**Especialização em Ciência de Dados – PUC/RS**

**Disciplina:** Bancos de Dados SQL e NoSQL Enunciado 2º trabalho prático - SQL 2019/1

**Professor:** Duncan Dubugras Alcoba Ruiz

**Tarefa:** Exportação de dados de SGBD Oracle para inserção em coleções de documentos do serviço “NoSQL” MongoDB e consultas específicas utilizando a API PyMongo.

**Fonte de dados:** Comunicação de Acidentes de Trabalho – CAT Base (2018-2019)

**Equipe 11:** Danilo Heroso de Deus Pereira e João Paulo Medeiros Cecílio

Com base no diagrama da base, acessos fornecidos e informações solicitadas aos integrantes da equipe 11, a visão utilizada e as 3 consultas realizadas para o Trabalho 2 da disciplina são demonstradas a seguir:

**Visão Utilizada na Extração**

**Campos:** Nome do município, nome do estado do respectivo município, população do município e todos os campos da tabela de acidentes de trabalho;

CREATE VIEW equipe11\_view AS

SELECT m.nome\_município,

m.nome\_uf,

p.população,

a.\*

FROM duncanbda.acid\_trab\_2018jul\_2019mar\_11 a

INNER JOIN duncanbda.municipios m

ON a.munic\_empregador = m.cod\_munic

INNER JOIN duncanbda.municipios\_populacoes p

ON m.cod\_munic = p.cod\_munic

**Visão Utilizada - Explicação:**

A criação da visão (*view*) é composta de uma consulta que itera primeiramente pelos registros da tabela acid\_trab\_2018jul\_2019mar\_11 para trazer os registros da tabela de acidentes e assim faz as junções com tabelas complementares de município e população de município para obter informações acerca de nome de município, seu respectivo estado e sua respectiva população. Este agrupamento gera redundância dos valores dos registros encontrados, mas o formato de documento para o MongoDB apresenta melhor desempenho na obtenção de documentos se não houver relacionamento entre coleções.

**Visão Utilizada – Dimensões de retorno**

As dimensões da visão resultante possuem 26248 linhas (além do cabeçalho) e 25 colunas.

**Visão Utilizada – Primeiras 19 linhas e 8 colunas de resultados:**

| POPULAÇÃO | NOME\_MUNICÍPIO | NOME\_UF | MES\_ANO\_ACIDENTE | DATA\_ACIDENTE | TIPO\_DO\_ACIDENTE | UF\_MUNIC\_ACIDENTE | AGENTE\_CAUSADOR\_ACIDENTE |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 10886518 | São Paulo | São Paulo | 01/07/2018 | 16/07/2018 | Típico | Maranhão | Escada Movel ou Fixa |
| 144857 | São Caetano do Sul | São Paulo | 01/07/2018 | 15/07/2018 | Típico | Maranhão | Tanque, Cilindro (Tr |
| 10334 | Nhandeara | São Paulo | 01/07/2018 | 05/07/2018 | Típico |  | Veiculo Sobre Trilho |
| 212956 | São Carlos | São Paulo | 01/06/2018 | 26/06/2018 | Típico | Maranhão | Cadeira Banco - Mobi |
| 13752 | Borborema | São Paulo | 01/06/2018 | 25/06/2018 | Trajeto | Maranhão | Bicicleta |
| 701012 | Osasco | São Paulo | 01/07/2018 | 03/07/2018 | Doença | Maranhão | Ataque de Ser Vivo, |
| 91479 | Itatiba | São Paulo | 01/07/2018 | 02/07/2018 | Típico | Maranhão | Maquina, Nic |
| 73014 | Paulínia | São Paulo | 01/07/2018 | 01/07/2018 | Típico | Maranhão | Rua e Estrada - Supe |
| 594948 | São José dos Campos | São Paulo | 01/07/2018 | 02/07/2018 | Típico | Maranhão | Ferramenta Manual se |
| 10886518 | São Paulo | São Paulo | 01/07/2018 | 05/07/2018 | Típico | Maranhão | Rampa - Superficie U |
| 34018 | Tietê | São Paulo | 01/07/2018 | 03/07/2018 | Típico | Maranhão | Couro Cru ou Curtido |
| 1236192 | Guarulhos | São Paulo | 01/06/2018 | 21/06/2018 | Típico | Maranhão | Escada Movel ou Fixa |
| 9045 | Guaraci | São Paulo | 01/07/2018 | 12/07/2018 | Típico | Maranhão | Escada Movel ou Fixa |
| 88815 | Caraguatatuba | São Paulo | 01/02/2018 | 02/02/2018 | Trajeto | Maranhão | Chao - Superficie Ut |
| 10886518 | São Paulo | São Paulo | 01/07/2018 | 11/07/2018 | Trajeto | Maranhão | Calcada ou Caminho p |
| 102311 | Salto | São Paulo | 01/07/2018 | 10/07/2018 | Típico | Maranhão | Madeira (Toro, Madei |
| 103394 | Birigui | São Paulo | 01/07/2018 | 12/07/2018 | Trajeto | Maranhão | Veiculo Rodoviario M |
| 109362 | Catanduva | São Paulo | 01/07/2018 | 12/07/2018 | Típico | Maranhão | Prensa - Maquina |
| 88815 | Caraguatatuba | São Paulo | 01/07/2018 | 12/07/2018 | Típico | Maranhão | Produto Biologico (S |

