POSTEGRES

ALUNOS: VICTOR FERRARI ANA MARIA

É um banco de dados relacional de código aberto que foi criado pela OSI (open source initiative)

É um banco de dados relacional de código aberto que foi criado pela OSI (open source initiative)

BANCO DE DADOS RELACIONAL

Um banco de dados relacional é um tipo de banco de dados que armazena e fornece acesso a pontos de dados relacionados entre si. Bancos de dados relacionais são baseados no modelo relacional, uma maneira intuitiva e direta de representar dados em tabelas.

BANCO DE DADOS OBJETO-RELACIONAL

Um banco de dados objeto relacional permite armazenamento de objetos, que reflete todo o paradigma de orientação a objetos, com herança, classes e afins. Porem mantem o padrão ANSI-SQL como linguagem de inserção e consulta.

Esse banco de dados é derivado do pacote POSTGRES

Esse banco de dados é derivado do pacote POSTGRES

escrito na universidade da california em barkeley, liderado pelo professor michael stonebraker e patrocinado pela Defense Advanced Research Projects Agency (DARPA), the Army Research Office (ARO), the National Science Foundation (NSF), e ESL, inc. sua implementação começou em 1986, passando a ser postgre95 e sendo substituido pelo nome postgresql em 1996.

Esse banco de dados é derivado do pacote POSTGRES

esse banco de dados também tem inúmeras interfaces nativas, podendo ser acessado por: odbc, jdbc, c, c++, php, perl, tcl, ecpg, python e ruby.

Esse banco de dados é derivado do pacote POSTGRES

o postgresql se mantem por doações e pelo patrocinio de empresas como fujitsu, hub.og, ntt group, entre outros.





além do seu uso pela comunidade, o postgresql tem sido usado tbm por grandes empresas:

- fuiitsu
- apple
- skype
- caixa economica
- detran ceara

BENEFÍCIOS DO USO DO POSTGRES



ALÉM DOS BENEFÍCIOS TRAZIDOS PRA OPERAÇÃO DE UMA EMPRESA, ESSE BANCO POSSUI AS SEGUINTES VANTAGE

recuperação automática após crash de sistema MVCC (controle de concorrência de multiversão)

Logging de transações

ALÉM DOS BENEFÍCIOS TRAZIDOS PRA OPERAÇÃO DE UMA EMPRESA, ESSE BANCO POSSUI AS SEGUINTES VANTAGE

Commit/ Rollback/ Checkpoints Triggers/Stored procedures

Constraints/ Foreign Keys

ALÉM DOS BENEFÍCIOS TRAZIDOS PRA OPERAÇÃO DE UMA EMPRESA, ESSE BANCO POSSUI AS SEGUINTES VANTAGENS

verificação de força de senha e comandos de gestão de permissões de banco de dados facilitados

ALÉM DOS BENEFÍCIOS TRAZIDOS PRA OPERAÇÃO DE UMA EMPRESA, ESSE BANCO POSSUI AS SEGUINTES VANTAGE

Gatilhos condicionais, gatilhos por colunas, ordenamento de agregações

Blocos de código de procedimento anônimos

Full text search melhor e integrada antes do mysql

Depois que

• MySQL foi comprado pela Oracle, foi criado um fork do projeto, chamado MariaDB, o qual no começo se comprometia a ser compatível com o MySQL, porém totalmente livre em termos de licença. Infelizmente ele já não é 100% compatível com o MySQL, o que talvez seja um problema para algumas empresas ou desenvolvedores (ou não).

Já o PostgreSQL sempre foi open-source e apesar de ter uma licença própria, ela é muito parecida com as licenças BSD e MIT.

O POSTGRES

TINGIONA ICAC ES

mprado pela do um fork do do MariaDB, o o se ser o MySQL, nte livre em ca. e já não é 100% o MySQL, o

um problema

npresas ou

es (ou não).

L sempre foi

apesar de ter

pria, ela é

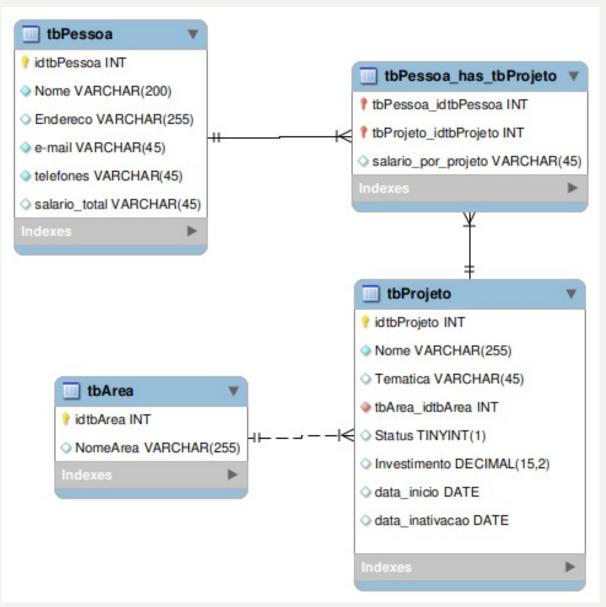
com as

Completa do MYSQL não é gratuita, enquanto o postgres oferta todas as suas funcionalidades de gratis

