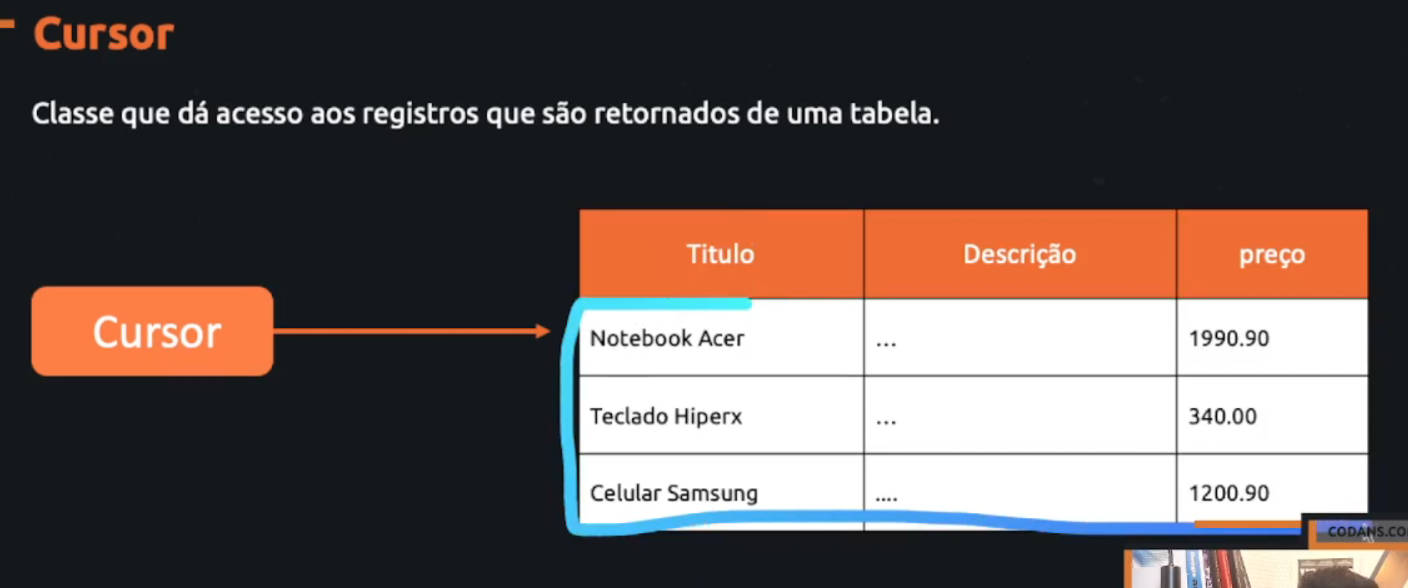


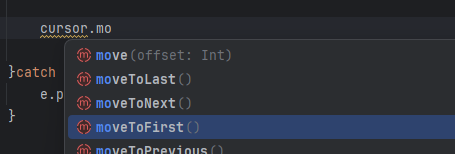
rawQuery retorna um objeto cursor

Dentro do cursor é onde esta todos os registros da tabela.



Vamos a pratica

Como uso o cursor?



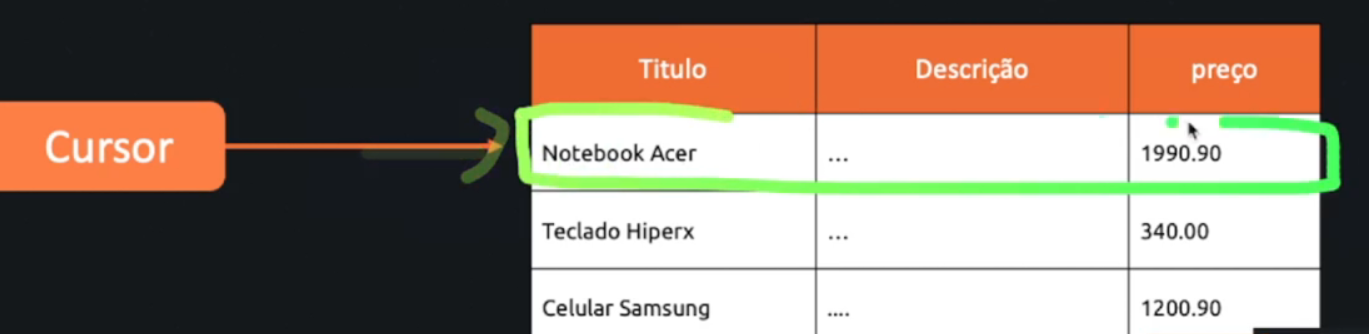
Aqui temos os métodos, o moveToFirts ele vai pegar o primeiro item da tabela

moveToLast = Vamos mover para o ultimo.

