

Banco de Dados e Relacionamento

Rais - Dia 4

Felipe Fernandes Akme-re Almeida

Agenda

- 1 Banco de Dados
- 2 Exemplos
- 3 Ferramenta - SQLiteMan
- 4 Tabelas
- 5 Cardinalidade
- 6 Relacionamentos
- 7 has_one
- 8 has_many
- 9 has_and_belongs_to_many

Conceito

- Coleção de dados armazenados em uma máquina.
- Base que pode ser acessada por softwares para consultas e atualizações.

Banco de Dados

Exemplos

PostgreSQL



ORACLE[®]
DATABASE



Ferramenta

SQLiteman - Ferramenta gráfica para gerenciar o Banco

The screenshot displays the SQLiteman application window. On the left, the 'Schema' pane shows a database named 'main' with a table 'avs' containing 7 columns: 'av', 'av_md5_hash', 'av_vendor', 'av_version', 'av_update', 'av_result', and 'av_timestamp'. Below the table, there are 3 indexes, 0 system indexes, and 0 triggers. Other database objects like 'connections', 'dcerpcbinds', 'dcerpcprequests', 'dcerpcservices', and 'downloads' are also listed.

The central 'Pragmas' pane shows a query editor with the following SQL code:

```

24 ORDER BY
25 strftime('%Y-%m-%d', connection_timestamp, 'unixepoch') ASC,
26 COUNT(*) DESC,
27 av_result ASC

```

Below the query editor, the execution status is shown: 'Duration: 26.005 seconds' and '* Col: 22 Row: 27/27'.

The 'Full View' tab is active, displaying a table with 7 rows and 6 columns. The columns are 'whenever', 'total', 'hits', 'cut', and 'av_result'. The data is as follows:

	whenever	total	hits	cut	av_result
1	2009-11-30	86877	85958	98.94	Net-Worm.Win32.Kido.ih
2	2009-11-30	86877	398	0.46	Trojan.Win32.Buzus.cqjn
3	2009-11-30	86877	274	0.32	Net-Worm.Win32.Kolab.ffa
4	2009-11-30	86877	273	0.31	-
5	2009-11-30	86877	207	0.24	Backdoor.Win32.Nepoe.mil
6	2009-11-30	86877	193	0.22	Trojan.Win32.Buzus.crh1
7	2009-11-30	86877	152	0.17	Trojan.Win32.Buzus.crgf

At the bottom, the 'Script Output' pane shows the query execution details: 'Query OK', 'Row(s) returned: 145', and the full SQL query used for the selection.

SQLite 3

Default Rails

- Banco de Dados Simples.
- Apenas para Estudo.
- Não deve ser usado em projetos robustos.
- Banco de dados localizado em um unico arquivo.

Banco de Dados

Conceitos Iniciais

- Bancos Possuem o que?
 - Usuários - Autenticação
 - Schemas - Organização
 - Tabelas - Armazenamento

SQLite

Tabela

- Tabelas
 - Estruturas de armazenamento para os dados.
 - Schemas - Organização.
 - Tabelas - Armazenamento.

Tabela

Tabela: Dividas.

CONTA	VALOR
Luz	100.50
Agua	50.55
Telefone	45.50

Index e Chave Primária

Tabela: Dividas.

ID	CONTA	VALOR
1	Luz	100.50
2	Agua	50.55
3	Telefone	45.50

Chave Estrangeira

Chave Estrangeira - Conceito

- Index Especial que liga duas Tabelas para melhor aproveitamento

Relacionamentos

Tipos

- 1×1
- $1 \times N$
- $N \times N$

Cardinalidade

1 e 1

ID	NOME	NASCIMENTO
1	Rodrigo Andrade Silva	01/09/1994
2	Felipe Almeida Zamora	05/10/1990

ID	IDENTIDADE	CPF	PES-ID
55	55997-65	555.999.777	2
56	56587-65	544.455.545	1

Cardinalidade

1 e 1

ID	NOME	NASCIMENTO	DOC-ID
1	Rodrigo Andrade Silva	01/09/1994	56
2	Felipe Almeida Zamora	05/10/1990	55

ID	IDENTIDADE	CPF
55	55997-65	555.999.777
56	56587-65	544.455.545

Relacionamentos 1xN

1xN

- Uma entidade(Tabela) se relaciona com outra de maneira desigual.
- 1 registro está relacionado a N outros registros.
- Chave estrangeira sempre do lado N

Relacionamento - 1xN

ID	TIME	SIGLA
10	Flamengo	CRF
11	Fluminense	FFC

ID	NOME	IDADE	TIME_ID
20	José	25	10
21	Mendonça	31	10
22	Daniel	26	11
23	Victor	28	11

Relacionamentos NxN

NxN

- N Registros de uma tabela possuem N de outra tabela.

