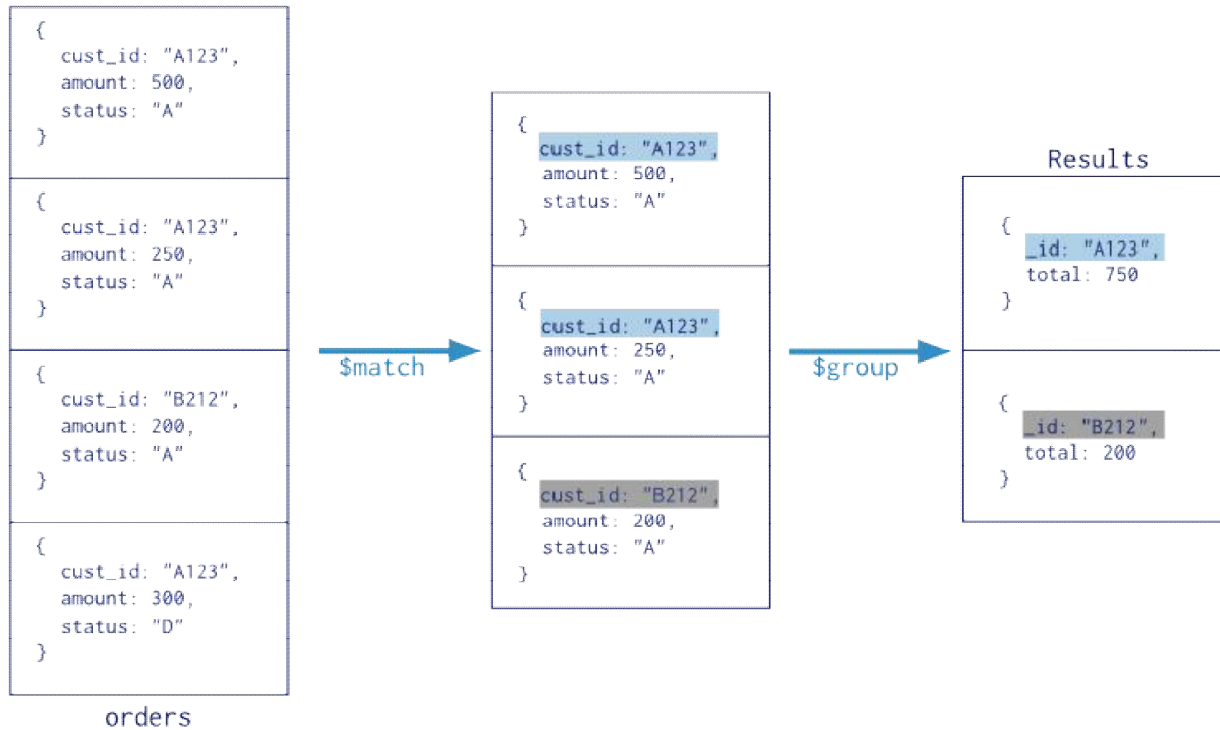


MongoDB Aggregation Framework

Aggregation Pipeline

Collection
↓
db.orders.aggregate([
 \$match stage → { \$match: { status: "A" } },
 \$group stage → { \$group: { _id: "\$cust_id", total: { \$sum: "\$amount" } } }
])



A gyakorlaton használt pipeline operátorok

- \$project
- \$match
- \$group
- \$sort
- \$limit
- És a dokumentáció:

<http://docs.mongodb.org/manual/reference/operator/aggregation/#aggregation-pipeline-operator-reference>

Importáljuk be az adatokat

- `mongoimport --db test --collection hotels --file /tmp/Postgres_json/hotels_example.json`

Belépés:

- `mongo test`
- `db.hotels.findOne()`

Egyszerű lekérdezések

Hány szálloda van a rendszerben?

- `db.hotels.count()`

Hány wellness-szálloda van Texasban?

- `db.hotels.find({state: "TX", wellness: true}).count()`

1.

Melyik az a három állam, ahol a legtöbb olyan szálloda van, amelyeknek nincsen parkolója, és mégis 50% fölötti a kihasználtsága?

