Técnicas de programação para Games
Aula06
Exemplo de Jogo e Bases SQL

Professor Mestre: Adilson Lopes Khouri

14 de outubro de 2018

Sumário

Cronograma

Aula	**Conteúdo**
12/04/2018	XP e banco de dados
17/04/2018	Introdução de estruturas de dados
24/04/2018	Arrays / Matrizes e Ordenação
26/04/2018	Recursão
03/05/2018	Lista Ligada
08/05/2018	Pilha, Fila
10/05/2018	Hash
15/05/2018	Árvore Binária
17/05/2018	Heap
22/05/2018	Grafos
24/05/2018	Prova

Jogo: Forca

- O jogo forca é muito simples, uma pessoa escolhe uma palavra aleatória e escreve em um papel. Em seguida é desenhada uma linha tracejada, em outra folha de papel, onde cada traço representa uma letra da palavra escrita.
- Outra pessoa deve descobrir qual é foi a palavra escrita chutando letras. Cada letra certa a primeira pessoa deve escrever a letra em cima do tracejado de sua posição correspondente.
- Para as letras erradas será desenhada uma parte de um boneco (representando um ser humano, como pernas, braços cabeça e tronco) em um equipamento de morte da idade média denominado: Forca.

Jogo: Forca

- Exemplo de jogo: "forca"

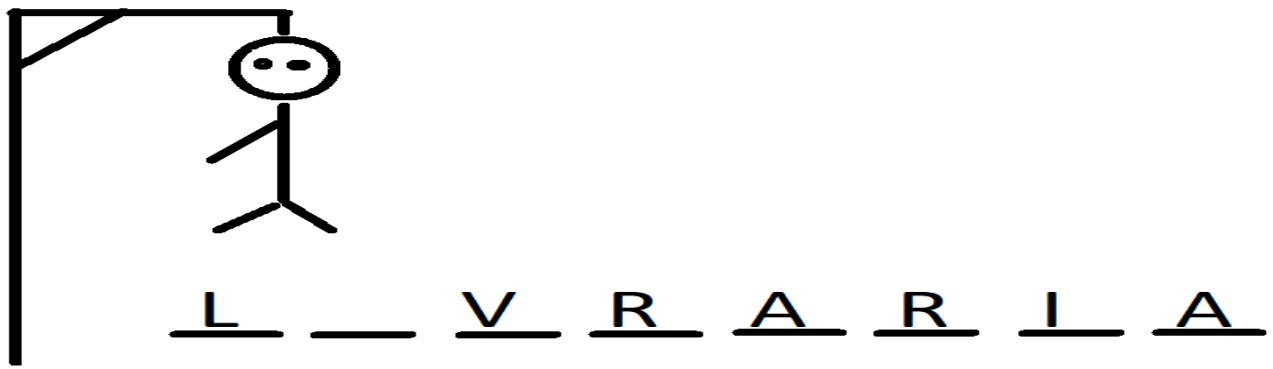

Figura: Jogo da forca

Jogo: Forca

- Nosso game vai precisar de um objeto palavra
- Nossa game vai precisar de um objeto Lista de palavras
- Nosso game precisa de condições de vitória, perda e continue jogando
- Nossa interface gráfica será o console

Jogo: Forca

- Mostrar o código para os alunos e explicar, mandar eles entenderem o código e produzirem um relatório sobre as partes do código.

Banco de dados relacional

- Antes de computadores a informação era armazenada em papel. Isso tornava o processo de análise, armazenamento e leitura de informação trabalhoso e demorado (Ficheiros são uma forma de indexação).
- Nos anos 60 surgiram os sistemas de arquivo (precisávamos conhecer as estruturas dos arquivos para pesquisar)
- Nos anos 70 surgiram os Sistemas Gerenciadores de Banco de Dados (SGBD) com o modelo de dados relacional
- Nos anos 80 o uso de banco de dados é difundido em meio acadêmico, surge o SQL.
- Anos 90, grandes empresas fornecedoras de SGBD, Microsoft,. Oracle, IBM, ... E software livre como MySql, Firebird, Postgresql

SQL

- Structured Query Language (SQL) linguagem para consulta de banco de dados relacionais padronizada entre diferentes bancos de dados.
- Vale ressaltar que nem todo banco de dados segue apenas o padrão SQL, cada distribuidor de ferramenta implementa algumas funcionalidades extras para facilitar o trabalho do DBA.

Banco de dados relacional

- Atualmente SGBD podem ser executados em simples computadores pessoais.
- Algumas definições:
- Dado: representação da informação: classe do herói, cor da roupa, tipo de arma
- Informação: Fato extraído de um conjunto de dados e.g. Guerreiro com nível de experiência 98, peso 105 Kg morador de Gondor pertencente a cavalaria real.

