



UC1: Desenvolver algoritmos e estrutura de dados
para aplicativos móveis

SQL

Docente: Felipe B. Benutti

Visão Geral de um Projeto (etapas)

1. Modelagem com ER (**Conceitual**)

- Modelagem de entidades (e seus atributos), relacionamentos (e suas cardinalidades)
- Proposta de solução.

2. Transformação entre Modelos (**Lógico**)

- Transformação de ER para Relacional

3. Definição do BD em SQL (**Físico**)

- Criação de Scripts DDL;

Recordando

- Existem diversos SGBDs relacionais no mercado atualmente. Alguns exemplos dos mais famosos:
 - **SQLite; (trabalharemos com este!)**
 - **MySQL;**
 - **SQL Server;**
 - **PostgreSQL;**
 - **Oracle;**
 - **Firebird;**
 - **Entre outros;**

SQL

- Notou que a maioria dos nomes de SGBDs tem uma sigla SQL? Mas o que é essa tal de SQL?
- **SQL** (Structured Query Language – Linguagem de Consultas Estruturada) é a linguagem padrão universal para manipular bancos de dados relacionais através dos SGBDs.
- É utilizada para interagir com o SGBD e executar várias tarefas como :
 - **inserir e alterar registros;**
 - **consultar dados;**
 - **criar bancos de dados;**
 - **gerenciar usuários;**
 - **controlar transações;**
 - **entre muitos outros comandos;**

SQL

- O SQL é dividido em 4 agrupamentos;
- Eles são de acordo com os **tipos de operação** a serem executadas;
 - **DDL** (Data Definition Language, ou Linguagem de Definição de Dados em português);
 - **DML** (Data Manipulation Language, ou Linguagem de Manipulação de Dados em português);
 - *DCL (Data Control Language, ou Linguagem de Controle de Dados em português);*
 - *DTL (Data Transaction Language, ou Linguagem de Transação de Dados em português)*
- *Toda a documentação SQL do SQLite encontra-se aqui:*
 - <https://www.sqlite.org/lang.html>

SQL

- **DDL** (Linguagem de Definição de Dados) é o subconjunto da **SQL** utilizado para gerenciar a estrutura do banco de dados. Com a DDL podemos criar, alterar e remover objetos (tabelas, visões, funções, etc.) no banco de dados.
- Os comandos deste subconjunto são:

SQL

- **CREATE:** utilizado para criar objetos no banco de dados.
 - Exemplo (criar uma tabela): CREATE TABLE CLIENTE (ID INTEGER PRIMARY KEY, NOME VARCHAR(50));
- **ALTER:** utilizado para alterar a estrutura de um objeto.
 - Exemplo (adicionar uma coluna em uma tabela existente): ALTER TABLE CLIENTE ADD SEXO CHAR(1);
- **DROP:** utilizado para remover um objeto do banco de dados.
 - Exemplo (remover uma tabela): DROP TABLE CLIENTE;

SQL

- **DML** (Linguagem de Manipulação de Dados) é o subconjunto mais utilizado da linguagem **SQL**, pois é através da DML que operamos sobre os dados dos bancos de dados com instruções de inserção, atualização, exclusão e consulta de informações.
- Os comandos SQL desse subconjunto são:

SQL

- **INSERT:** utilizado para inserir registros (tuplas), em uma tabela.
 - Exemplo: INSERT into CLIENTE(ID, NOME) values(1,'José');
- **UPDATE:** utilizado para alterar valores de uma ou mais linhas (tuplas) de uma tabela.
 - Exemplo: UPDATE CLIENTE set NOME = 'João' WHERE ID = 1;

SQL

- **DELETE:** utilizado para excluir um ou mais registros (tupla) de uma tabela.
 - Exemplo: DELETE FROM CLIENTE WHERE ID = 1;
- **SELECT:** O principal comando da SQL, o comando select é utilizado para efetuar consultas no banco de dados.
 - Exemplo: SELECT ID, NOME FROM CLIENTE;

SQL

- SQLite:
 - Esse SGBD não requer instalação. Como ele é um SGBD baseado em um sistema de arquivos (assim como o Microsoft Access), ele armazena todo o BD em um único arquivo.
 - Quem se encarrega do gerenciamento de dados é a própria aplicação/software.
 - Devido à isso, ele é o mais indicado para se trabalhar com dispositivos móveis, os quais demandam acesso rápido.

SQL

- SQLite Studio:
 - Esse é o programa que será utilizado para gerenciar bases de dados, ou seja, realizar todas as operações relacionadas à bancos de dados SQLite.
 - Ele é utilizado como uma interface gráfica para conexão aos bancos de dados.

Definição do BD em SQL - DDL

DDL – Data Definition Language
(linguagem de definição de dados)

- Linguagem de computador utilizada para definição da estruturas presentes em um banco de dados.
- Estruturas a Definir:
 - Tabelas
 - Colunas
 - Linhas
 - Índices
- Nos SGBDs atuais se caracteriza como um subconjunto da SQL.
- Comandos:
 - CREATE TABLE;
 - DROP TABLE;
 - RENAME TABLE;
- Todo comando deve ser seguido de ponto-e-vírgula (;)

Criar uma Tabela

- CREATE TABLE **nome_tabela** (colunas e seus tipos);

- Exemplo:

```
CREATE TABLE aluno (  
    idaluno INTEGER PRIMARY KEY NOT NULL AUTOINCREMENT,  
    nome varchar(100) NOT NULL,  
    idade INTEGER NOT NULL  
);
```

Eliminar uma Tabela

- DROP TABLE [IF EXISTS] <nome_tabela>;
- Exemplo:
 - DROP TABLE IF EXISTS ator;

