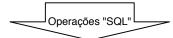


Graças à relação entre
as classes (Classe
Agente declarada
dentro da classe
Transação) com a
anotação
@ManyToOne, é
possível recuperar o
objeto AGENTE
quando recuperamos
uma transação

Recuperando 1 obj Transação

```
"id":1234,
   "dataHora" : "2020-06-01T18:00:00",
   "valorSolicitado": 1000.0,
   "valorAutorizado": 1000.0,
   "status": 0,
   "agente": {
        "id": 10,
        "nome": "Carrefour ",
        "volume":191821
   }
}
```



Possibilidade 1

Recuperar objetos puros (model)

- query By Method Name Ex:
- findAllByAgente (Agente agente)
- findAllByDataHora(LocalDateTime dataHora)
- findAllByStatus(int status)

Possibilidade 2

Consultas mais elaboradas (que não retornem objetos puros, mas tipos de dados derivados)

Exemplos:

- Totais de Eventos por status a partir de um determinado agente
- Numero de eventos de fraude a partir de uma data
- Totais de Fraude por Agente e por Dia

SOLUÇÃO: CONSULTAS CUSTOMIZADAS

Necessidade

Retornar uma lista onde tenha em cada linha os seguintes dados:

- Nome do agente financeiro
- Volume Transacional
- status da transação
- quantidade destes status

A partir da análise desta necessidade, temos o objeto **Transacao** como base

Necessidade do DTO

O Resultado da consulta me informa 4 valores distintos: String, float, int, long respectivamente para: nome, volume, status, count(status)

Para suportar isso, criamos uma nova classe (um novo tipo de dado) - neste caso a classe ConsolidadoStatus

Montando a Query (HQL)

HQL = vou fazer um SQL só que ao invés de usar Tabelas, uso Classes

Considerações:

- O objeto **Transacao** será depois internamente mapeado pa a respectiva tabela (**@Table**)
- O "alias" tr é para apenas facilitar a referência à classe Transacao

Modificando a Query

WHERE tr.agente.id = ?
GROUP BY tr.status;