



www.sbVB.com.br

## Banco de Dados

- Versão 2.1 de 28 de Junho de 2005
- Por Sergio Barbosa Villas-Boas
  - www.sbVB.com.br
  - sbvb@sbvb.com.br



www.sbVB.com.br

## Categorias de bancos de dados

- Existem 4 categorias principais de banco de dados
  - Hierárquico
  - Em rede
  - Relacional (usa PK, FK, SQL)
  - Orientado a objetos

- Some examples
- Oracle
  - MS SQL server
  - DB2
  - Sybase
  - Informix
  - MySQL
  - PostgreSQL (\*)

(\*) gratuito

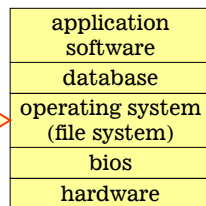


www.sbVB.com.br

## O que é um banco de dados?

- O banco de dados é um categoria de software que atua como uma camada para desenvolvimento de sistemas.
- Situa-se sobre o sistema operacional.
- SQL (Structured Query Language), is a database query language that was adopted as an industry standard in 1986.
- A relational database software is a tier over the file system. The application communicates with the database using SQL.
- The application is written in a computer programming language (such as C++), but SQL is not a programming language.

- Some examples
- MS Windows
  - Unix [linux(\*)]
  - Machintosh



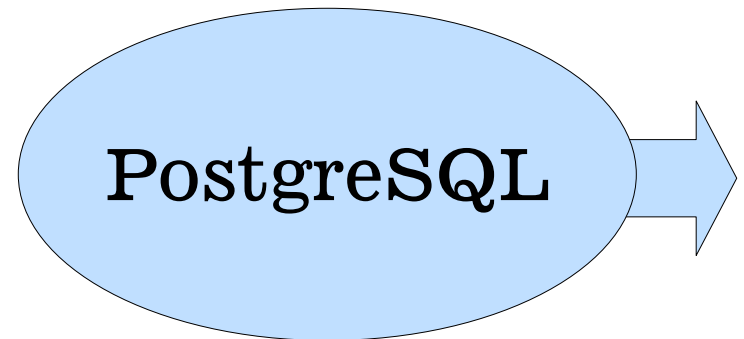
- Some examples
- Oracle
  - MS SQL server
  - DB2
  - Sybase
  - Informix
  - MySQL
  - PostgreSQL (\*)

(\*) gratuito



www.sbVB.com.br

## PostgreSQL





www.sbVB.com.br

## Breve história do PostgreSQL

- 1986 – início do desenvolvimento do POSTGRES, pelo Departamento de Ciência da Computação de Berkeley, na Universidade da Califórnia.
- 1989 – primeira versão liberada para alguns usuários, conhecida como “The implementation of POSTGRES”.
- 1990 – versão 2.
- 1991 – versão 3.
- 1993 – versão 4.2 (última versão no escopo do projeto de pesquisa de Berkeley).
- 1995 – A partir da iniciativa de Andrew Yu e Jolly Chen, versão PostgreSQL lançado como open-source, 100% ANSI C.
- 1996 – Projeto renomeado para “PostgreSQL”, com versão inicial 6.
- ? – PostgreSQL versão 7
- 2005 – versão 8, além de outras melhorias, versão Windows nativa. Introdução de tablespace.

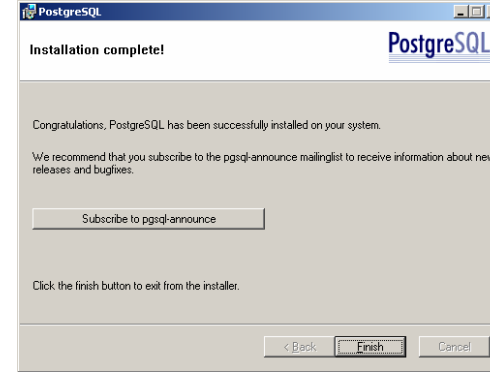
<http://www.postgresql.org/>

5



www.sbVB.com.br

## Instalando o PostgreSQL Versão Windows



↑

Administre o PostgreSQL usando o aplicativo GUI multiplataforma pgAdmin III (escrito em C++ e wxWidgets)

