

or select by  
→ group by ✓

\* find the max toy price in each category.

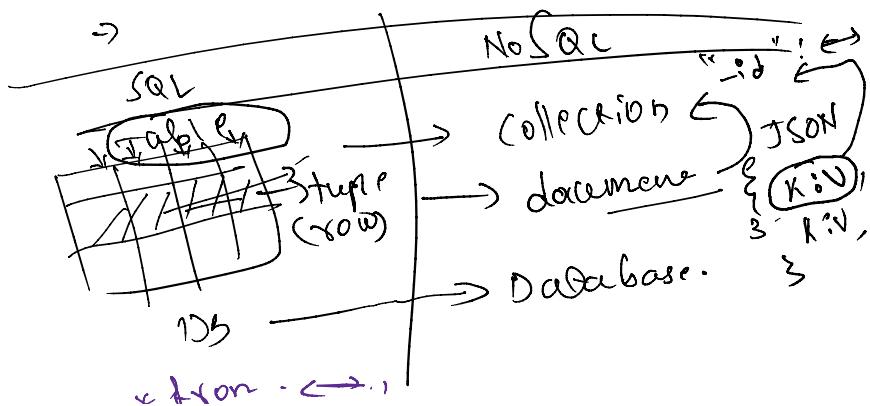
	toy_id	toy_name	price	category
▶	100234	Car	2560	Electronic
	100235	Teddy	560	Fluffed
	100236	Legos	1000	Construction
	100237	Building Set	1585	Construction
	100238	Panda	740	Fluffed
	100239	Helicopter	3600	Electronic
	100240	Snake & Ladder	65	Games

select max(price), category

from toy  
group by category.

NoSQL DB → MongoDB

→ High volume, big data  
→ No transactions()



DB — |  
Select \* from ..

### Mongo DB Commands:

1) show . dbs; → list all database

2) use elephant\_b2-2023 collection name  
↓ toy, producer

3) show collections.  
↓ db. toy. find(); db. toy collection

4) Show all docs. inside collection

db. toy. find(). pretty()

db. producer. find()

5) count()

db. producer. find(). count()  
no. of docs. in collection.

6) Filters

db. producer. find( { K : V, K:V } )

db. toy. findOne({ K,V }) → scans the first match

7) insert doc. onto collection:

db. <collection>. insert ( { . . . } )

db. toy.insert ( { 'toy.name' : 'Chess',  
'category' : 'games',  
'age' : '8+' } )

db. toy. insert many ( [ { . . . }, { . . . }, { . . . } ] )

~~db.toy~~ → ~~{ } ] )~~

To Create Collection

~~db.createCollection('collection')~~  
~~name.~~

Delete

~~db.collection.deleteOne({ })~~

~~.deleteMany({ })~~

Update

~~db.collection.({ <query>, <update> })~~

~~updateOne~~

~~updateMany { } → { }, { \$set: { key } }~~

~~\$set → add / update attribute~~

~~\$unset → remove attribute.~~

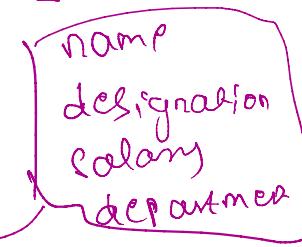
Find lt, gt, lte, gte,

→ find all toys with price  
less than 1000

~~db.toy.find({ 'price': { \$lt: 1000 } })~~

Use mongoSH commands.

Use mongoSH commands.

- \* Create collection 'Employee' 
  - name
  - designation
  - salary
  - department
- \* Insert at least 5-10 documents.
- \* List employees
  - all
  - count
  - based on Project "Telstra"
  - Salary (lt, gt)
- \* update Salary of Employee with designation → 'Manager'