Banco de dados relacional

- Conjunto de tabelas relacionadas armazenadas em disco rígido (HD) em uma estrutura de dados eficiente para leitura e escrita.
- Garantem consistência dos dados, atomicidade de operações, isolamento, durabilidade (ACID)
- Permitem que se construa uma estrutura que evita redundância de informação, se corretamente normalizado.

Comandos SQL

- Vamos estudar os comandos para manipular dados em bancos de dados relacionais (select, create, update, delete, ...).
- Comandos de gerenciamento e definição de dados não serão avaliados (grant, revoke, index, ...)
- Vamos usar um banco de dados de exemplo em SQLite um banco de dados que não exige SGBD, tipicamente usado em dispositivos móveis (como celulares)

Comando	Descrição
SELECT	Seleciona colunas da tabela
INSERT	Insere registros na tabela
UPDATE	Atualiza valores das colunas da tabela
DELETE	Deleta linhas da tabela
CREATE	Cria tabelas
ALTER	Altera estrutura das tabelas
DROP	Deleta tabelas e outros objetos do SGBD
HAVING	Filtra linhas agrupadas
GROUP BY	Agrupa dados
WHERE	Filtra linhas não agrupadas

JOINS SQL

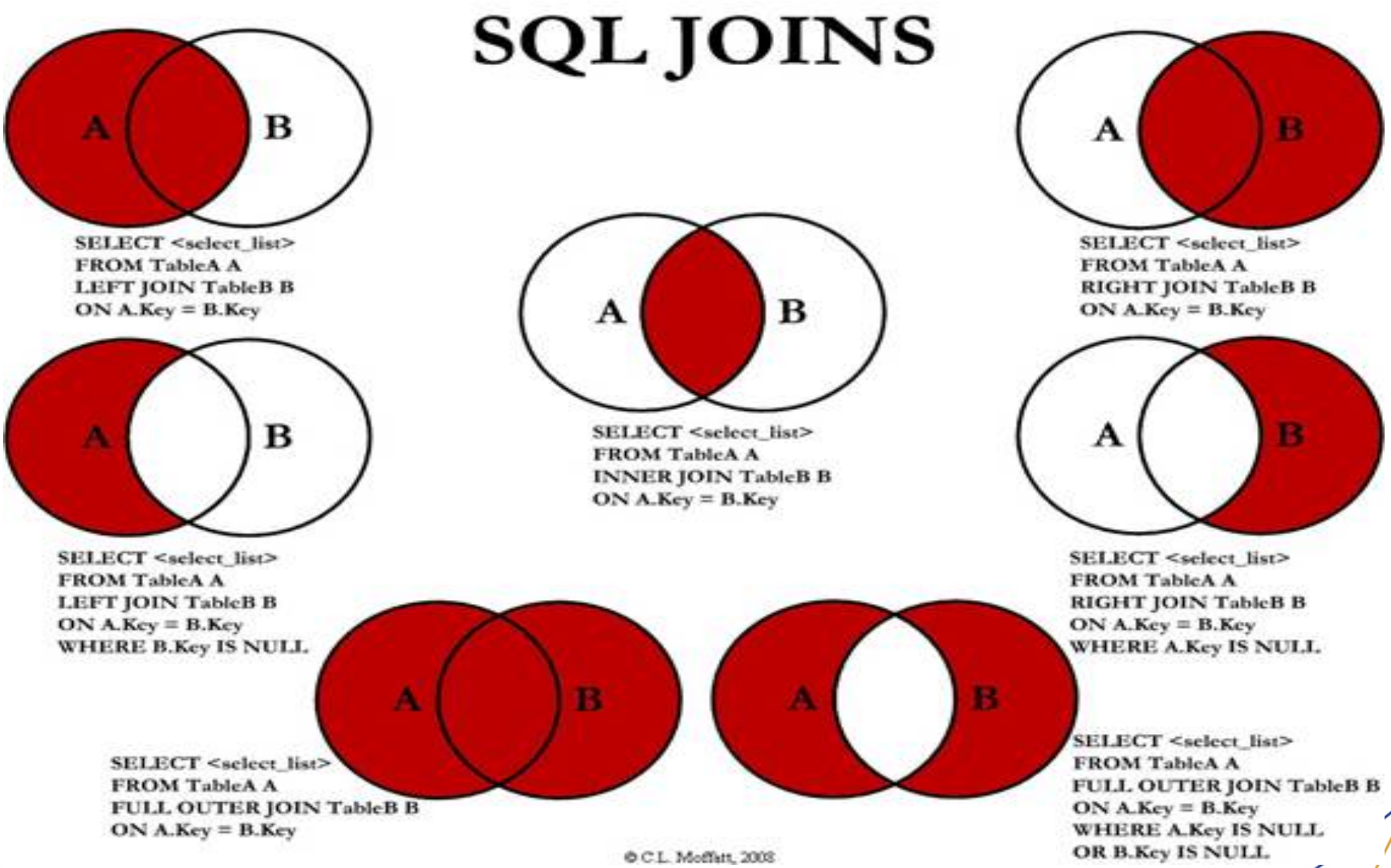

Figura: Joins do SQL

Base de Exemplo

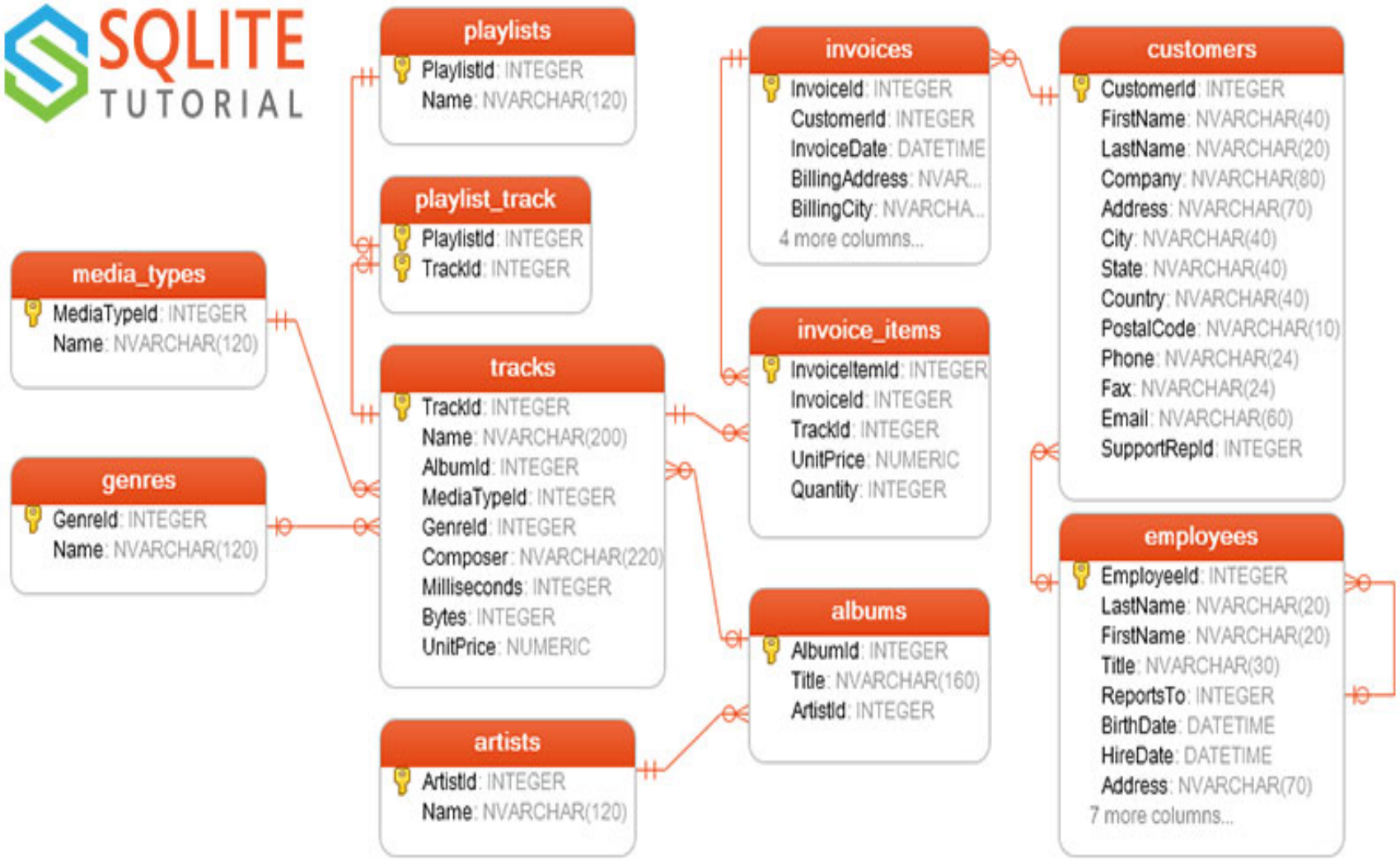

Figura: Base de Exemplo

Select

Comando para selecionar determinadas colunas de uma (ou mais) tabelas do banco de dados:

```
1  SELECT
2      trackid,
3      name,
4      albumid,
5      mediatypeid,
6      genreid,
7      composer,
8      milliseconds,
9      bytes,
10     unitprice
11 FROM tracks;
```