0 postgres

Tem uma maior quantidade de linguagens para transações

DATABASE UTILIZADA COMO BASE



CRIANDO DATABASE

[victorferrari@Victors-MacBook-Pro ~ % createdb exemplo

ENTRANDO NO DATABASE

```
[victorferrari@Victors-MacBook-Pro ~ % psql exemplo
psql (14.4)
Type "help" for help.
exemplo=#
```

INTERFACE PRINCIPAL POSTGRES

```
[victorferrari@Victors-MacBook-Pro ~ % psql postgres
[psql (14.4)
Type "help" for help.
```

postgres=# \list										
List of databases										
Name	Owner	Encoding	Collate	Ctype	Access privileges					
exemplo	 victorferrari	 UTF8	C	C	+ 					
postgres	victorferrari	UTF8	C	C						
template0	victorferrari 	UTF8 	C	C	=c/victorferrari victorferrari=CTc/victorferrari					
template1	victorferrari 	UTF8 	C	C	=c/victorferrari victorferrari=CTc/victorferrari					
(4 rows)										
(END)										

CRIANDO TABELAS

```
[exemplo=# create table tbPessoa (
[exemplo(# idtbPessoa int not null,
[exemplo(# nome varchar(200) not null,
[exemplo(# endereco varchar(255) null,
[exemplo(# email varchar(45) not null,
[exemplo(# telefone varchar(45) not null,
[exemplo(# salario_total int null);
```

ADICONANDO CHAVES PRIMARIAS E ESTRANGEIRAS

exemplo=# alter table tbProjeto add constraint pk_tbProjeto primary key (idtbProjeto);
ALTER TABLE

[exemplo=# alter table tbProjeto add constraint fk_Area foreign key (idtbArea) references tbArea (idtbArea); | ALTER TABLE

ADICIONANDO VALORES

MOSTRANDO AS TABELAS CRIADAS

```
[exemplo=# \dt
                     List of relations
 Schema
                    Name
                                     Type
                                                  Owner
 public
                                     table
                                              victorferrari
           tbarea
 public
                                     table
                                              victorferrari
           tbpessoa
 public
          tbpessoa_has_tbprojeto
                                              victorferrari
                                     table
 public
           tbprojeto
                                              victorferrari
                                      table
(4 rows)
```

DESCREVENDO UMA TABELA

Column	Туре	Collation	ble "public Nullable			Compression	Stats target	Description	
idtbprojeto	 integer		+ not null	+ 	 plain	+ 	+ 	+ 	
nome	character varying(255)		not null	1	extended	İ	İ	İ	
tematica	character varying(255)		ĺ		extended	ĺ	ĺ		
idtbarea	integer		not null		plain		<u> </u>	i	
status	integer		i		plain	İ	ì	İ	
investimento	integer		i		plain	İ	Ì	j	
data_inicio	date		i		plain	İ	ì	İ	
data_inativacao	date		i	1	plain	İ	Ì	İ	
Indexes:									
"pk_tbprojeto'	" PRIMARY KEY, btree (idtl	projeto)							
oreign-key const	raints:								
"fk_area" FOR	EIGN KEY (idtbarea) REFERI	NCES tbarea	(idtbarea)						
Referenced by:									
TABLE "tbpessoa_has_tbprojeto" CONSTRAINT "fk_projeto" FOREIGN KEY (idtbprojeto) REFERENCES tbprojeto(idtbprojeto)									

CONSULTANDO TODOS OS VALORES DE UMA TABELA

```
[exemplo=# select * from tbPessoa;
                                                                                       salario_total
 idtbpessoa |
                                                          email
                                endereco
                                                                            telefone |
                 nome
                                               emailpessoaum@email.com
              Pessoa Um
                             Rua Pessoa Um
                                                                             99999999
                                                                                                 1234
              Pessoa Dois l
                                               emailpessoadois@email.com
                            Rua Pessoa Dois
                                                                            99999999
                                                                                                 5678
(2 rows)
```