**Consultas MongoDB - Introdução**

As consultas dos dados no serviço MongoDB são elaboradas para utilização da biblioteca PyMongo, que consiste em chamadas usando cursores apontando à coleção especificada.

Neste caso, a base definida db na conexão chamando client.datascience. E para interagir com documentos e estruturas desta base, é definido o cursor acidentes para interagir com a coleção "dupla\_11\_collection" dentro da base db.

Para elaboração das consultas do cursor, foi necessário utilizar o método .aggregate(), que consiste na formulação de canalização sequencial de operações para criação de agrupamentos dos dados solicitados. As sequências de operações são definidas dentro da variável pipeline.

A conversão em lista no final de cada execução ocorre para transformar o retorno do cursor em uma sequência que o comando pprint() pode exibir em tela.

**Consulta MongoDB 1:** Quais os 10 municípios com maior taxa de acidentes por 100 mil habitantes?

db = client.datascience

acidentes = db["dupla\_11\_collection"]

pipeline = [

{"$group":

{

"\_id": "$NOME\_MUNICÍPIO",

"count": {"$sum": 1},

"pop": { "$avg": "$POPULAÇÃO" }

}

},{"$addFields":

{

"acidentes\_por\_100mil": { "$divide": ["$count", { "$divide": ["$pop", 100000] }] },

}

},{

"$sort": {"acidentes\_por\_100mil": -1}

},{

"$limit": 10

}

]

pprint(list(acidentes.aggregate(pipeline)))

**Consulta 1 - Explicação:**

O pipeline destas operações de agregação consiste na utilização dos seguintes estágios de filtros em sequência:

$group: Agrupar os registros de acidente pelo identificador $NOME\_MUNICÍPIO, e realizando a contagem para quantificar os registros por cidade, e a operação de média de população (cada registro de acidente da mesma cidade possui o mesmo valor para população apontado, assim a média retorna o mesmo valor de um único registro), para obter valor da população.

$addFields: Estágio que adiciona novos campos aos documentos, no caso o acidentes\_por\_100mil, que se baseia na divisão do número de acidentes registrados em $count pela proporção da população da cidade dividida por 100.000 ($pop/100000).

$sort: Este estágio realiza a ordenação decrescente pelo campo (anteriormente criado) acidentes\_por\_100mil .

$limit: Este último estágio realiza a seleção dos 10 primeiros registros de cidade que iniciam pelo campo de acidentes por 100 mil habitantes.

