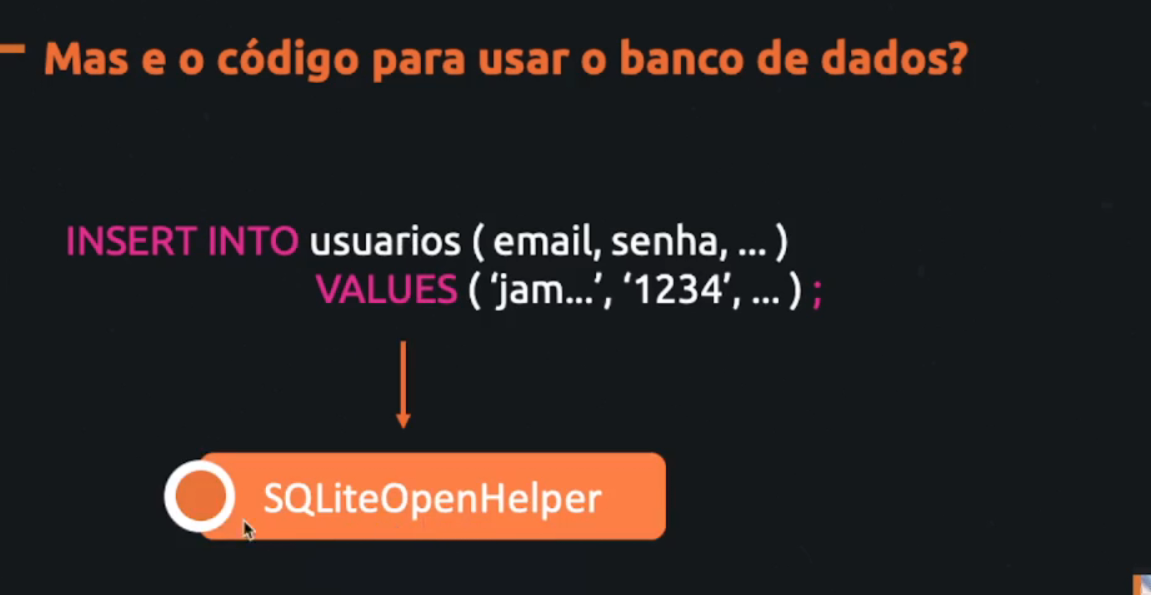
Vamos utilizar os mesmos códigos que aprendemos com MySQL

Select, drop, update...

E para usar no android vamos utilizar junto com a classe

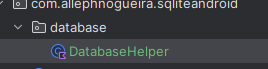


# 1 Vamos começar organizando o projeto.



Criamos esse pacote para organizar as nossas classes

E ai vamos criar essa classe



Helper = ajudante / auxiliar

Criamos essa classe para poder ajudar a gente.

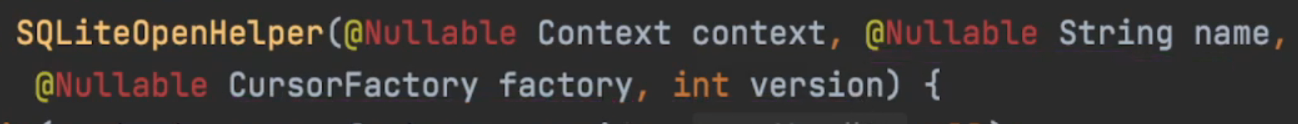
Essa classe vai herda os dados de SQLiteOpenHelper

