

DAMG 6210: Data Management and Database Design: Lab 1

Part 1 (5 points)

- Enter the normalized entities and attributes from the first part of the Normalization Exercise 8 in your design tool. You don't need to add any extra entity or attribute. (20% of grade)
- Establish the relationships between entities using the Crow's Foot notation. Please make sure each relationship line is completely visible and don't cross lines. (80% of grade)

Note: Don't add any unnecessary relationship. An unnecessary relationship clutters the design and may degrade the database and application performance.

- Upload the ERD as an image or in the pdf format to here.

Part 2 (4 points)

Our data usage pattern is to find the average quarterly sold quantity of a color. Use the embedding design technique and the provided data to design a MongoDB database. Create the database on MongoDB Atlas that includes document(s) reflecting the data contained in the attached file. (50% of grade)

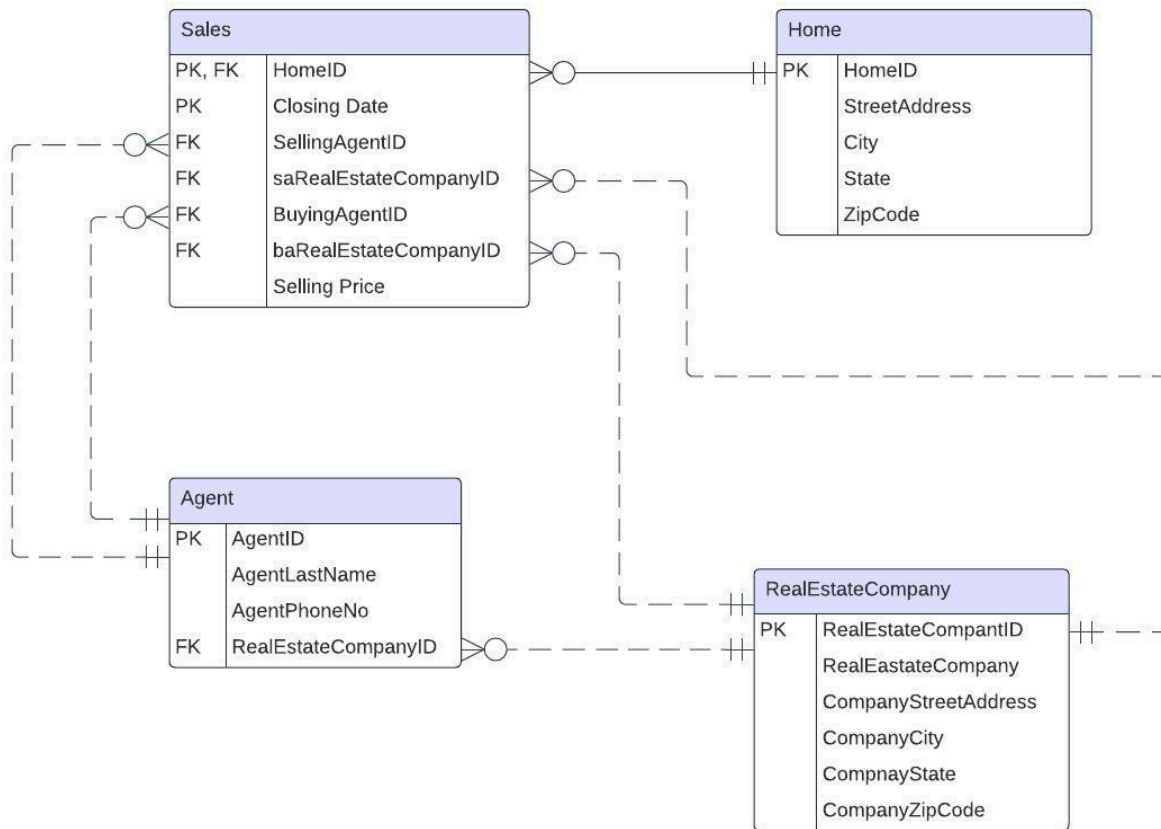
Then use the MongoDB Compass, JavaScripts and MongoDB Aggregation Pipeline to calculate the average quarterly sold quantity of the blue color. (50% of grade)

Submit your code, and a screenshot of the created documents and code execution results in the pdf format.

