# SQL vs NoSQL

NoSQL (Not only SQL)

Diagram

Description automatically generated

A picture containing diagram

Description automatically generated

Graphical user interface, text, application, table

Description automatically generated

The popularity of NoSQL has been driven by the following reasons:

The pace of development with NoSQL databases can be much faster than with a SQL database

The structure of many different forms of data is more easily handled and evolved with a NoSQL database

The amount of data in many applications cannot be served affordably by a SQL database

The scale of traffic and need for zero downtime cannot be handled by SQL

New application paradigms can be more easily supported

Diagram

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, diagram

Description automatically generated with medium confidence

# Demo

Can use ‘mongosh’ in any shell/terminal to activate the mongo shell

db.<collection\_name>.insertOne(<dict>) – insert one document

Text

Description automatically generated

db.<collection\_name>.insertMany([<dict1>, <dict2>]) – insert multiple documents

Graphical user interface, text

Description automatically generated

db.<collection\_name>.find() – displays all data in the collection

db.employee.find({name:'Alan'}) – find a specific document

db.learners.find({name:{$in:['Sarah', 'Faiz']}}) – find multiple data based on condition

db.employee.findOne() – finds one document from the collection

Text

Description automatically generated

‘$’ dollar sign represents operator

‘gt’ is greater than

‘lt’ is less than

Graphical user interface, text

Description automatically generated

{$and:[{bedrooms:{$gt:8}},{beds:{$gte:16}}]}

$and: for multiple conditions to be met

$or: for either condition to be met

Text

Description automatically generated

<https://www.mongodb.com/developer/products/mongodb/cheat-sheet/>

<https://www.mongodb.com/docs/manual/replication/>

db.inventory.find({qty:{$gte:75}})

^finding item greater than or equal to 75

db.inventory.find({$or:[{qty:{$gte:75}},{item:'journal'}]})

^ finding item greater than or equal to 75 OR journal

db.inventory.find({qty:$gte:75},{item’paper’})

^don’t have to use $and at the beginning for and

db.inventory.find({size:{h:14,w:21,uom:’cm’}})

uom=unit of measurement. Will not return anything without uom

db.inventory.find({“size.h”:14})

If accessing a nested document

db.inventory.find({‘size.h’:{$lt:15},’size.uom’:’in’,status:’D’})

db.<collection\_name>.find( { <filter> } , {  <fields\_filter>: 1 } ) – you can use 1 to show fields filtered, or 0 to hide fields. **(profiling)**

AGGREGATE:

Group:

{

$group:

{

\_id: <expression>, // Group key

<field1>: { <accumulator1> : <expression1> },

...

}

}

When accessing an array:

(missed notes but look at website)

Text

Description automatically generated

If want to find just one specific product in an array, put it in a solo list []

Text

Description automatically generated

Text

Description automatically generatedIf want to find with only 2 specific items, add them to list []

To find all where a product is included in the list. Can also do this one without the $all

Collection=table. Made up of

Document=rows

Fields=titles

Values=corresponding info