



**DATA BASE ADMIN**



**Y U NO LEARN NOSQL?!?!?**

**Propor um novo  
índice para algum  
NoSQL**



mongoDB

# Pesquisas

404 :(

# **Análise Experimental de Bases de Dados Relacionais e NoSQL no Processamento de Consultas sobre Data Warehouse**

**O MongoDB apresentou os  
piores resultados**



**O Fastbit foi o mais rápido**

# Implementação de índice Bitmap para o banco de dados MongoDB

# Bitmap Join

## Index

**Não há join no MongoDB**

**Implementação de índice**

**Bitmap para o banco de dados**

**MongoDB**

# **Análise do Banco de Dados NoSQL MongoDB no Processamento de Consultas sobre Data Warehouse**

# Nova proposta

**Obter melhor performance em  
consultas OLAP**



# **Modelo estrela**

# **Multidimensional**

# **Parallel Data Generation Framework**

**parts**

**800.000**

**customers**

**300.000**

**suppliers**

**20.000**

**dates**

**2.557**

**lineorders**

**10.001.101**

# Consultas

Q1

```
SELECT SUM(l_extendedprice*l_discount) AS revenue
FROM lineorders, dates
WHERE l_orderdate = d_datekey
      AND d_year = 1993
      AND l_discount BETWEEN 1 AND 3
      AND l_quantity < 25;
```

```
var date_ids = [];  
db.dates.find({d_year: 1993}).forEach( function (d) {  
    date_ids.push(d._id);  
});
```

```
db.lineorders.aggregate([  
    { $match: { l_orderdate: { $in: date_ids },  
                l_discount: { $gte: 1, $lte: 3},  
                l_quantity: { $lt: 25} } },  
    { $group: { _id: null,  
                revenue: { $sum: {$multiply:  
["$l_extendedprice", "$l_discount"]} } } }  
]);
```

```
SELECT  SUM(l_revenue), d_year, p_brand
FROM    lineorders , dates, parts, suppliers
WHERE   l_orderdate = d_datekey
        AND l_partkey = p_partkey
        AND l_suppkey = s_suppkey
        AND p_category = 'MFGR#12'
        AND s_region = 'AMERICA'
GROUP BY d_year, p_brand
ORDER BY d_year, p_brand
```

