Resumo aula anterior:

Criamos o projeto

Criamos a classe

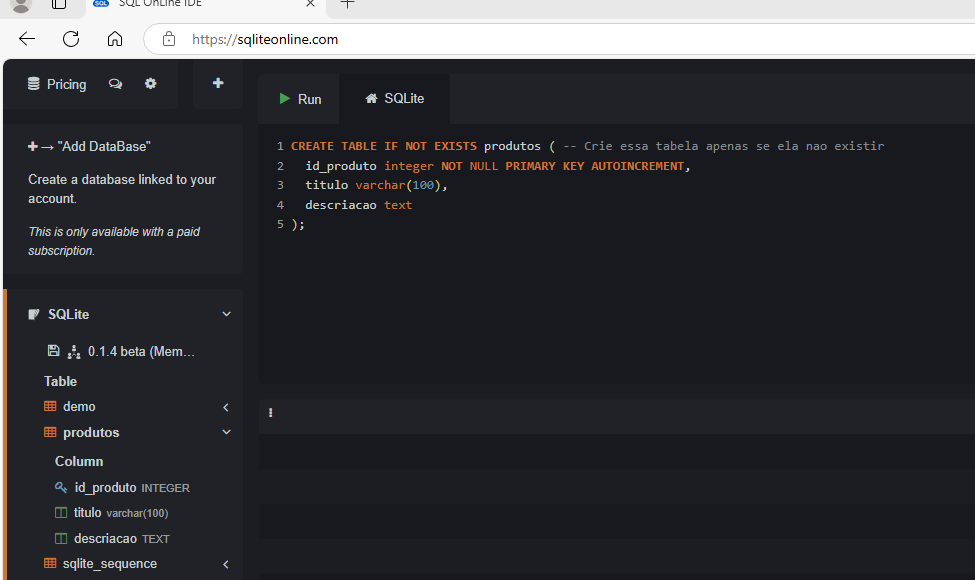
Implementamos os métodos e entendemos um pouco sobre ele

Aula agora:

# Criando tabela:

Primeiro vamos escrever nossos comandos SQL no site online para evitar erros e depois vamos para o android Studio

[SQL OnLine IDE](https://sqliteonline.com/)



Ai criamos nossa tabela e repara a esquerda que podemos ver nossa tabela

Também repare que não pode ser int deve ser integer

Ficando dessa forma

CREATE TABLE IF NOT EXISTS produtos ( -- Crie essa tabela apenas se ela nao existir

id\_produto integer NOT NULL PRIMARY KEY AUTOINCREMENT,

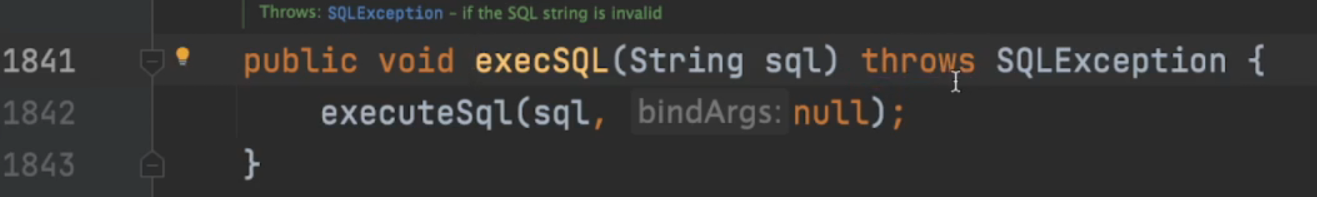
titulo varchar(100),

descriacao text

);

# Adicionando o comando no método

override fun onCreate(db: SQLiteDatabase?) {  
 */\*\* Esse metodo é chamado apenas uma vez  
 \* Quando o usuario instala meu aplicativo o metodo é chamado para criar o banco de dados  
 \* Se o usuario remover o app todos os dados é perdido.  
 \*  
 \* Aqui é um metodo importante, aqui é onde vamos criar as tabelas para que o usuario salve os dados  
 \*/* val codigoSql = "CREATE TABLE IF NOT EXISTS produtos (" +  
 "id\_produto integer NOT NULL PRIMARY KEY AUTOINCREMENT," +  
 "titulo varchar(100)," +  
 "descriacao text" +  
 ");"  
  
  
  
 try {  
 db?.execSQL(codigoSql) // Aqui ele recebe um codigo SQL e faz a execução  
 Log.i("info\_db", "Sucesso ao criar tabela")  
 } catch (e: Exception){  
 //e.message // aqui vamos pegar a mensagem  
 e.printStackTrace() // aqui vamos exibir no log o erro  
 Log.i("info\_db", "Erro ao criar tabela")  
  
 }



Quando tiver métodos que pode enviar uma exceção de volta devemos fazer dessa forma

try {  
 db?.execSQL(codigoSql) // Aqui ele recebe um codigo SQL e faz a execução  
 Log.i("info\_db", "Sucesso ao criar tabela")  
} catch (e: Exception){  
 //e.message // aqui vamos pegar a mensagem  
 e.printStackTrace() // aqui vamos exibir no log o erro  
 Log.i("info\_db", "Erro ao criar tabela")  
  
}

# Inserindo produtos no banco

Como pode ver primeiro fazemos o carregamento da classe que criamos

Depois chamamos a variável que é a classe que criamos com o writableDatabase.excelSQL passando a instrução do que queremos.

class MainActivity : AppCompatActivity() {  
  
 private val bancoDeDados by *lazy* **{** DatabaseHelper(this)  
 **}** override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *enableEdgeToEdge*()  
 setContentView(R.layout.*activity\_main*)  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->** val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
 insets  
 **}** try {  
 bancoDeDados.*writableDatabase*.execSQL(  
 "INSERT INTO produtos (id\_produto, titulo, descriacao) VALUES (null, 'Iphone 15', 'Descrição...');"  
 )  
   
 Log.i("info\_db", "Produto cadastrado com sucesso.")  
 }catch (e: Exception){  
 e.printStackTrace()  
 }  
 }  
}

# LOGS DO BANCO DE DADOS

Após executar o app uma vez recebemos os logs, após mudar a versão recebemos também outro log

A versão era 1

Quando mudamos para versão 2 ele chamou o método onUpgrade automaticamente

2025-01-25 16:16:18.964 21992-21992 info\_db com.allephnogueira.sqliteandroid I Executou o onCreate

2025-01-25 16:16:18.964 21992-21992 info\_db com.allephnogueira.sqliteandroid I Sucesso ao criar tabela

2025-01-25 16:16:18.979 21992-21992 info\_db com.allephnogueira.sqliteandroid I Produto cadastrado com sucesso.

---------------------------- PROCESS ENDED (21992) for package com.allephnogueira.sqliteandroid ----------------------------

---------------------------- PROCESS STARTED (22060) for package com.allephnogueira.sqliteandroid ----------------------------

2025-01-25 16:17:21.447 22060-22060 info\_db com.allephnogueira.sqliteandroid I Executou o onUpgrade