**SQL TO MONGO STORE JS**

sql to mongo store js: <http://www.querymongo.com/>

1. table CRUD  
→ <http://docs.mongodb.org/manual/reference/sql-comparison/>

2. SQL to Aggregation mapping  
→ <http://docs.mongodb.org/manual/reference/sql-aggregation-comparison/>

3. no exists  
select \* from tableA a into tableC  
where not exists(select 1 from tableB b where a.columnA = b.columnB)  
→ db.tableA.find().forEach(  
function(x) {  
var record = db.tableB.findOne({columnB: x.columnA});  
if(record == null) {  
db.tableC.insert(x);  
}  
}   
)

4. join  
select a.columnA, b.columnB into tableC from tableA a join tableB b on a.id = b.id  
→db.tableA.find().forEach(  
function(x) {  
var record = db.tableB.findOne({id: x.id});  
if( record != null) {  
db.tableC.insert({columnA: x.columnA, cloumnB: record.columnB});  
}  
}  
)

5. left join  
select a.columnA, b.columnB into tableC from tableA a left join tableB b on a.id = b.id  
→db.tableA.find().forEach(  
function(x) {  
var record = db.tableB.findOne({id: x.id});  
if( record != null) {  
db.tableC.insert({columnA: x.columnA, cloumnB: record.columnB});  
}  
else {  
db.tableC.insert({columnA: x.columnA});  
}  
}  
)

6. multiple left join  
→ for each single left join ,create temp collection to get result like [5]

7. set @sqlStr = '' + @parameter + ''  
exec(@sqlStr)  
→for dynamic column name and dynamic value:  
var columnName = 'testColumn’;  
var columnValue = 'testValue’;  
var query = {};  
query[columnName] = columnValue;  
db.testCollection.find(query);

8. dateadd(day,-1,convert(varchar(7),getdate(),120)+ '-01’) – last day of last month   
→ var date = new Date();  
new Date(date.getFullYear(), date.getMonth(), 0);

9. datepart(Year,dateadd(month,-1,getdate())) – year of last month  
→ var date = new Date();  
date.setMonth(date.getMonth()-1);  
date.getFullYear();

10. datepart(month,dateadd(month,-1,getdate())) – month of last month  
→var date = new Date();  
date.getMonth(); ( month index start from 0 )

11. datediff(Day,date1,date2)  
→ Math.floor(date2.getTime()/86400000) - Math.floor(date1.getTime()/86400000)

12. RIGHT – place holder  
→ var date = new Date();  
date.getMonth() < 9 ? '0’ + (date.getMonth()+1) : '' + date.getMonth()+1;

13. cast(str as int)  
→ parseInt(str)

14. select \* into tableA from tableB  
→ db.tableB.find().forEach(function(x){db.tableA.insert(x)});

15. delete from tableA where id in ( select id from tableB)  
→ var idList = db.tableB.distinct('id’);  
db.tableA.delete({id: {$in: idList}});

16. select a.column1, b.column2 from tableA a , tableB b where a.id = b.id  
→ db.tableA.find().forEach(  
function(x) {  
var tableBData = db.tableB.findOne({id: x.id});  
if(tableBData != null) {  
db.tempTable.insert({col1: x.column1, col2: tableBData.column2});  
}  
}  
)

17. union all  
insert into tableC select columnA from tableA union all select columnB from tableB   
→ db.tableA.find({}, {\_id:0, columnA:1}).forEach(  
function(x) {  
db.tableC.insert({columnC: x.columnA});  
}   
);  
db.tableB.find({}, {\_id:0, columnB:1}).forEach(  
function(x) {  
db.tableC.insert({columnC: x.columnB});  
}   
);

18. union – upsert  
insert into tableC select columnA from tableA union select columnB from tableB   
→db.tableA.find({}, {\_id:0, columnA:1}).forEach(  
function(x) {  
db.tableC.insert({columnC: x.columnA});  
}   
);  
db.tableB.find({}, {\_id:0, columnB:1}).forEach(  
function(x) {  
var record = db.tableC.findOne({columnC: x.columnB});  
if(record == null) {  
db.tableC.insert({columnC: x.columnB});   
}  
}   
);

19. select top 1 \* from tableA order by columnA desc;  
→ db.tableA.find().sort({columnA:-1}).limit(1);

20. #table – temp table  
→ use real collection and drop after using

21. select \* from table where DT > getDate() - 20  
→ db.table .find({DT:{$gt : new Date ( new Date() - 20)}})

22.insert into tableA (A,B, C)  
 select a , b,sum(c)   
 form tableB  
 group by a,b  
→db.tableB.find({},{\_id:0,a:1,b:1,c:1}).forEach(  
function(x){  
var count = db.tableA.count(  
{A:x.a,B:x.b}  
);  
if(count == 0){  
db.tableA.insert(  
{A:x.a,B:x.b,C:x.c}  
 );  
}else{  
db.tableA.update(  
{ A:x.a,B:x.b },  
{$inc:{C:x.c}},  
false,false  
 );  
}  
}  
);

23.group by (distinct)

select column1,colunm2,count(distinct column3) count

from table

group by column1,column2

🡪 db.table.aggregate(

[

{

$group:{

\_id: {column1: "$column1", column2: "$column2"},

column1: {$push: "$column1"},

column2: {$push: "$ column2"},

column3: {$push: "$ column3"}

}

}

]

). forEach(function(x){

db.temp.insert( // 把分完组的数据放入到一个临时collectiom

{

column1: x. column1[0] ,

column2: x. column2[0],

count: x. column3

}

);

});

// 统计去重后的结果数有多少条，然后更新临时collection

db.temp.find().forEach(function(x){

var count = db.temp.distinct("count",{\_id:x.\_id}).length;

print(count);

db.temp.update(

{\_id: x.\_id},

{$set: {count: count}},

false,

false

)

});

24. left join (one to one ,one to many)

select a.columnA ,b.columnB

into tableC

from tableA a

left join tableB b on a.column1 = b.column2

→ db.tableA.find({},{column1:1,columnA:1}).forEach(

function(x){

var count = db.tableB.count({column2:x.column1});

if(count != 0){

db.tableC.insert({\_id:x.\_id,columnA:x.columnA});

db.tableB.find({column2:x.column1},{\_id:0,columnB:1}).forEach(

function(y){

db.tableC.update({\_id:x.\_id},{$push:{tableB:{columnB:y.columnB}}},false,false);

}

);

}else{

db.tableC.insert({\_id:x.\_id,columnA:x.columnA});

}

}

);