- Acrescente ao path o diretório C:\Program Files\PostgreSQL\8.0\bin

6



www.sbVB.com.br

## Operações básicas (console)

- Criar o usuário root no PG

```
[root]$ su postgres
[postgres]$ createuser root
Shall the new user be allowed to create databases? (y/n) y
Shall the new user be allowed to create more new users? (y/n) y
CREATE USER
```

- Criar o usuário <user> no PG

```
[root]$
[root]$ createuser -P sbvb
Enter password for new user:
Enter it again:
Shall the new user be allowed to create databases? (y/n) n
Shall the new user be allowed to create more new users? (y/n) n
CREATE USER
```

7



www.sbVB.com.br

## Operações básicas (console)

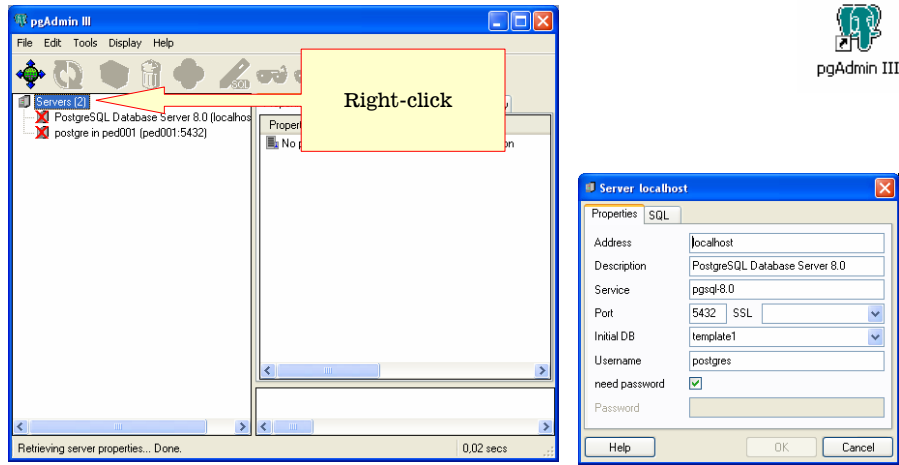
- Criar / apagar uma base de dados
  - \$ createdb db\_sbVB owner=sbvb
  - \$ dropdb db\_sbVB
- Entrar no console da base de dados, no usuário sbvb
  - \$ psql db\_sbVB
- Veja maiores detalhes no capítulo “Getting Started” do manual

