistico	CHARACTER VARYING, LENGTH: 40
	naturalidade	CHARACTER VARYING, LENGTH: 30
	estado	CHARACTER VARYING, LENGTH: 20
	pais	CHARACTER VARYING, LENGTH: 20
	dt_nascimento	DATE

Armazene dados na tabela **ator**. Pesquise na internet e compartilhe com os colegas da turma, num total de pelo menos 20 atores.

id_ator	nome	nome_artistico	naturalidade	estado	pais	dt_nascimento
1						
2						
....						
20						

Parte III – Desenvolver o enunciado correspondente a uma cláusula (uma por grupo até o número (conforme a turma).

Algumas funções com datas interessantes:

Extrai o ano de uma data

- EXTRACT(YEAR FROM <nome_campo_data>)

Extrai o mês de uma data

- EXTRACT(MONTH FROM <nome_campo_data>)

Extrai o dia de uma data

- EXTRACT(DAY FROM <nome_campo_data>)

Subtração entre a data atual e outra data qualquer

- CURRENT_DATE – <nome_campo_data>

A tabela atuação é a primeira situação do uso da chave estrangeira. Uma chave estrangeira a coluna numa determinada tabela que se relaciona com a coluna que é chave primaria de outra tabela. Em resumo, duas tabelas ficam interligadas por meio da relação entre suas chaves primária e estrangeira. Na verdade, se trata de uma restrição. Para armazenar um valor na coluna definida como chave estrangeira, o mesmo valor já deve estar armazenado na coluna definida como chave primária de outra tabela.

Tabela: atuacao

PK	FK	Relacionamento com a tabela	Atributo	Domínio
X	X	filme	id_filme	integer
	X	ator	id_ator	integer
			cache	numeric 12,2

Obs.: A única chave primária (**PK** – **P**rimary **K**ey) é composta: id_filme + id_ator. São DUAS chaves estrangeiras (**FK** – **F**oreign **K**ey): id_filme, que referencia a tabela filme e id_ator, que referencia a tabela ator.

Alguns dados (sugestão) para a tabela atuacao

id_filme	id_ator	cache
1	3	156.000,00
1	4	258.000,00
1	5	65.232,00
2	3	9.875,23
3	5	589.745,23
4	2	789.000,00
4	1	45.800,00
5	8	200.000,00
6	9	328.000,00
7	3	9.875,23
8	5	589.745,23

59) Apresente os nomes artísticos dos atores que participaram do filme de ID igual a 1 (tabelas **atuacao** e **ator**)

```
SELECT ator.nome_artistico
```

```
FROM atuacao, ator
WHERE atuacao.id_ator = ator.id_ator AND atuacao.id_filme = 1;
```

Usando alias

```
SELECT ator.nome_artistico
FROM atuacao atua, ator
WHERE atua.id_ator = ator.id_ator AND atua.id_filme = 1;
```

60) Idem a 59 mostrando também o título original do filme que o ator trabalhou

61) Apresente os títulos originais dos filmes que o ator de ID igual a 2 atuou (tabelas **atuacao** e **filme**)

```
SELECT f.titulo_original, atua.id_ator
FROM atuacao atua, filme f
WHERE atua.id_filme = f.id_filme AND atua.id_ator = 1;
```

62) Apresente o total pago para os atores em cada filme

```
SELECT id_filme, SUM(valor_pago) AS TOTAL
FROM atuacao
GROUP BY id_filme;
```

63) Apresente o total pago aos atores do filme 1.

64) Apresente a média de valor que cada ator recebeu por suas atuações

65) Apresente os títulos dos filmes dos atores que nasceram nos Estados Unidos

66) Apresente os títulos dos filmes e os nomes dos atores que atuaram em filmes do ano de 2008

67) Apresente o nome do ator e o cachê recebido para atuações em filmes do ano de 2000 a 2011.

```
SELECT ator.nome_artistico, SUM(atuacao.cache)
FROM ator, atuacao, filmes
WHERE ator.id_ator = atuacao.id_ator AND
      filme.id_filme = atuacao.id_filme AND
```

```
                filme.ano BETWEEN 2000 AND 2010
    GROUP BY ator.nome_artistico;
```

68) Encontre todos atores que participaram de filmes após o ano 2000 e que a idade dos atores seja menor ou igual a 35 anos.

69) Apresente o nome do ator e os títulos dos filmes ordenados pela idade dos atores.

Tabela: exibicao

PK	FK	Relacionamento com a tabela	Atributo	Domínio
Sim	Sim	filme	id_filme	integer
	Sim	cine	id_cine	integer
			data_inicio	date
			data_fim	date
			qtde_ingressos	bigint

70) Mostre todos os filmes exibidos no cine de id = 1.

```
SELECT f.titulo_original
FROM filme f, exibicao ex
WHERE f.id_filme = ex.id_filme AND
            ex.id_cine = 1;
```

71) Apresente o nome do cinema e o título original dos filmes exibidos no ano de 2010

```
SELECT c.nome_fantasia, f.titulo_original
FROM filme f, exibicao ex, cine c
WHERE ex.id_cine = c.id_cine AND
            ex.id_filme = f.id_filme AND
            EXTRACT (YEAR FROM ex.dt_inicio) = 2010;
```

72) Mostre todos os filmes (título em português) exibidos nos cinemas da cidade de Campinas.

73) Mostre o total de ingressos vendidos e o nome do cinema para o filme de título original “Tropa de Elite 2”

74) Apresente o número de filmes já exibidos em cada cinema

75) Apresente todos os filmes e onde estão sendo exibidos na data de 01/01/2011 a 31/03/2011

76) Apresente o total arrecadado em cada cinema sabendo que o valor médio de cada ingresso é de R$ 7,00

77) Mostre a média de arrecadação em cada uma das cidades onde tem cinema cadastrado

78) Mostre o filme com maior número de ingressos vendidos em cada cinema

79) Apresente o filme de menor arrecadação em cada CIDADE

80) Mostre o nome artístico dos atores que já tiveram seus filmes exibidos na cidade de Bauru

81) Mostre os filmes em cartaz em todos os cinemas hoje

- ✓ **Atributo (campo) –** Cabeçalho da coluna
  Quantos atributos tem esta tabela?
  São 7: id_filme, titulo_original, tit_portugues, ano, genero, pais, diretor

- O tipo de dados que descreve os tipos de valores que podem aparecer em cada coluna é chamado de domínio

- ✓ **Dado –** conteúdo do atributo

  Dado do atributo titulo_portugues na tupla 4? O Poderoso Chefão

- ✓ **Tupla –** cada linha da nossa tabela, formada por um conjunto de atributos

  Quais os atributos da Tupla 1? E da tupla 2?

  Analogia com os tipos de dados das linguagens estruturadas, exemplo Pascal

```
Type filme = record
        id_filme : integer;
        titulo_original : string[60];
        titulo_portugues : string[60];
        ano : string[4];
        genero : string[20];
        end;
```

Este código define um novo registro chamado **filme** com cinco atributos.

Cada atributo tem um nome e um tipo a ele associado (**domínio**).

- ✓ **Tabela ou Relação –** conjunto de tuplas

  Quantos registros têm esta tabela? 6

- ✓ **Banco de Dados –** armazenamento físico das tabelas ou relações