1. feladat Pipeline

- `{$match: {parking: false}},`
- Azon szállodák érdekelnek, amiknek nincs parkolója

- `{$project: {_id:1, state:1, pct: {$divide: ["$free", "$rooms"]}}},`
- Megtartjuk az `_id`-t, a `state`-et, illetve `pct` néven behozunk egy új (számított) attribútumot, ami a `free/rooms` hányadosként áll elő

1. feladat Pipeline

- `{$match: {pct: {$gt: 0.5}}}`,
- Ezekből továbbengedjük azokat, ahol a pct 0.5 fölött van

- `{$group: {_id: "$state", n: {$sum: 1}}}`,
- Csoportosítunk, a csoportosítás kulcsa a state attribútum, és az egyes eredmény-rekordoknak lesz még egy n nevű attribútuma, ami úgy áll elő, hogy az adott state kódú bemenő rekordokra szummázzuk az 1-et, azaz megszámloljuk az ilyen szállodákat

1. feladat Pipeline

- `{$sort: {n: -1}}`,
- A szállodaszám szerinti csökkenő sorrendből érdekel minket....

- `{$limit: 3}`,
- Az első három.

1. feladat megoldás

- `db.hotels.aggregate([{$match: {parking: false}}, {$project: {_id:1, state:1, pct: {$divide: ["$free", "$rooms"]}}}, {$match: {pct: {$gt: 0.5}}}, {$group: {_id: "$state", n: {$sum: 1}}}, {$sort: {n: -1}}, {$limit: 3},])`

2. Listázzuk ki azon öt város, ahol a legtöbb szabad szoba van olyan szállodákban, ahol van parkolóhely és engednek állatokat is.

2. feladat Pipeline

- `{$match: {parking: true, pets: true}}`,
- Csak a parkolós-állatos szállodák érdekelnek

- `{$group: {_id: "$city", total_free: {$sum: "$free"}}`,
- Csoportosítunk a city mentén, az eredményekben legyen egy új total_free nevű attr, ami úgy áll össze, hogy a csoport tagjain szummázzuk a free-t

2. feladat Pipeline

- `{sort: {total_free: -1}}, {limit: 5},`
- Ebből kérjük a csökkenő sor első 5 elemét

2. feladat megoldás

- `db.hotels.aggregate([{$match: {parking: true, pets: true}}, {$group: {_id: "$city", total_free: {$sum: "$free"}}}, {$sort: {total_free: -1}}, {$limit: 5},])`

3.

Listázzuk ki a 20 legnagyobb szállodából a 10 legkevésbé kihasználtak a városait.

3. feladat Pipeline

- `{sort: {rooms: -1}}, {limit: 20},`
- A 20 legnagyobb szálloda érdekel csak minket
- `{project: {city: 1, pct: {divide: ["free", "rooms"]}}},`
- Megtartjuk a city attribútumot, és behozunk egy új számított attribútumot a kihasználtság ábrázolására: `pct := free/rooms`

3. feladat Pipeline

- `{ $sort: { pct: 1 } }, { $limit: 10 },`
- A fentiek közül a legkevésbé kihasznált 10 szálloda érdekel csak

- `{ $group: { _id: "$city" }`
- Csoportosítunk a city mentén, és másra nem is vagyunk kíváncsiak

3. feladat megoldás

- `db.hotels.aggregate([{$sort: {rooms: -1}}, {$limit: 20}, {$project: {city: 1, pct: {$divide: ["$free", "$rooms"]}}}, {$sort: {pct: 1}}, {$limit: 10}, {$group: {_id: "$city"}}])`

4. feladat

- Melyik az az öt legnagyobb Houstoni szálloda(TX), aminek van parkolója és van legalább 10 szabad szobája?

4. feladat megoldás

- `db.hotels.find({state: "TX", city: "Houston", parking: true, free: { $gte: 10}}).sort({rooms: -1}).limit(5)`

5. feladat

- Melyik az a tíz legkevésbé telített texasi szálloda, ahol van parkoló és engednek állatokat?

5. feladat megoldás

- `db.hotels.aggregate([{$match:{state: "TX",
pets: true, parking: true}}, {$project:{_id:1,
state:1, city:1, name:1, address:1, free_pct:
{$divide:["$free", "$rooms"]}},
{$sort:{free_pct:-1}}, {$limit: 10}])`

6. feladat

- Melyek azok a texasi városok, ahol az ottani wellness-szállodák átlagos kihasználtsága szerint csökkenő sorrendben?

6. feladat megoldás

- `db.hotels.aggregate([{$match:{state: "TX", wellness: true}}, {$project:{_id:1, city:1, free_pct: {$divide:["$free", "$rooms"]}}}, {$group:{_id:"$city", avg_pct:{$avg:"$free_pct"}}, {$sort:{avg_pct:-1}},])`