8



www.sbVB.com.br

## Operações básicas (pgAdmin III)



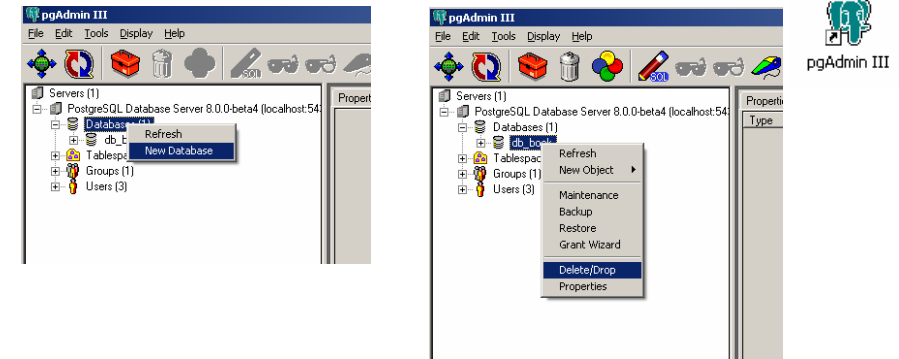
Criar uma conexão

9



www.sbVB.com.br

## Operações básicas (pgAdmin III (2))



Criar / apagar uma base de dados

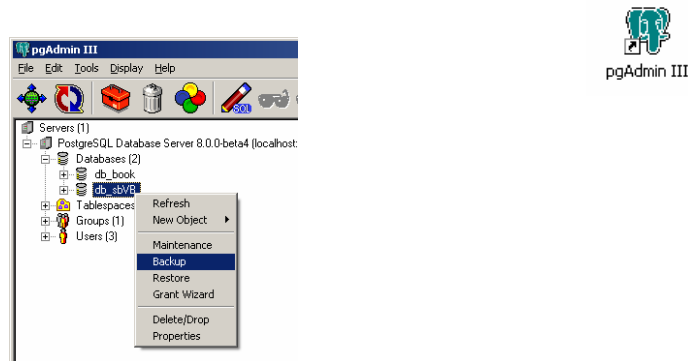
Apagar uma base de dados

10



www.sbVB.com.br

## Operações básicas (pgAdmin III (3))



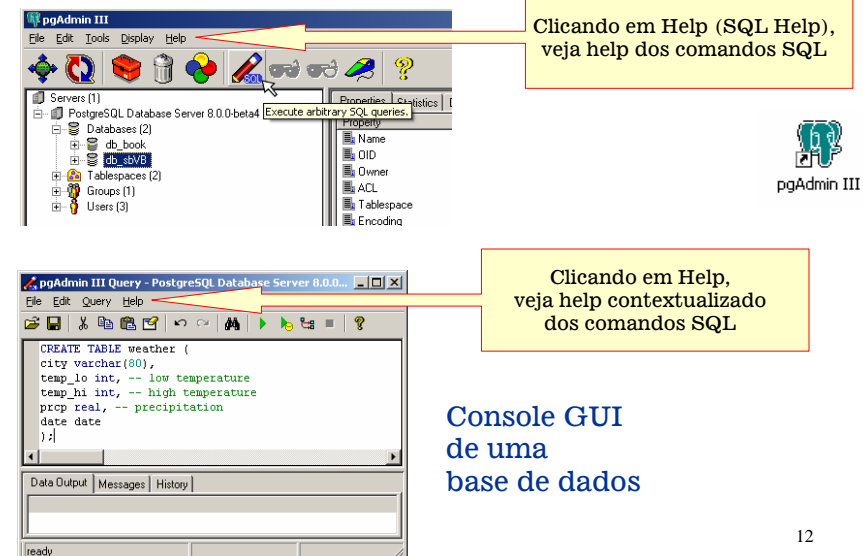
Backup / restore de uma base de dados

11



www.sbVB.com.br

## Operações básicas (pgAdmin III (4))



Clicando em Help (SQL Help), veja help dos comandos SQL

Clicando em Help, veja help contextualizado dos comandos SQL

Console GUI de uma base de dados

12



www.sbVB.com.br

## PostgreSQL v8 dá suporte a unicode!

- Não apenas dados, mas nomes de tabelas e de colunas podem ser em unicode.
- Seu aplicativo desenvolvido usando PostgreSQL poderá ser exportado para qualquer país do mundo, inclusive os que tem letras incompatíveis com o nosso alfabeto (por exemplo: Japonês).

13



www.sbVB.com.br

## PostgreSQL v8 dá suporte a unicode! (2)

The top screenshot shows a pgAdmin III Query window with the following SQL code:

```
CREATE TABLE 天気 (
町 varchar(80),
低温 int, -- low temperature
高温 int, -- high temperature
雨 real, -- precipitation
年月日 date
);
```

The bottom screenshot shows the 'Data Output' window with the following data:

oid	町	低温	高温	雨	年月日
1	千葉	22	25	2.33	2005.1.1
2	名古屋	21	28	3.43	2005.2.28
3	東京	21	27	4.22	2004.1.1



www.sbVB.com.br

## Tabelas SQL

- Os dados ficam em tabelas.

TableName
Col_1 (type)
Col_2 (type)
Col_3 (type)
Col_4 (type)

design view

TableName				
Col_1	Col_2	Col_3	Col_4	
data	data	data	data	
data	data	data	data	
data	data	data	data	
data	data	data	data	

data view

15



www.sbVB.com.br

## Tabelas SQL (2)

- Exemplo de tabela

Store
Name (varchar(50))
Sales (float)
LastReport (date)

design view

Store		
Name	Sales	LastReport
Rio de Janeiro	1200.0	26/08/2003
São Paulo	2000.0	20/08/2003
Paraná	1000.0	1/08/2003
Brasília	800.0	1/07/2003

data view

16



www.sbVB.com.br

## select

- Assuming that is data already stored in the database tables, one can get information about it by using the SQL command “select”.
- The syntax is
 

```
select column[,column] from table
```
- Your application can send to the database a select command, and it will produce an anonymous table in the database with the answer. Then, you can scan this anonymous table and use it for your purposes. Example:

```
select Name from Store
```

(anonymous)
Name
Rio de Janeiro
São Paulo
Paraná
Brasília

```
select Name,Sales from Store
```

(anonymous)	
Name	Sales
Rio de Janeiro	1200.0
São Paulo	2000.0
Paraná	1000.0
Brasília	800.0