**Não é possível agrupar por atributos de documentos de outras coleções :(**

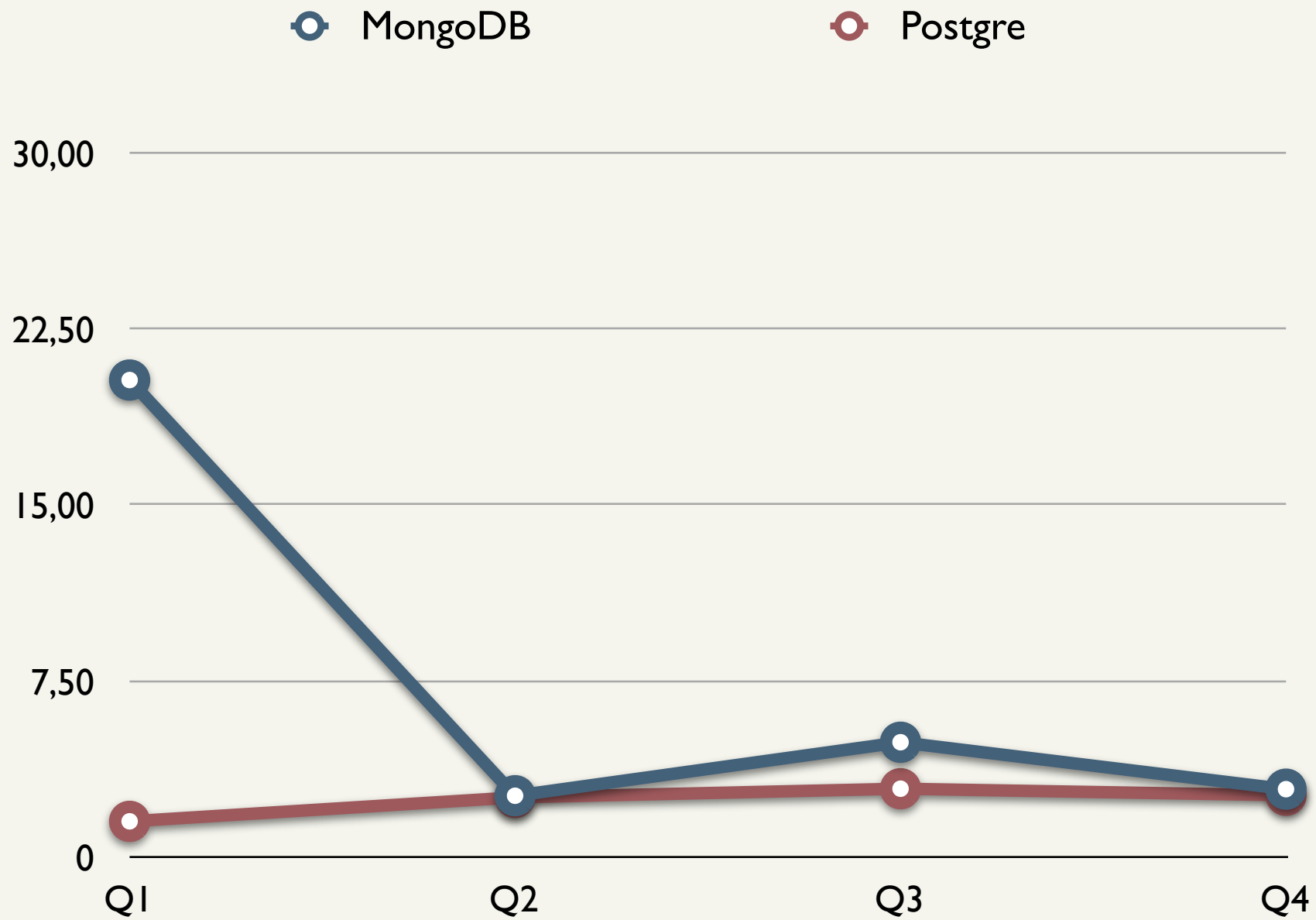


## Q2

```
SELECT  SUM(l_revenue), d_year
FROM    lineorders , dates, parts, suppliers
WHERE   l_orderdate = d_datekey
        AND l_partkey = p_partkey
        AND l_suppkey = s_suppkey
        AND p_category = 'MFGR#12'
        AND s_region = 'AMERICA'
GROUP BY d_year
ORDER BY d_year
```

```
var part_ids = [];  
db.parts.find({p_category: "MFGR#12"}).forEach( function (p) {  
    part_ids.push(p._id);  
});  
  
var supplier_ids = [];  
db.suppliers.find({s_region: "AMERICA"}).forEach( function (s) {  
    supplier_ids.push(s._id)  
});  
  
var map = function () {  
    var year = this.l_orderdate.toString().match(/[0-9]{4}/)[0];  
    emit(year, this.l_revenue);  
};  
  
var reduce = function (key, values) {  
    return Array.sum(values);  
};  
  
db.lineorders.mapReduce(  
    map,  
    reduce,  
    {  
        out: "lineorders_grouped_by_year",  
        query: {  
            l_partkey: { $in: part_ids },  
            l_suppkey: { $in: supplier_ids }  
        }  
    }  
);  
  
db.lineorders_grouped_by_year.find();
```

# Resultados



# Conclusão

**Não é possível realizar todas  
consultas OLAP “comuns” no  
MongoDB utilizando o modelo  
estrela**

**As consultas no MongoDB  
ficaram bem mais complexas  
do que em SQL**

# Trabalhos futuros



**Comparar a performance de  
consultas entre aggregation  
framework e map reduce**