Color	Quarter	Sold Quantity
Blue	1	230
Blue	2	452
Blue	3	6351
Blue	4	5280
Red	1	2453
Red	2	3486
Red	3	1535
Red	4	1250
Silver	1	814
Silver	2	1039
Silver	3	5788
Silver	4	4511

Part 1: ERD - Normalization Exercise 8 (Crow's Foot Notation)

Track each Sale of Home



Assumptions:

- Only 1 Selling Agent & 1 Buying Agent for each sale
- Only the Last Name of Agent is tracked
- Only 1 Phone Number of Agent is tracked
- 1 Agent is associated with only 1 Real Estate Company
- A Home can exist without any of its Sales
- A Real Estate company can exist without any Agent present in the company
- An Agent can exist without participating in a Selling Sale
- An Agent can exist without participating in a Buying Sale
- A Real Estate company can exist without participating in a Selling Sale
- A Real Estate company can exist without participating in a Buying Sale

Part 2: MongoDB - Design & Aggregation Pipeline

Data usage pattern: Find the average quarterly sold quantity of a color

(Output: Average Quarterly Sold Quantity of Blue Color = 3078.25)

Embedded Design MongoDB Document:

Color: Blue

MongoDB Compass - cluster0.whic0sq.mongodb.net/color_sales.color_totals

cluster0.whic0sq... My Queries color_sales color_totals

color_sales.color_totals 3 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

ADD DATA EXPORT DATA UPDATE DELETE 1 - 3 of 3

```
{
  "_id": ObjectId("65bb08a5217083362c5f8538"),
  "color": "Blue",
  "sales": Array (4)
    0: Object
      quarter: 1
      sold_quantity: 230
    1: Object
      quarter: 2
      sold_quantity: 452
    2: Object
      quarter: 3
      sold_quantity: 6351
    3: Object
      quarter: 4
      sold_quantity: 5280
  }
}
```

> MONGOSH

Temps to rise Tomorrow

Color: Red

MongoDB Compass - cluster0.whic0sq.mongodb.net/color_sales.color_totals

cluster0.whic0sq... My Queries color_sales color_totals

color_sales.color_totals 3 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

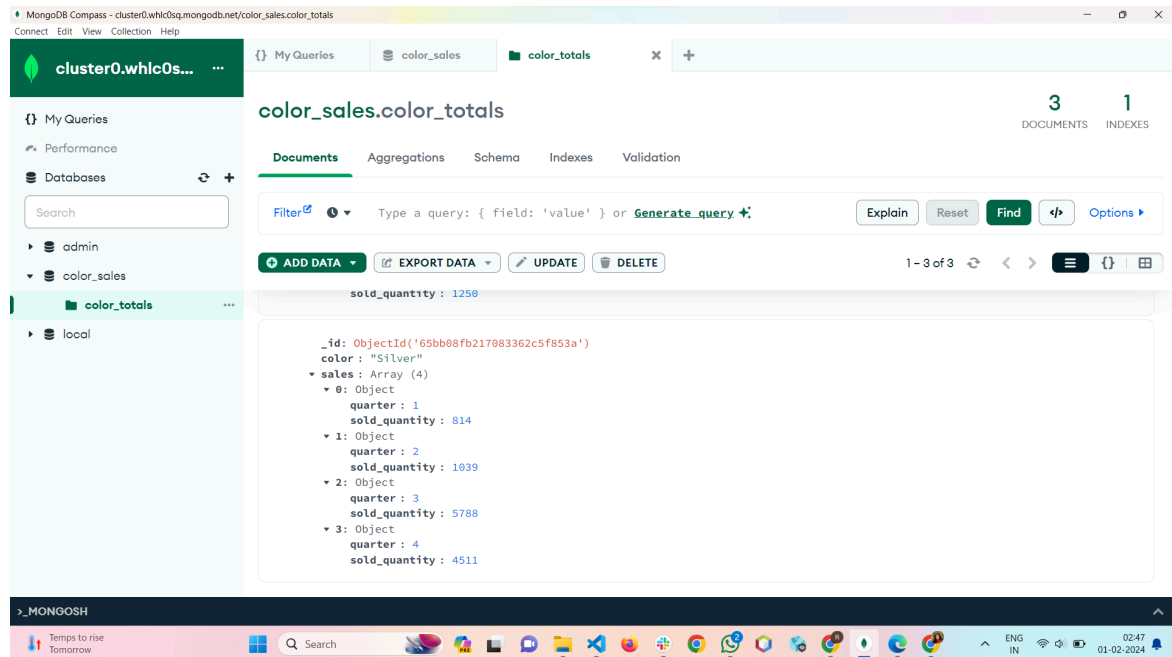
ADD DATA EXPORT DATA UPDATE DELETE 1 - 3 of 3

```
{
  "_id": ObjectId("65bb08d8217083362c5f8539"),
  "color": "Red",
  "sales": Array (4)
    0: Object
      quarter: 1
      sold_quantity: 2453
    1: Object
      quarter: 2
      sold_quantity: 3486
    2: Object
      quarter: 3
      sold_quantity: 1535
    3: Object
      quarter: 4
      sold_quantity: 1250
  }
}
```