17



www.sbVB.com.br

## Aprendendo mais sobre SQL

- Links
  - <http://www.1keydata.com/>
  - <http://www.sqlcourse.com/>
  - <http://www.sqlcourse2.com/>
  - <http://www.baycongroup.com/tocsql.htm>
- Help do PostgreSQL

18



www.sbVB.com.br

## Praticando funções básicas de SQL no PostgreSQL

- Veja os exemplos no capítulo 2 do manual.
- Pratique para obter familiaridade com os conceitos abaixo.
  - Criar tabelas
  - Popular tabelas
  - Query em tabelas
  - Join entre tabelas
  - Funções agregadas (max, min, like, etc.)
  - Updates de dados em tabelas
  - Deletes de dados em tabelas

19



www.sbVB.com.br

## Funções avançadas do SQL do PostgreSQL

- Views (an alias of a select)
- Foreign keys
- Array

20



www.sbVB.com.br

## Transações (transactions)

- É um conjunto de comandos que “faz o conjunto todo ou não faz nada”.
- Por exemplo: a transferência bancária de \$100 da conta da Alice para a conta do Bob. Atualiza-se também o total de uma loja (*branch*).

```
UPDATE accounts SET balance = balance - 100.00
WHERE name = 'Alice';

UPDATE branches SET balance = balance - 100.00
WHERE name = (SELECT branch_name FROM accounts WHERE name = 'Alice');

UPDATE accounts SET balance = balance + 100.00
WHERE name = 'Bob';

UPDATE branches SET balance = balance + 100.00
WHERE name = (SELECT branch_name FROM accounts WHERE name = 'Bob');
```

21



www.sbVB.com.br

## Características das transações

- Não faz sentido fazer uma transação incompleta, ou seja, uma transação é atômica.
- A transação deve garantidamente ser persistida em disco.
- A transação somente deve ser reconhecida após ser completada.
  - No caso de múltiplas transações concorrentes, não pode acontecer de uma transação considera como feita uma outra transação que não foi completada.
  - Uma transação “faz o conjunto todo ou não faz nada” tanto para o disco quanto para sua visibilidade por outras transações.
- Se, durante a transação, surge alguma informação que indique que se deve cancela-la (por exemplo: saldo negativo), pode-se cancelar todos os efeitos da transação com *rollback*.
- Uma única linha de SQL é tratada como uma transação.

22



www.sbVB.com.br

## Transações em SQL

```
BEGIN
```

Inicia transação

```
UPDATE accounts SET balance = balance - 100.00
WHERE name = 'Alice';

UPDATE branches SET balance = balance - 100.00
WHERE name = (SELECT branch_name FROM accounts WHERE name = 'Alice');

UPDATE accounts SET balance = balance + 100.00
WHERE name = 'Bob';

UPDATE branches SET balance = balance + 100.00
WHERE name = (SELECT branch_name FROM accounts WHERE name = 'Bob');

COMMIT;
```

Termina transação

23



www.sbVB.com.br

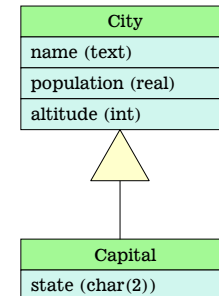
## Herança (inheritance)

```
CREATE TABLE City (
  name text,
  population real,
  altitude int -- (in mt)
);
```

```
CREATE TABLE Capital (
  state char(2)
) INHERITS (cities);
```

```
SELECT name, altitude
FROM cities
WHERE altitude > 500;
```

```
SELECT name, altitude
FROM ONLY cities
WHERE altitude > 500;
```



Todas as cidades, incluindo capitais, que tem altitude > 500

Todas as cidades, excluindo capitais, que tem altitude > 500



www.sbVB.com.br

## Blocking – Blocked

- Seja tabela “weather” abaixo, com alguns valores

```
CREATE TABLE weather (
  city varchar(80),
  temp_lo int, -- low temperature
  temp_hi int, -- high temperature
  prep real, -- precipitation
  date date
);
```

```
INSERT INTO weather (city, temp_lo, temp_hi, prep, date)
VALUES ('San Francisco', 43, 57, 0.0, '1994-11-29');
INSERT INTO weather (date, city, temp_hi, temp_lo)
VALUES ('1994-11-29', 'Hayward', 54, 37);
```

	oid	city varchar	temp_lo int4	temp_hi int4	prcp float4	date date
1	109915	San Francisco	43	57	0	1994-11-29
2	109916	Hayward	37	54		1994-11-29

2 rows.

