

Atualizações

Para a atualização de dados, utilizamos a estrutura `Update`. No entanto, necessitamos apenas definir em qual tabela atuaremos, quais são os atributos e respectivos valores e em quais registros esta alteração será aplicada. Faremos as alterações de maneira semelhante aos updates SQL a seguir:

```
UPDATE Musica SET idEstilo = 2 WHERE idBanda = 3;  
UPDATE Album SET idEstilo = 2 WHERE idBanda = 3;
```

Para iniciar, crie um arquivo do tipo Swift com o nome `AlterarEstilo` e implemente o seguinte código:

```
import Foundation  
import SwiftKuery  
  
func atualizaEstilo() {  
    let utils = CommonUtils.sharedInstance  
    let musicas = Musicas()  
    let bandas = Bandas()  
    let albuns = Albuns()  
    let estilos = Estilos()  
  
}
```

Para alterar os estilos de álbuns e músicas e constatar as alterações, vamos construir rotinas para apresentar o conteúdo das tabelas antes e depois das alterações. Para alcançar esse objetivo, obtivemos as instâncias das classes que representam as tabelas envolvidas no processamento. Feito isso, construiremos uma função que será dedicada à apresentação do conteúdo das tabelas com alguma formatação:

```
func atualizaEstilo() {
    let utils = CommonUtils.sharedInstance
    let musicas = Musicas()
    let bandas = Bandas()
    let albuns = Albuns()
    let estilos = Estilos()

    func imprime() {
        print("""
            \("Banda".fill()) \("Música".fill()) Estilo
            -----
            """)

        consulta(Select(bandas.nome, musicas.nome,
            estilos.nome,
                from: musicas)
            .join(bandas).on(musicas.idBanda ==
            bandas.idBanda)
            .join(estilos).on(musicas.idEstilo ==
            estilos.idEstilo)
            .where(musicas.idBanda == 3))

        print("""
            \("Álbum".fill()) Estilo
            -----
            """)
    }
}
```

```

    consulta(Select(albums.nome, estilos.nome, from:
albums)
    .join(estilos).on(albums.idEstilo ==
estilos.idEstilo)
    .where(albums.idBanda == 3))
}
}

```

A função `imprime` foi declarada internamente na função `atualizaEstilo` para permitir o acesso às instâncias das representações das tabelas, aproveitando de umas das vantagens desse tipo de implementação que deixa o código mais limpo. Nele criamos dois `Select`, um para cada tabela que receberá alterações em seus dados. Agora, implementaremos a atualização dos dados:

```

func atualizaEstilo() {
  let utils = CommonUtils.sharedInstance
  let musicas = Musicas()
  let bandas = Bandas()
  let albums = Albums()
  let estilos = Estilos()

  func imprime() {
    print("""
      \("Banda".fill()) \("Música".fill()) Estilo
      -----
    """)
  }
}

```

```

        consulta(Select(bandas.nome, musicas.nome,
estilos.nome,
            from: musicas)
                .join(bandas).on(musicas.idBanda ==
bandas.idBanda)
                .join(estilos).on(musicas.idEstilo ==
estilos.idEstilo)
                .where(musicas.idBanda == 3))

        print("""

            \("Álbum".fill()) Estilo
            -----
            """)

        consulta(Select(albuns.nome, estilos.nome, from:
albuns)
                .join(estilos).on(albuns.idEstilo ==
estilos.idEstilo)
                .where(albuns.idBanda == 3))
    }

    imprime()
    print("\n- - - - -
- \n")

    utils.executaQuery(Update(musicas, set:
[(musicas.idEstilo, 2)],
                        where: musicas.idBanda == 3))
    utils.executaQuery(Update(albuns, set:
[(albuns.idEstilo, 2)],
                        where: albuns.idBanda == 3))

    imprime()
    print()

}

```

O código em destaque inicia apresentando os dados antes da atualização, seguido das atualizações, e como ficou depois da atualização. A sintaxe para o `Update` segue o mesmo padrão que a do SQL (apresentada anteriormente).

Vamos alterar o programa main para adicionar a chamada à função `atualizaEstilo` e ver o resultado da execução:

```
import Foundation

//criaTabelas()
//insereDados()
//consulta1()
//consulta2()
atualizaEstilo()
```

O resultado dessa execução pode ser visto abaixo:

Banda	Música	Estilo
Alix Perez	Burning Babylon	Jungle
Alix Perez	Empty Words	Jungle
Álbum	Estilo	
Empty words	Jungle	

Banda	Música	Estilo
Alix Perez	Burning Babylon	Dance
Alix Perez	Empty words	Dance
Álbum	Estilo	
Empty words	Dance	

As consultas que aparecem acima da linha tracejada correspondem às informações antes da atualização e as que aparecem abaixo da linha correspondem às informações após a atualização. Assim, é possível constatar que as alterações foram feitas com sucesso, pois os estilos das músicas e álbum mudaram de **Jungle** para **Dance**.