MoveToPosition = Podemos por a posição que queremos.

Lembrar que a posição começa com 0, igual ao array.

Ex exibindo o item 0



moveToPrevious = mover para o anterior

moveToNext = mover para o próximo

Imagina o moveToNext, como ele saberia quando acabou?

Porque toda vez que ele consegue acessar o próximo registro ele vai retornar um true e com isso podemos fazer um while, enquanto for verdadeiro ele vai avançando para o próximo registro



Ai quando chegar no ultimo registro ele vai tentar ler o próximo e vai retornar um false, ai vamos sair do loop pq é sinal que acabou nossos registros.

# Pratica:

Temos um botão para exibir os itens

Nesse botão chamamos esse método

E aqui esta o código

private fun listar() {  
 val comandoSQL = "SELECT \* FROM produtos;"  
  
  
 val cursor = bancoDeDados.*readableDatabase* .rawQuery(  
 comandoSQL,  
 null) // Readable Database é para fazer a leitura do banco de dados.  
  
 while ( cursor.moveToNext() ) {  
 */\*\* Cursor = Onde nosso mouse esta  
 \* Exemplo imagina os registros/linhas e nosso mouse em cima da linha 1  
 \* Então o cursor é isso é onde vai esta nosso mouse  
 \* moveToNext ele vai sair da primeira casa e vai avançar para a segunda, toda vez que ele avançar ele vai gerar um TRUE OU FALSE  
 \* Se for false ele para, pq é sinal que não tem mais registros  
 \*  
 \* Por isso estamos fazendo um while, pq enquanto for verdadeiro ele vai avançando...  
 \* MoveToNext ele começa sempre no primeiro registro e vai andando para baixo...  
 \*/* Log.i("info\_db", "Posição do cursor = ${cursor.*position*}") // Contagem para saber quantos registros ele vai pegar...  
 }  
  
  
}

# Agora vamos recuperar cada uma das colunas

private fun listar() {  
 val comandoSQL = "SELECT \* FROM ${DatabaseHelper.TABELA\_PRODUTOS};"  
  
  
 val cursor = bancoDeDados.*readableDatabase* .rawQuery(  
 comandoSQL,  
 null) // Readable Database é para fazer a leitura do banco de dados.  
 */\*\* Recuperar as colunas/Registros  
 \* getInt porque estamos recuperando o ID e sabemos que o ID vai ser sempre um Int  
 \* No parametro vamos colocar a casa da coluna que vai se comportar como se fosse um array  
 \* Temos 3 colunas  
 \* id\_produto = 0 / titulo = 1 / descrição = 2  
 \* \*/* val idProduto = cursor.getInt(0)  
 val tituloProduto = cursor.getString(1)  
 val descricaoProduto = cursor.getString(2)  
  
 Log.i("info\_db", "id: $idProduto - titulo: $tituloProduto - descrição: $descricaoProduto")  
  
  
 Log.i("info\_db", "Posição do cursor = ${cursor.*position*}") // Contagem para saber quantos registros ele vai pegar...  
 }  
  