25



www.sbVB.com.br

## Blocking – Blocked (2)

```
begin transaction;
update weather set city=city;
(commit transaction ainda não executado!)
```

no pgAdmin III  
Tools -> Server Status

PID	Database	Relation	User	TX	Mode	Granted	Start	Query
4748	test	109913			AccessSha...	Yes		<backend information not available>
4748	test	109913			RowExclus...	Yes		<backend information not available>
4748				97523	Exclusivel...	Yes		<backend information not available>

26



www.sbVB.com.br

## Blocking – Blocked (3)

(em uma janela diferente da primeira)  
begin transaction;  
update weather set city=city;  
(commit transaction ainda não executado!)

Note que o botão  
ficou vermelho

```
begin transaction;
update weather set city=city;
commit transaction;
```

Query is running. 12672 ms

27



www.sbVB.com.br

## Blocking – Blocked (4)

- Agora os dois processos (o blocking [4748] e o blocked [4688]) aparecem nessa lista

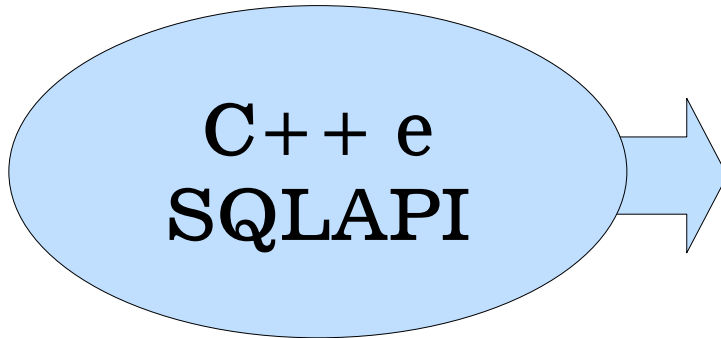
PID	Database	Relation	User	TX	Mode	Granted	Start	Query
4748								
4688								
4748								
4688								
4688								
4688								

28



www.sbVB.com.br

## C++ e SQLAPI



29



www.sbVB.com.br

## O que é SQLAPI?

- SQLAPI é uma biblioteca paga, em C++ multiplataforma, para acesso a banco de dados.
  - Embora seja paga, a SQLAPI não gera “gordura” no produto, pois a licença permite a distribuição gratuita da versão binária da SQLAPI para o cliente.
  - Testada em Windows e Unix (Linux)
  - Funciona com a maioria dos bancos de dados comerciais, baseados em SQL.
- Veja os detalhes na URL <http://www.sqlapi.com>

30



www.sbVB.com.br

## Definindo uma base de teste

- Nome da base: db\_bib
- Uma tabela: tb\_book

table “tb_book”			
title	author	editor	year
Microelectronics Circuits	Sedra / Smith	HRW Saunders	1991
Practical Unix Security	Simson Garfinkel	O'Reilly	1994

31



www.sbVB.com.br

## Criando a tabela de teste tb\_book no PostgreSQL

```
CREATE DATABASE db_bib
  WITH OWNER = postgres
  ENCODING = 'UNICODE'
  TABLESPACE = pg_default;
GRANT TEMPORARY ON DATABASE db_bib TO public;
GRANT ALL ON DATABASE db_bib TO postgres;
GRANT ALL ON DATABASE db_bib TO GROUP sbvb_grp;

create table tb_book ( author text, editor text, title text,
  year int4);

insert into tb_book (title,author,editor,year)
  values ('Microelectronics Circuits','Sedra / Smith',
  'HRW Saunders',1991);

insert into tb_book (title,author,editor,year)
  values ('Practical Unix Security','Simson Garfinkel',
  'O'Reilly',1994);
```

32



www.sbVB.com.br

## Criando a tabela de teste tb\_book no PostgreSQL (2)

	author text	editor text	title text	year int4
1	Sedra / Smith	HRW Saunders	Microelectronics Circuits	1991
2	Simson Garfinkel	O'Reilly	Practical Unix Security	1994