Renomear uma Tabela

- `RENAME TABLE <nome_tabela> TO <novo_nome_tabela>;`
- Exemplo:
 - `RENAME TABLE ator TO novo_ator;`

Tipos de Campos – SQLite

- Existem 4 tipos de campos que podem ser criados no SQLite. São eles:
 - Tipos Numéricos:
 - INTEGER e REAL;
 - Tipo de Texto:
 - TEXT;
 - Tipo binário (para guardar arquivos):
 - BLOB (Binary Large Object);
- Referência:
 - <https://www.sqlite.org/datatype3.html>

Especificações adicionais à campos (restrições)

- Os campos do banco de dados podem conter outras especificações além do tipo. Abaixo estão listadas algumas delas:
 - NULL;
 - NOT NULL;
 - AUTOINCREMENT;
 - DEFAULT;
- Referência:
 - https://www.tutorialspoint.com/sqlite/sqlite_constraints.htm

Chave Primária

- Criação de uma chave primária em uma tabela dessa forma:

```
CREATE TABLE curso(  
    idcurso INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
    nome VARCHAR(100) NOT NULL,  
    area VARCHAR(90) NOT NULL,  
    periodo VARCHAR(50) NOT NULL,  
    sala VARCHAR(10) NOT NULL,  
);
```

Chave Primária

- Ou dessa forma:

```
CREATE TABLE curso(  
    idcurso INTEGER NOT NULL,  
    nome VARCHAR(100) NOT NULL,  
    area VARCHAR(90) NOT NULL,  
    periodo VARCHAR(50) NOT NULL,  
    sala VARCHAR(10) NOT NULL,  
    PRIMARY KEY (idcurso)  
);
```

Obs.: desta maneira não pode ser usado o AUTOINCREMENT

Chave Estrangeira

- Criação de chave estrangeira em uma tabela:

```
CREATE TABLE aluno(  
    ra INT NOT NULL,  
    nome VARCHAR(100) NOT NULL,  
    data_nascimento DATE NOT NULL,  
    endereco_rua VARCHAR(100) NOT NULL,  
    endereco_numero INT,  
    idcurso INT NOT NULL,  
    PRIMARY KEY (ra),  
    FOREIGN KEY (idcurso) REFERENCES curso(idcurso)  
);
```

Referência: <https://www.sqlite.org/foreignkeys.html>

Prática

- Iremos agora criar um banco de dados e sua estrutura de tabelas utilizando o SQLite Studio.

Prática - Exercícios

- Criar os scripts de bancos de dados a partir dos modelos lógicos criados no exercícios anterior:
 1. Escritório de advocacia (clientes);
 2. Loja de materiais de construção (estoque);
 3. Rede Social (Instagram);

Definição do BD em SQL - DML

DML – Data Manipulation Language (linguagem de manipulação de dados)

- Linguagem utilizada para inserir, atualizar e eliminar dados em um banco de dados relacional.
- Subconjunto do SQL
- Linguagem declarativa
 - Indica “o que” se deseja fazer e não “como”.
- Comandos:
 - INSERT (**C**reate);
 - SELECT (**R**ead);
 - UPDATE;
 - DELETE;
- Obs: Esses três comandos são chamados também de CRUD. Um sistema de controle de informações deve possuir um CRUD.
- Todo comando deve ser seguido de ponto-e-vírgula (;)

Inserir Dados em uma Tabela

```
INSERT INTO NomeTabela  
    (NomeColuna,...)  
VALUES  
    (valores a serem inseridos,...);
```

- Exemplo:

```
INSERT INTO aluno  
    (Nome, Idade)  
VALUES  
    ('Fulano da Costa', 37);
```

Consultar todos os Dados de uma Tabela

- `SELECT * FROM NomeTabela;`
 - O asterisco indica que **todos** os campos da tabela serão selecionados.
- Exemplo:
`SELECT * FROM aluno;`
- Este comando apresenta muito mais particularidades, sendo um dos mais completos e complexos do SQL, pois trabalha com o conceito de conjuntos da matemática.

Atualizar Dados de Uma Tabela

- UPDATE nomeTabela SET <nome_coluna1>=expr1
[,<nome_coluna2>=expr2 ...] [WHERE <definição_where>]
- Atualizando um conjunto de linhas de uma tabela...
 - UPDATE aluno
SET Idade = 27
WHERE Nome = 'Fulano da Costa';
- Atualizando TODAS as linhas de uma tabela...
 - UPDATE aluno
SET Idade = 27;

Eliminar Linhas de uma Tabela

- DELETE FROM <table_name> [WHERE <definição_where>]
- Eliminando um conjunto de linhas de uma tabela...
 - DELETE FROM aluno
 - WHERE Nome = 'Fulano da Costa';
- Eliminando TODAS as linhas de uma tabela...
 - DELETE FROM aluno;

Prática - Exercícios

- Criar **ao menos um script** SQL de CRUD para **um dos bancos** criados no exercício anterior.
- Ou seja, fazer:
 - INSERT;
 - SELECT;
 - UPDATE;
 - DELETE;
- Os dados deverão ser fictícios (inventados).

Mais referências:

- Mais sobre SQLite:
 - <https://www.tutorialspoint.com/sqlite/index.htm>
- Diferenças para outros bancos:
 - <https://www.quora.com/What-are-the-differences-between-SQLite-and-MySQL-Are-they-both-the-same-company>
 - <https://www.devmedia.com.br/artigo-da-sql-magazine-41-comparando-firebird-mysql-postgresql-e-sqlite/7009>
 - <https://www.quora.com/Why-do-some-people-prefer-SQLite-over-MySQL>
- Ranking de Bancos de Dados usados no mundo:
 - <https://db-engines.com/en/ranking>