**Consulta 1 - Resultados**:

[{'\_id': 'Borá',

'acidentes\_por\_100mil': 1119.402985074627,

'count': 9,

'pop': 804.0},

{'\_id': 'Vista Alegre do Alto',

'acidentes\_por\_100mil': 278.6885245901639,

'count': 17,

'pop': 6100.0},

{'\_id': 'Onda Verde',

'acidentes\_por\_100mil': 267.6659528907923,

'count': 10,

'pop': 3736.0},

{'\_id': 'Cosmorama',

'acidentes\_por\_100mil': 258.9555459646094,

'count': 18,

'pop': 6951.0},

{'\_id': 'Ipiguá',

'acidentes\_por\_100mil': 254.77707006369425,

'count': 10,

'pop': 3925.0},

{'\_id': 'Mendonça',

'acidentes\_por\_100mil': 226.13065326633165,

'count': 9,

'pop': 3980.0},

{'\_id': 'Rio das Pedras',

'acidentes\_por\_100mil': 208.77619192225933,

'count': 55,

'pop': 26344.0},

{'\_id': 'Araçariguama',

'acidentes\_por\_100mil': 203.40086241965665,

'count': 25,

'pop': 12291.0},

{'\_id': 'Nova Independência',

'acidentes\_por\_100mil': 201.61290322580646,

'count': 5,

'pop': 2480.0},

{'\_id': 'Pontes Gestal',

'acidentes\_por\_100mil': 201.04543626859672,

'count': 5,

'pop': 2487.0}]

**Consulta MongoDB 2:** Quais as Naturezas das lesões, e correspondentes números de acidentes de trabalho, para naturezas de lesões cujo número de acidentes de trabalho é > 100?

db = client.datascience

acidentes = db["dupla\_11\_collection"]

pipeline = [

{"$group":

{

"\_id": "$NATUREZA\_DA\_LESAO",

"count": {"$sum": 1}

}

},{

"$match": { "count": { "$gte": 100 } }

},{

"$sort": {"count": -1}

}

]

pprint(list(acidentes.aggregate(pipeline)))

**Consulta 2 - Explicação**:

O pipeline destas operações de agregação consiste na utilização dos seguintes estágios de filtros em sequência:

$group: Agrupa os registros pela identidade $NATUREZA\_DA\_LESAO e guarda a quantidade pelo valor count, para realizar a contabilização dos registros pela natureza da lesão.

$match: Dentre os agrupamentos realizados, filtrar para aqueles cuja contagem seja maior ou igual ($gte) a 100 (incidentes de trabalho).

$sort: Ordenar por campo de contabilização de registros em ordem decrescente (-1).

**Consulta 2 - Resultados:**

[{'\_id': 'Corte, Laceracao, Fe', 'count': 5518},

{'\_id': 'Contusao, Esmagament', 'count': 4337},

{'\_id': 'Fratura ', 'count': 2672},

{'\_id': 'Distensao, Torcao ', 'count': 1731},

{'\_id': 'Fratura', 'count': 1544},

{'\_id': 'Lesao Imediata, Nic ', 'count': 1542},

{'\_id': 'Escoriacao, Abrasao ', 'count': 1346},

{'\_id': 'Distensao, Torcao', 'count': 930},

{'\_id': 'Luxacao ', 'count': 888},

{'\_id': 'Lesao Imediata, Nic', 'count': 765},

{'\_id': 'Queimadura ou Escald', 'count': 750},

{'\_id': 'Escoriacao, Abrasao', 'count': 725},

{'\_id': 'Luxacao', 'count': 512},

{'\_id': 'Lesao Imediata ', 'count': 412},

{'\_id': 'Lesoes Multiplas ', 'count': 406},

{'\_id': 'Doenca, Nic ', 'count': 257},

{'\_id': 'Inflamacao de Articu', 'count': 235},

{'\_id': 'Lesoes Multiplas', 'count': 211},

{'\_id': 'Amputacao ou Enuclea', 'count': 209},

{'\_id': 'Lesao Imediata', 'count': 193},

{'\_id': 'Doenca, Nic', 'count': 168},

{'\_id': 'Queimadura Quimica (', 'count': 158},

{'\_id': 'Doenca Contagiosa ou', 'count': 134},

{'\_id': 'Concussao Cerebral ', 'count': 126}]