33



www.sbVB.com.br

## Criando a tabela de teste tb\_book no MySQL

```
mysql -u root -p
<root password>
create database db_bib;
grant all on db_bib.* to sbvb identified by 'pwd';
connect db_bib;
create table tb_book (title varchar(30), author varchar(30),
editor varchar(30), year int(4));
insert into tb_book (title,author,editor,year)
values ("Microelectronics Circuits","Sedra / Smith","HRW Saunders",1991);
insert into tb_book (title,author,editor,year)
values ("Practical Unix Security","Simson Garfinkel","O'Reilly",1994);
show tables;
-----+-----+-----+-----+
| Tables_in_db_bib |
+-----+-----+-----+-----+
| tb_book           |
+-----+-----+-----+-----+
select * from tb_book;
-----+-----+-----+-----+
| title              | author          | editor          | year |
+-----+-----+-----+-----+
| Microelectronics  | Sedra / Smith  | HRW Saunders   | 1991 |
| Practical Unix    | Simson         | O'Reilly       | 1994 |
+-----+-----+-----+-----+
```

34



www.sbVB.com.br

## Hello SQLAPI

```
#include <iostream>
#include <SQLAPI.h> // main SQLAPI++ header
int main()
{
    using namespace std;
    SAConnection con; // create connection object
    try
    {
        // con.Connect(<computerName>@<database name>,<user name>,
        // <password>,<client>);
        con.Connect("db_bib","sbvb","sbvb_pwd", SA_PostgreSQL_Client);
        cout << "We are connected!" << endl;
        con.Disconnect();
        cout << "We are disconnected!" << endl;
    }
    catch(SAException & x)
    {
        // print error message
        cout << (const char*)x.ErrText() << endl;
    }
    return 0;
}
```

Explicação do motivo de falha na conexão

We are connected!  
We are disconnected!

Login failed for user 'testLogin'.



www.sbVB.com.br

## Configurando o VC6 (SP5) para usar a SQLAPI

- Project Settings (ALT-F7), no tab C/C++

diretório onde está instalado o SQLAPI, /include

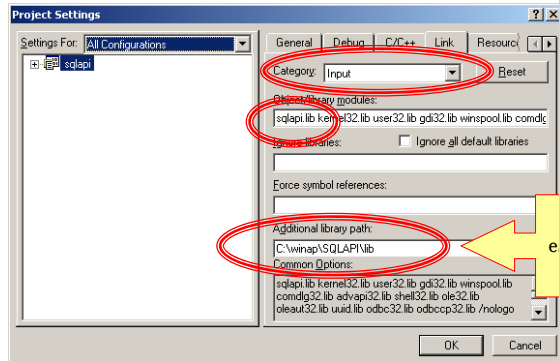
36



www.sbVB.com.br

## Configurando o VC6 (SP5) para usar a SQLAPI (2)

- Project Settings (ALT-F7), no tab Link



37



www.sbVB.com.br

## Fazendo uma query

```
#include <iostream>
#include <SQLAPI.h> // main SQLAPI++ header
using namespace std;
int main() {
    SAConnection con; // create connection object
    try {
        // connect to database
        con.Connect("db_bib", "sbvb", "sbvb_pwd", SA_PostgreSQL_Client);
        const char* SQL_command = "Select title,author,editor,year from tb_book";
        SACommand cmd(&con,SQL_command);
        cmd.Execute(); // Select from our test table
        unsigned rowCount = 1;
        // fetch results row by row and print results
        while(cmd.FetchNext()) {
            cout << "Row [" << rowCount++ << "] = " <<
                (const char*)cmd.Field("title").asString() << ";" <<
                (const char*)cmd.Field("author").asString() << ";" <<
                (const char*)cmd.Field("editor").asString() << ";" <<
                (unsigned)cmd.Field("year").asLong() <<
                endl;
        }
    }
    catch(SAException &x)
    { cout << "SA Exception: " << (const char*)x.ErrText() << endl; }
    return 0;
}
```

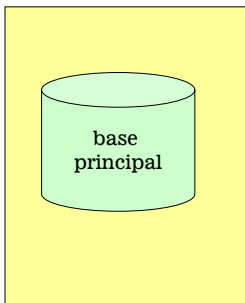
```
Row [1] = Microelectronics Circuits;Sedra / Smith;HRW Saunders;1991
Row [2] = Practical Unix Security;Simson Garfinkel;O'Reilly;1994
```

38



www.sbVB.com.br

## Fazendo backup no PostgreSQL



computador

39