Select asterísco

Comando para selecionar todas as colunas de uma (ou mais) tabelas do banco de dados:

```
1 SELECT * FROM tracks;
```

Order by ASC

Comando para ordernar os registros (maior para o menor) de acordo com a coluna seguinte ao comando:

```
1  SELECT
2          name,
3          milliseconds,
4          albumid
5  FROM tracks
6  ORDER BY albumid ASC;
```

Order by DESC

Comando para ordernar os registros (menor para o maior) de acordo com a coluna seguinte ao comando:

```
1 SELECT
2     city
3 FROM customers
4 ORDER BY city desc;
```

Where

Comando para filtrar linhas:

```sql
SELECT
        name,
        milliseconds,
        bytes,
        albumid
FROM tracks
WHERE albumid = 1;
```

Like

Comando de expressão regular para filtrar linhas:

```
1 SELECT
2         trackid,
3         name
4 FROM tracks
5 WHERE name LIKE "Wild%"
```

In

Comando para determinar range de valores finitos, tipicamente usado em condições case ou where:

```
1  SELECT
2          trackid,
3          name,
4          mediatypeid
5  FROM tracks
6  WHERE mediatypeid IN (1, 2)
7  ORDER BY name ASC;
```

Inner Join

Comando para juntar tabelas, apenas registros que existem em ambas as tabelas:

```
1  SELECT
2          trackid,
3          name,
4          title
5  FROM tracks
6  INNER JOIN albums
7  ON albums.albumid = tracks.albumid;
```

Left Join

Comando para juntar tabelas, apenas registros que existem em ambas as tabelas ou na tabela da esquerda:

```
1  SELECT
2          artists.ArtistId,
3          albums.albumId
4  FROM artists
5  LEFT JOIN albums
6  ON albums.artistid = artists.artistid
7  ORDER BY albumid;
```

Count

Comando de agregação para sumarizar dados, nesse caso contá-los:

```
1  SELECT
2          albumid,
3          COUNT(trackid)
4  FROM tracks
5  GROUP BY albumid
6  ORDER BY COUNT(trackid) DESC;
```

Union

Comando para juntar dois selects empilhando as colunas de ambos:

```
1 SELECT firstname, lastname
2 FROM employees
3 UNION
4 SELECT firstname, lastname
5 FROM customers
6 ORDER BY firstname, lastname;
```

Having

Comando para filtrar resultados agrupados:

```
1  SELECT
2          albumid,
3          COUNT(trackid)
4  FROM tracks
5  GROUP BY albumid
6  HAVING albumid = 1;
```

Insert

Comando para inserir registros no banco de dados:

```
1 INSERT INTO
2 artists (name)
3 VALUES ("Buddy Rich"), ("Candido"), ("Charlie Byrd");
```

Update

Comando para atualizar registros do banco de dados:

```
1 UPDATE employees SET lastname = "Smith" WHERE employeeid = 3;
```

O que ocorre se esquecermos a cláusula where?

Delete

Comando para deletar registros do banco de dados:

```
1 DELETE FROM artists_backup WHERE artistid = 1;
```

O que ocorre se esquecermos a cláusula where?

Case

Comando para tomar decisões sobre valores, seria o equivalente ao if:

```
1  SELECT
2          customerid,
3          firstname,
4          lastname,
5          CASE country WHEN "USA"
6          THEN "Dosmetic"
7          ELSE "Foreign"
8          END CustomerGroup
9  FROM customers
10 ORDER BY LastName, FirstName;
```

Alter

Comando para alterar objetos do banco de dados , entre outros, nomes de tabelas:

```
1 ALTER TABLE devices RENAME TO equipment;
```

Exercícios

- Acessar o site SqliteTutorial e realizar os tutorias:
- Select
- Order by
- Where
- IN
- Like
- Todos os joins!
- Union
- Having
- Case
- Insert, Update, Delete
- Group by e SQLITE functions (AVG, MIN, MAX, SUM)

Dúvidas...

Alguma dúvida?

Contato

- E-mail: 0800*dirso@gmail.com* (alunos SENAC)
- E-mail: *adilson.khouri.usp@gmail.com*
- Phone: $+55119444-26191$
- Linkedin
- Lattes
- GitHub

Referências

[1] A. V. Aho, J. E. Hopcroft, and J. Ullman, *Data Structures and Algorithms*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1983.

[2] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change (2Nd Edition)*. Addison-Wesley Professional, 2004.

[3] Beck, *Test Driven Development: By Example*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[4] M. Fowler, *Refactoring: Improving the Design of Existing Code*. Boston, MA, USA: Addison-Wesley, 1999.

[5] Geeks. (2018) A computer science portal for geeks. [Online]. Available: https://www.geeksforgeeks.org