  
}

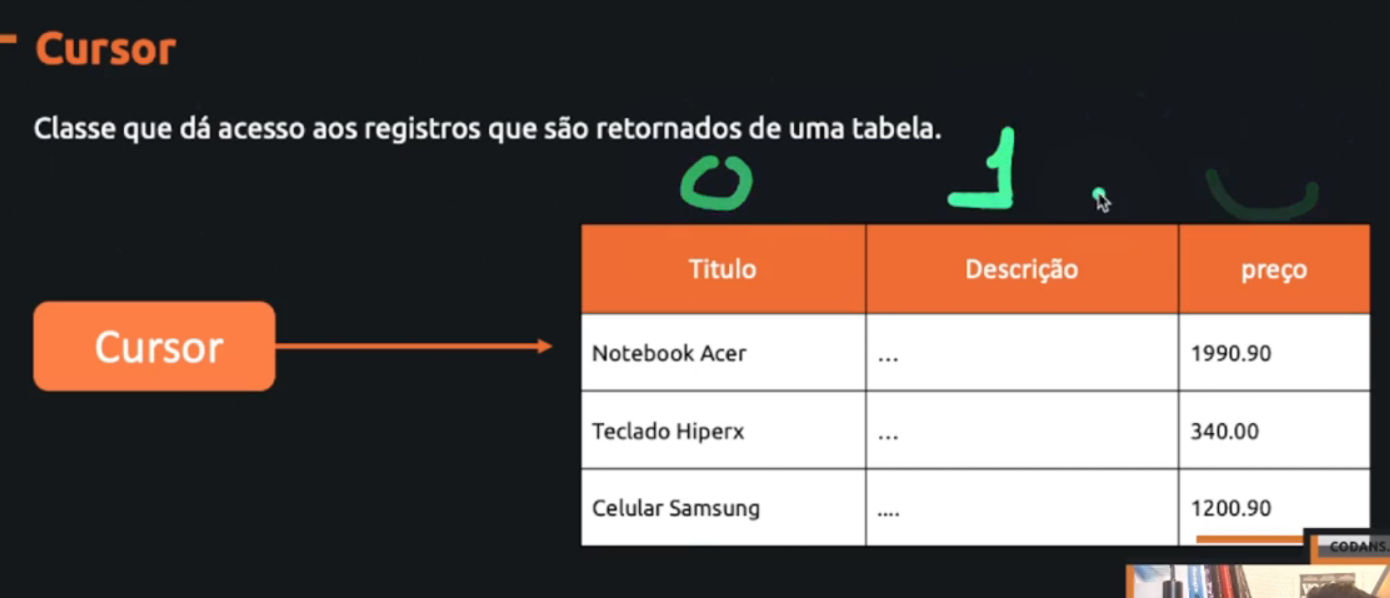
Agora no nosso Log, vamos ter cada item do nosso banco de dados

Lembrar que a coluna ela se comporta como se fosse um array

Titulo = 0

Descrição = 1

Preço = 2



# Outra forma de saber qual o índice da coluna para evitar de ficar escrevendo números

Dessa forma não precisamos ficar lembrando a ordem que esta os itens, apenas chamamos pelo nome, porque a ordem pode mudar de acordo com o SELECT que eu dou...

private fun listar() {  
 val comandoSQL = "SELECT \* FROM ${DatabaseHelper.TABELA\_PRODUTOS};"  
  
  
 val cursor = bancoDeDados.*readableDatabase* .rawQuery(  
 comandoSQL,  
 null) // Readable Database é para fazer a leitura do banco de dados.  
  
  
 */\*\*Dessa forma em vez de lembrar qual é cada indice, vamos passar o nome e o proprio kotlin vai passar o indice para a gente   
 \* Agora ele vai retornar para a gente os dados que estao dentro de produto, titulo e descricao\*/* val indiceId = cursor.getColumnIndex("id\_produto")  
 val indiceTitulo = cursor.getColumnIndex("titulo")  
 val indiceDescricao = cursor.getColumnIndex("descricao")  
  
 while ( cursor.moveToNext() ) {  
 */\*\* Cursor = Onde nosso mouse esta  
 \* Exemplo imagina os registros/linhas e nosso mouse em cima da linha 1  
 \* Então o cursor é isso é onde vai esta nosso mouse  
 \* moveToNext ele vai sair da primeira casa e vai avançar para a segunda, toda vez que ele avançar ele vai gerar um TRUE OU FALSE  
 \* Se for false ele para, pq é sinal que não tem mais registros  
 \*  
 \* Por isso estamos fazendo um while, pq enquanto for verdadeiro ele vai avançando...  
 \* MoveToNext ele começa sempre no primeiro registro e vai andando para baixo...  
 \*/  
  
 /\*\* Recuperar as colunas/Registros  
 \* getInt porque estamos recuperando o ID e sabemos que o ID vai ser sempre um Int  
 \* No parametro vamos colocar a casa da coluna que vai se comportar como se fosse um array  
 \* Temos 3 colunas  
 \* id\_produto = 0 / titulo = 1 / descrição = 2  
 \* \*/* val idProduto = cursor.getInt(indiceId)  
 val tituloProduto = cursor.getString(indiceTitulo)  
 val descricaoProduto = cursor.getString(indiceDescricao)  
  
 Log.i("info\_db", "id: $idProduto - titulo: $tituloProduto - descrição: $descricaoProduto")  
  
  
 Log.i("info\_db", "Posição do cursor = ${cursor.*position*}") // Contagem para saber quantos registros ele vai pegar...  
 }