> MONGOSH

Temps to rise Tomorrow

Color: Silver



JavaScript Code: To find the average quarterly sold quantity of blue color

```
[
  {
    '$match': {
      'color': 'Blue'
    }
  }, {
    '$project': {
      'quarterly_sold_quantities': '$sales.sold_quantity'
    }
  }, {
    '$unwind': {
      'path': '$quarterly_sold_quantities'
    }
  }, {
    '$group': {
      '_id': 1,
      'avg_quarterly_sold_quantity': {
        '$avg': '$quarterly_sold_quantities'
      }
    }
  }
]
```

MongoDB Aggregation Pipeline

Preview of Documents

cluster0.whlc0s...

My Queries

Performance

Databases

Search

admin

color_sales

color_totals

local

My Queries

color_sales

color_totals

color_sales.color_totals

31

DOCUMENTSINDEXES

DocumentsAggregationsSchemaIndexesValidation

Pipeline

SmatchProjectSunwindSgroup

Generate aggregation

Explain

Export

Run

Options

Average Quar...

SAVE

CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

STAGES

TEXT

WIZARD

3 Documents in the collection

Preview of documents

_id: ObjectId('65bb08a5217083362c5f8538')

color: "Blue"

sales: Array (4)

0: Object

quarter: 1

sold_quantity: 230

1: Object

quarter: 2

sold_quantity: 230

_id: ObjectId('65bb08d8217083362c5f8539')

color: "Red"

sales: Array (4)

0: Object

quarter: 1

sold_quantity: 2453

1: Object

quarter: 2

sold_quantity: 2453

_id: ObjectId('65bb08fb217083362c5f853a')

color: "Silver"

sales: Array (4)

0: Object

quarter: 1

sold_quantity: 814

1: Object

quarter: 2

sold_quantity: 814

Stage1

Smatch

Output after Smatch stage (Sample of 1 document)

1 /**

2 * query: The query in MQL.

3 */

sold_quantity: 230

Temp to rise Tomorrow

Search

ENG IN

02:58

01-02-2024

cluster0.whlc0s...

My Queries

Performance

Databases

Search

admin

color_sales

color_totals

local

color_sales.color_totals

31

DOCUMENTSINDEXES

DocumentsAggregationsSchemaIndexesValidation

Pipeline

SmatchProjectSunwindSgroup

Generate aggregation

Explain

Export

Run

Options

Average Quar...

SAVE

CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

STAGES

TEXT

WIZARD

3 Documents in the collection

Preview of documents

sold_quantity: 230

1: Object

quarter: 2

sold_quantity: 452

2: Object

quarter: 3

sold_quantity: 6351

3: Object

quarter: 4

sold_quantity: 230

1: Object

quarter: 2

sold_quantity: 3486

2: Object

quarter: 3

sold_quantity: 1535

3: Object

quarter: 4

sold_quantity: 1250

sold_quantity: 647

1: Object

quarter: 2

sold_quantity: 1039

2: Object

quarter: 3

sold_quantity: 5788

3: Object

quarter: 4

sold_quantity: 4511

Stage1

Smatch

Output after Smatch stage (Sample of 1 document)

1 /**

2 * query: The query in MQL.

3 */

sold_quantity: 230

Temp to rise Tomorrow

Search

ENG IN

02:58

01-02-2024

Stage 1: \$match - Filters the documents to pass only the documents that match the specified condition(s) to the next pipeline stage.

The screenshot displays the MongoDB Compass interface for the 'cluster0.whlc0s...' cluster, specifically the 'color_sales.color_totals' collection. The 'Aggregations' tab is active, showing a pipeline with two stages: '\$match' and '\$project'.

Stage 1: \$match

- Query (MQL):

```
1 /**
2  * query: The query in MQL.
3  */
4 {
5   color: "Blue"
6 }
```
- Output after \$match stage (Sample of 1 document):

```
{
  "_id": ObjectId('65bb8a5217883362c5f8538'),
  "color": "Blue",
  "sales": Array (4)
    0: Object
      quarter: 1
      sold_quantity: 238
    1: Object
      quarter: 2
      sold_quantity: 452
```

Stage 2: \$project

- Output after \$project stage (Sample of 1 document):

```
{
  "sold_quantity": 452,
  "quarter": 2
}
```

The interface also shows a sidebar with 'My Queries', 'Performance', 'Databases', and 'color_totals'. The top bar indicates 3 documents and 1 index.

