SQL to Mongo Mapping Chart

MySQL executable	Oracle executable	Mongo executable
mysqld	oracle	mongod
mysql	sqlplus	mongo

MySQL term	Mongo term/concept
database	database
table	collection
index	index
row	BSON document
column	BSON field
join	embedding and linking
primary key	_id field
group by	aggregation

MongoDB queries are expressed as JSON (BSON) objects. The following chart shows examples as both SQL and in Mongo Query Language syntax.

The query expression in MongoDB (and other things, such as index key patterns) is represented as JSON (BSON). However, the actual verb (e.g. "find") is done in one's regular programming language; thus the exact forms of these verbs vary by language. The examples below are Javascript and can be executed from the mongo shell.

SQL Statement	Mongo Statement
`CREATE TABLE USERS (a Number, b Number)`	implicit; can also be done explicitly with `db.createCollection("mycoll")`
`ALTER TABLE users ADD ...`	implicit
`INSERT INTO USERS VALUES(3,5)`	`db.users.insert({a:3,b:5})`
`SELECT a,b FROM users`	`db.users.find({}, {a:1,b:1})`
`SELECT * FROM users`	`db.users.find()`

`SELECT * FROM users WHERE age=33`	`db.users.find({age:33})`
`SELECT a,b FROM users WHERE age=33`	`db.users.find({age:33}, {a:1,b:1})`
`SELECT * FROM users WHERE age=33 ORDER BY name`	`db.users.find({age:33}).sort({name:1})`
`SELECT * FROM users WHERE age>33`	`db.users.find({age:{$gt:33}})`
`SELECT * FROM users WHERE age!=33`	`db.users.find({age:{$ne:33}})`
`SELECT * FROM users WHERE name LIKE "%Joe%"`	`db.users.find({name:/Joe/})`
`SELECT * FROM users WHERE name LIKE "Joe%"`	`db.users.find({name:/^Joe/})`
`SELECT * FROM users WHERE age>33 AND age<=40`	`db.users.find({'age':{$gt:33,$lte:40}})`
`SELECT * FROM users ORDER BY name DESC`	`db.users.find().sort({name:-1})`
`SELECT * FROM users WHERE a=1 and b='q'`	`db.users.find({a:1,b:'q'})`
`SELECT * FROM users LIMIT 10 SKIP 20`	`db.users.find().limit(10).skip(20)`
`SELECT * FROM users WHERE a=1 or b=2`	`db.users.find( { $or : [ { a : 1 } , { b : 2 } ] } )`
`SELECT * FROM users LIMIT 1`	`db.users.findOne()`

`SELECT order_id FROM orders o, order_line_items li WHERE li.order_id=o.order_id AND li.sku=12345`	`db.orders.find({"items.sku" :12345},{_id:1})`
`SELECT customer.name FROM customers,orders WHERE orders.id="q179" AND orders.custid=customer.id`	`var o = db.orders.findOne({_id:"q179" });` `var name = db.customers.findOne({_id:o.custid})`
`SELECT DISTINCT last_name FROM users`	`db.users.distinct('last_name')`
`SELECT COUNT(*y)` `FROM users`	`db.users.count()`
`SELECT COUNT(*y)` `FROM users where AGE > 30`	`db.users.find({age: {'$gt': 30}}).count()`
`SELECT COUNT(AGE) from users`	`db.users.find({age: {'$exists': true }}).count()`
`CREATE INDEX myindexname ON users(name)`	`db.users.ensureIndex({name:1})`
`CREATE INDEX myindexname ON users(name,ts DESC)`	`db.users.ensureIndex({name:1,ts:-1})`
`EXPLAIN SELECT * FROM users WHERE z=3`	`db.users.find({z:3}).explain()`
`UPDATE users SET a=1 WHERE b='q'`	`db.users.update({b:'q'}, {$set:{a:1}}, false, true)`

`UPDATE users SET a=a+2 WHERE b='q'`	`db.users.update({b:'q'}, {$inc:{a:2}}, false, true)`
`DELETE FROM users WHERE z="abc"`	`db.users.remove({z:'abc'});`

More examples, specifically aggregation examples, here

See Also

- The MongoDB Manual Pages are a good place to learn more.
- SQL to Shell to C++