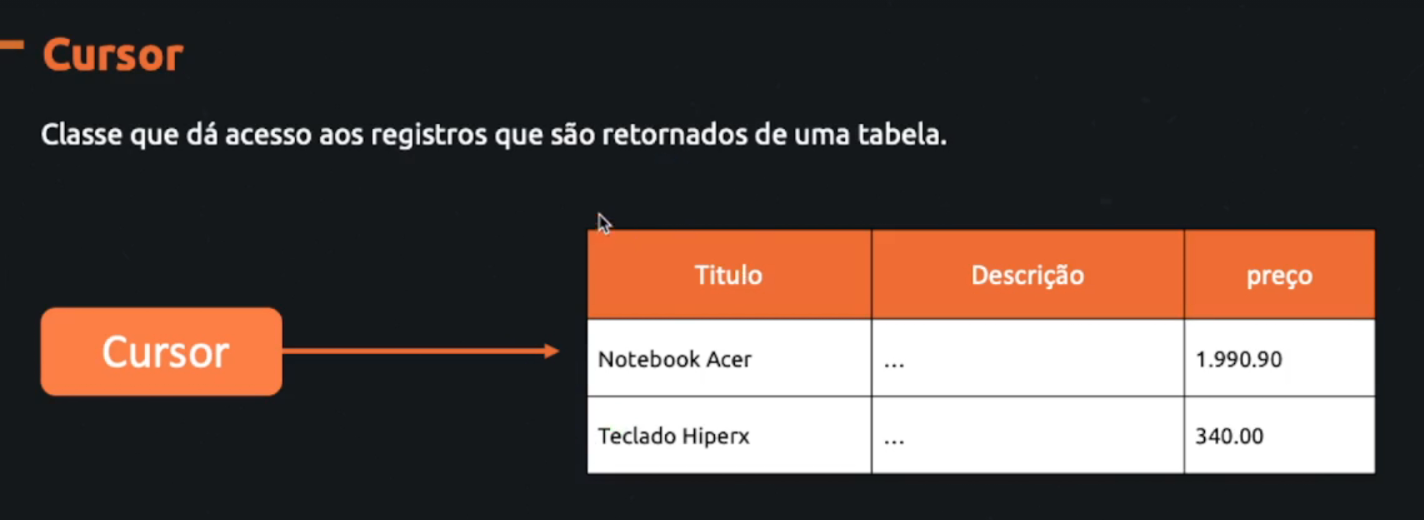
# 2 Implementar os métodos

class DatabaseHelper : SQLiteOpenHelper() {  
 override fun onCreate(db: SQLiteDatabase?) {  
 *TODO*("Not yet implemented")  
 }  
  
 override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  
 *TODO*("Not yet implemented")  
 }  
}

# 3 Construtor da classe

No construtor vamos precisar de 3 parametros





class DatabaseHelper(context : Context) : SQLiteOpenHelper(  
  
 */\*\* O SQLiteOpenHelper  
 \* Espera receber 3 parametros.  
 \*/  
  
 /\*\* 1 PARAMETRO  
 Primeiro contexto, como nao temos ele aqui vamos pegar de outro local  
 Essa classe vamos precisar instanciar e quando a gente for instanciar vamos passar o contexto  
 Classe instanciada em mainActivity passando o contexto val dbHelper = DatabaseHelper(this) \*/  
  
  
 /\*\* 2 SEGUNDO PARAMETRO  
 \* É o nome do banco de dados  
 \* O proprio SQLiteOpneHelper que vai criar para a gente  
 \* o db não é necessario, mas é sempre bom colocar para os proximos devs entender que estamos falando de um banco de dados.  
 \*/  
  
 /\*\* 3 PARAMETRO Cursor Factor  
 \* Esse cursos da acesso aos registros que sao retornados de uma tabela  
 \* Usamos a classe Cursor para acessar os registros dentro de cada uma das tabelas  
 \* Atenção esse Cursor os dados retorna como se fosse um select, mas podemos fazer um personalizado para definir como os dados vai ser retornados para nos  
 \* Aqui vamos usar o NULL porque nao vamos criar um Cursor customizado  
 \*/  
  
 /\*\* 4 Versao do SQLite  
 \* Vamos colocar a versao 1  
 \* Pra que serve?  
 \* Imagina que temos os postos de GNV do RJ versao 1  
 \* E ai decidimos adicionar calibradores  
 \* Então vamos ter a versao 2 (postos gnv e calibradores)  
 \* Então o metodo onUpgrade vai atualizar a nova versao que esta dentro do celular do usuario  
 \*  
 \* Resumindo imagina que meu app esta na loja e saiu uma nova versao do SQLite o proprio onUpgrade vai atualizar para a nova versao  
 \* Sem apagar os dados antigos, para evitar que o usuario possa perde seus dados.  
 \*/* context, "loja.db", null, 1  
  
) {  
 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  
 \*/  
  
  
 TODO*("Not yet implemented")  
 }  
  
 override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  
 */\*\* Aqui é onde vai ser atualizado as versoes do banco de dados  
 \* Sempre que tiver alguma mudança (alguma tabela nova...) vamos atualizar aqui.  
 \*/  
 TODO*("Not yet implemented")  
 }  
}