Banco de Dados

Exemplos

ID	NOME
15	Rodrigo Andrade Silva
16	Felipe Almeida Zamora
17	Andre Luiz de Souza

ID	TURMA
30	CC1TA
31	BSINA
32	BECTA

ID_PROF	ID_TURMA
15	30
15	31
16	31
17	30
17	31
17	32

Prática e SQL

Prática

- Demonstração desses Conceitos usando o sqliteman.
- Linguagem SQL (Structure Query Language)

SQL

Prática

SQL - Criando Tabelas

```
CREATE TABLE Dividas(id INT PRIMARY KEY,  
                      conta VARCHAR(20),  
                      valor DECIMAL  
);
```

SQL

Prática

SQL - Crie a Tabela Pessoas

- Coluna id - INT -Chave Primária
- Coluna nome - VARCHAR
- Coluna nascimento - DATE

Cuidado

Comente ou delete as linhas de criação da tabela anterior para que a ferramenta não tente criar novamente a tabela Dividas: (Comentários em SQL são representados por '-- -')

```
-- CREATE TABLE dividas(id INT PRIMARY KEY,  
--                          conta VARCHAR(20),  
--                          valor DECIMAL  
-- );
```

SQL

Prática

SQL - Criando Tabelas

```
CREATE TABLE Pessoas(id INT PRIMARY KEY,  
                        nome VARCHAR,  
                        nascimento DATE  
);
```

SQL

Prática

SQL - Inserindo Dados na Tabela

```
INSERT INTO dividas VALUES (1,'LUZ', 100.52);
```

SQL

Prática

SQL - Criando a Tabela Intermediaria - professores_turmas

```
CREATE TABLE professores_turmas(  
    id_professor INT,  
    id_turma INT,  
    FOREIGN KEY(id_professor) REFERENCES professores(id),  
    FOREIGN KEY(id_turma) REFERENCES turmas(id)  
);
```


SQL

Prática

SQL - Criando a Chave Estrangeira - Documentos

```
CREATE TABLE Documentos(  
    id INT PRIMARY KEY,  
    identidade VARCHAR,  
    cpf VARCHAR,  
    pessoa_id INT,  
  
    FOREIGN KEY(pessoa_id) REFERENCES Pessoas(id)  
);
```

SQL

Prática

SQL - Criando a Chave Estrangeira - Documentos

```
CREATE TABLE Professores(id INT PRIMARY KEY, nome VARCHAR);  
CREATE TABLE Turmas(id INT PRIMARY KEY, nome_turma);
```

SQL

Prática

SQL - Criando a Tabela Intermediaria - professores_turmas

```
CREATE TABLE professores_turmas(  
    id_professor INT,  
    id_turma INT,  
    ... ,  
    ...  
);
```

SQL

Prática

SQL - Criando a Tabela Intermediaria - professores_turmas

```
CREATE TABLE professores_turmas(  
    id_professor INT,  
    id_turma INT,  
    FOREIGN KEY(id_professor) REFERENCES professores(id),  
    FOREIGN KEY(id_turma) REFERENCES turmas(id)  
);
```

Modelos MVC

Modelo

- Biblioteca - ActiveRecord::Base
- Estrutura - Convenções
- Funcionamento - ORM

CRUD

CRUD

- CREATE
- READ
- UPDATE
- DELETE

Relacionamentos

CRUD

- has_one - Pessoa e Documento
- has_many - Jogador e Time de Futebol
- belongs_to
- has_and_belongs_to_many - Professores e Turmas

Projeto Relacionamentos

1x1

```
$ rails new one_one  
$ rails g model Person name age:integer  
$ rails g model Document rg cpf
```


Relacionamentos

has_one e belongs_to

```
class Person < ActiveRecord::Base
  has_one :document
end

class Document < ActiveRecord::Base
  belongs_to :person
end
```

Não Se Esqueça da Migration

```
$ rake db:migrate
```

Rails Console

Testando os Modelos no Rails Cosole

```
$ rails c
```

Projeto Relacionamentos

1xn

```
$ rails new one_many  
$ rails g model Team name  
$ rails g model Player name age:integer
```

Relacionamentos

has_many e belongs_to

```
class Team < ActiveRecord::Base
  has_many :players
end
```

```
class Player < ActiveRecord::Base
  belongs_to :team
end
```

Não Se Esqueça da Migration

1x1

```
$ rake db:migrate
```

Rails Console

Testando os Modelos no Rails Cosole

```
$ rails c
```

Projeto Relacionamentos

1xn

```
$ rails new many_many  
$ rails g model Classroom name  
$ rails g model Teacher name
```


Relacionamentos

has_and_belongs_to_many

```
class Classroom < ActiveRecord::Base
  has_and_belongs_to_many :teachers
end
```

```
class Teacher < ActiveRecord::Base
  has_and_belongs_to_many :classrooms
end
```

Rails Console

Nova Tabela precisa ser criada

1xn

```
$ rails g migration create_classrooms_teachers
```

Relacionamentos

Nova Migration Essencial

```
def change
  create_table :classrooms_teachers do |t|
    t.integer :teacher_id
    t.integer :classroom_id
  end
end
```

Rails Console

Testando os Modelos no Rails Cosole

```
$ rails c
```