**Consulta MongoDB 3:** Quais os agentes causadores de acidentes, e correspondentes números de acidentes, por mês-ano e para acidentes com óbitos, onde o número de óbitos por agente causador é > 2?

db = client.datascience

acidentes = db["dupla\_11\_collection"]

pipeline = [

{ "$match": { "INDICA\_OBITO\_ACIDENTE": { "$eq": 'Sim' } } },

{ "$group": {

"\_id": {

"agente" : "$AGENTE\_CAUSADOR\_ACIDENTE",

"mes\_ano" : "$MES\_ANO\_ACIDENTE"

},

"count": {"$sum": 1}

}

},{

"$match": {

"count": { "$gte": 2 }

}

},{

"$sort": {"count": -1}

}

]

pprint(list(acidentes.aggregate(pipeline)))

**Consulta 3 - Explicação**:

O pipeline destas operações de agregação consiste na utilização dos seguintes estágios de filtros em sequência:

$match: Realiza a separação para apurar apenas os registros onde há indicação de óbito usando o operador $eq ao valor 'Sim'.

$group: Dentre os registros selecionados, agrupar os registros numa registro de identidade composto (\_id) pelos valores $AGENTE\_CAUSADOR\_ACIDENTE e $MES\_ANO\_ACIDENTE fazendo a contabilização dos registros pelo agregador count.

$match: Nova realização de etapa match, onde apenas os grupos com 2 ou mais ($gte) registros contabilizados serão considerados.

$sort: Ordenar de forma decrescente (-1) os grupos (de agente causador por mês ano) por quantidade de registros (count) de óbito (maiores que 2).

**Consulta 3 - Resultados**:

[{'\_id': {'agente': 'Veiculo Rodoviario M', 'mes\_ano': '01/11/18'}, 'count': 7},

{'\_id': {'agente': 'Veiculo, Nic', 'mes\_ano': '01/08/18'}, 'count': 4},

{'\_id': {'agente': 'Veiculo Rodoviario M', 'mes\_ano': '01/01/19'}, 'count': 4},

{'\_id': {'agente': 'Veiculo Rodoviario M', 'mes\_ano': '01/03/19'}, 'count': 3},

{'\_id': {'agente': 'Veiculo, Nic', 'mes\_ano': '01/07/18'}, 'count': 3},

{'\_id': {'agente': 'Veiculo Rodoviario M', 'mes\_ano': '01/09/18'}, 'count': 3},

{'\_id': {'agente': 'Veiculo Rodoviario M', 'mes\_ano': '01/07/18'}, 'count': 3},

{'\_id': {'agente': 'Motocicleta, Motonet', 'mes\_ano': '01/02/19'}, 'count': 2},

{'\_id': {'agente': 'Motocicleta, Motonet', 'mes\_ano': '01/11/18'}, 'count': 2},

{'\_id': {'agente': 'Veiculo, Nic ', 'mes\_ano': '01/11/18'}, 'count': 2},

{'\_id': {'agente': 'Motocicleta, Motonet', 'mes\_ano': '01/03/19'}, 'count': 2},

{'\_id': {'agente': 'Motocicleta, Motonet', 'mes\_ano': '01/07/18'}, 'count': 2},

{'\_id': {'agente': 'Veiculo, Nic ', 'mes\_ano': '01/01/19'}, 'count': 2},

{'\_id': {'agente': 'Veiculo, Nic ', 'mes\_ano': '01/10/18'}, 'count': 2},

{'\_id': {'agente': 'Trator ', 'mes\_ano': '01/01/19'}, 'count': 2},

{'\_id': {'agente': 'Veiculo Rodoviario M', 'mes\_ano': '01/08/18'}, 'count': 2},

{'\_id': {'agente': 'Veiculo Rodoviario M', 'mes\_ano': '01/10/18'}, 'count': 2}]