JOIN

	[exemplo=# select * from tbarea as A join tbprojeto as P on A.idtbarea = P.idtbarea order by P.investimento;										
\	idtbarea	nomearea	idtbprojeto	r	ome	tematica	idtbarea	status	investimento	data_inicio	data_inativacao
		++				ŀ	+	+	+		
	5	Matematica	5	Projeto	Calculos	Numeros	5	1	20	2021-01-01	2021-10-01
	3	Topologia	3	Projeto	Terra	Terra	3	0	200	2019-01-01	2022-01-01
	1	Biologia	1	Projeto	Tartarugas	Tartarugas	1	1	1200	2020-01-01	2021-01-01
	2	Fisica	2	Projeto	Espaço	Espaço	2	1	2200	2020-01-01	2022-01-01
	4	Oncologia	4	Projeto	Tumor	Cancer	4	1	11200	2011-01-01	2031-01-01
	(5 rows)			- N. 100 - 1100							

CONSULTA ALINHADA

```
[exemplo=# select PT.idtbpessoa from tbpessoa_has_tbprojeto as PT where PT.salario_por_projeto > 7000 group by PT.idtbpessoa;
idtbpessoa
------
5
4
(2 rows)
```

```
exemplo=# select P.nome from tbPessoa as P where P.idtbpessoa in (select PT.idtbpessoa from tbpessoa_has_tbprojeto as PT where PT.salario_por_projeto > 7000 group by PT.idtbpessoa);
nome
-------------
Pessoa Cinco
Pessoa Quatro
(2 rows)
```

INDEX

```
[exemplo=# create index on tbPessoa (nome);
CREATE INDEX
```

[exemplo=# create index idx_pessoa1 on tbPessoa using hash (nome); CREATE INDEX

```
exemplo=# create extension pg_trgm;
CREATE EXTENSION
exemplo=# create index idx_pessoa2 on tbPessoa using gin (nome gin_trgm_ops, email gin_trgm_ops);
CREATE INDEX
```

[exemplo=# create index idx_pessoa3 on tbPessoa using gist (nome gist_trgm_ops, email gist_trgm_ops); CREATE INDEX

PROCEDURE

```
exemplo=# \d+ accounts
                                                            Table "public.accounts"
                                  Collation | Nullable |
                   Type
 Column |
                                                                      Default
                                                                                            Storage | Compression | Stats target | Description
 id
                                                          generated by default as identity | plain
          integer
                                               not null |
          character varying(100)
                                               not null
                                                                                             extended
 name
balance | numeric(15,2)
                                               not null |
                                                                                             main
Indexes:
   "accounts_pkey" PRIMARY KEY, btree (id)
Access method: heap
```

```
[exemplo=# select * from accounts;
  id | name | balance
----+-----
  1 | Bob | 10000.00
  2 | Alice | 10000.00
(2 rows)
```

```
exemplo=# create or replace procedure transfer(sender int, receiver int, amount dec)
exemplo-# language plpgsql
exemplo-# as $$
exemplo$# begin
exemplo$# update accounts set balance = balance - amount where id = sender;
exemplo$# update accounts set balance = balance + amount where id = receiver;
exemplo$# commit;
exemplo$# commit;
exemplo$# end;
exemplo$# $$;
CREATE PROCEDURE
```

PROCEDURE

MVCC - MULTIVERSION CONCURENCY CONTROL

- Para instruções DML é utilizado o esquema MVCC.
- Esse protocolo depende do nivel de isolamento solicitado pela aplicação. Dentre esses niveis de isolamento e seriação, temos no postgres:
 - Leitura não repetitiva: transação lê o valor duas vezes, na segunda vez encontra um valor diferente embora a transação não tenha mudado o valor.
 - Leitura suja: transação lê valores escritos por outras transações que ainda não foram confirmadas.
 - Leitura fantasma: a transação reexecuta as linhas que satisfazem uma condição na consulta e descobre que essas linhas mudaram como o resultado de outra transação confirmada recentemente.

MVCC - MULTIVERSION CONCURENCY CONTROL

- No MVCC cada transação ve um instantaneo dos dados, consistindo em dados que foram confirmados no momento do inicio da transação. Esses instantaneo não é necessariamente igual ao estado atual dos dados.
- No MVCC leitores nunca bloqueiam escritores e virse-versa.
- A tupla somente é visivel para a transação se satisfazer essas duas condições:
 - A tupla feita pela Tb devera ser confirmada antes de ser executada pela Ta.
 - Atualizações feita pela tupla deverão ser executadas pela Tc caso: seja abortada; começou a ser executada apos a Ta ou estava em processo de leitura pela Ta
- O objetivo dessas condições é assegurar que cada transação acesse apenas dados que foram confirmados no momento em que a transação iniciou sua execução

MUITO OBRIGADO