Stage 2: \$project - Passes along the documents with the requested fields to the next stage in the pipeline.

The screenshot shows the MongoDB Compass interface for the 'color_sales' database, specifically the 'color_totals' collection. The 'Aggregations' tab is active, and the pipeline consists of four stages: \$match, \$project, \$unwind, and \$group. Stage 2, '\$project', is selected and expanded. The left pane shows the JSON query:

```
1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   "quarterly_sold_quantities":
7   "$sales.sold_quantity"
8 }
```

 The right pane shows the 'Output after \$project stage (Sample of 1 document)' as a JSON object:

```
{
  "_id": ObjectId('65bb08a5217083362c5f8538'),
  "quarterly_sold_quantities": Array (4)
    0: 230
    1: 452
    2: 6351
    3: 5280
}
```

 The bottom status bar shows the system time as 02:59 on 01-02-2024.

Stage 3: \$unwind - Deconstructs an array field from the input documents to output a document for each element. Each output document is the input document with the value of the array field replaced by the element.

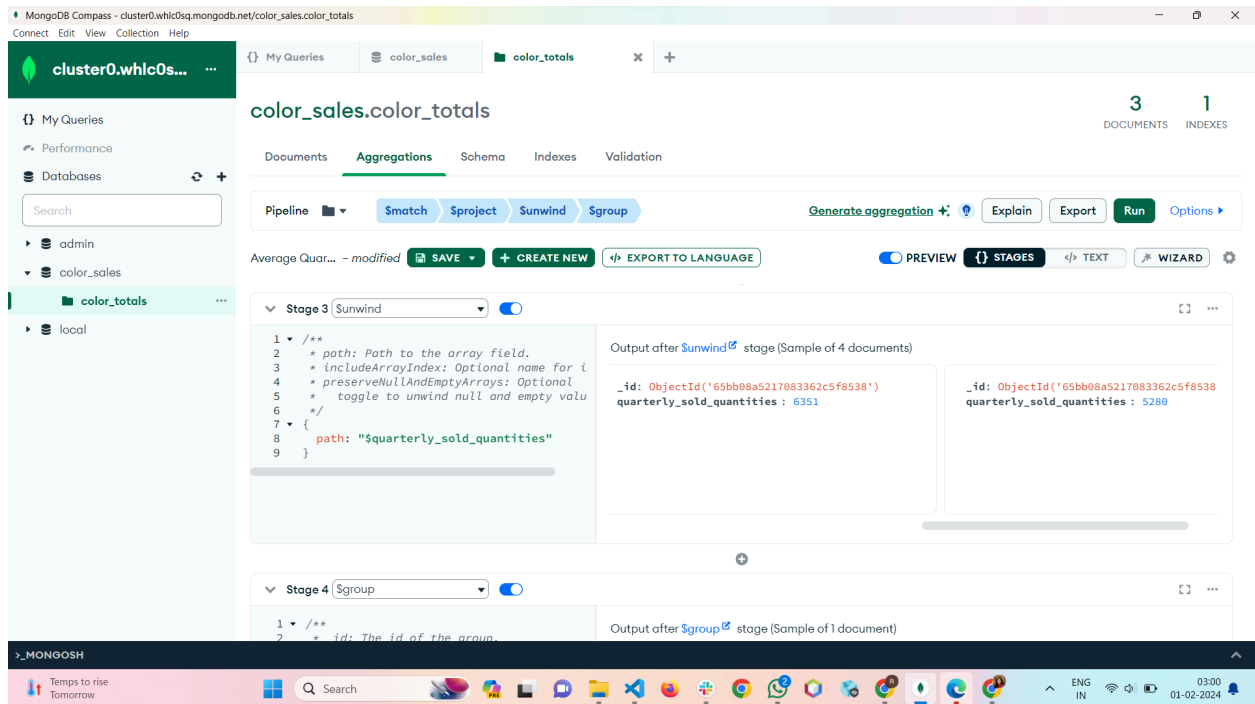
The screenshot shows the MongoDB Compass interface for the 'color_sales' database, specifically the 'color_totals' collection. The 'Aggregations' tab is active, and the pipeline consists of four stages: \$match, \$project, \$unwind, and \$group. Stage 3, '\$unwind', is selected and expanded. The left pane shows the JSON query:

```
1 /**
2  * path: Path to the array field.
3  * includeArrayIndex: Optional name for i
4  * preserveNullAndEmptyArrays: Optional
5  * toggle to unwind null and empty val
6  */
7 {
8   path: "$quarterly_sold_quantities"
9 }
```

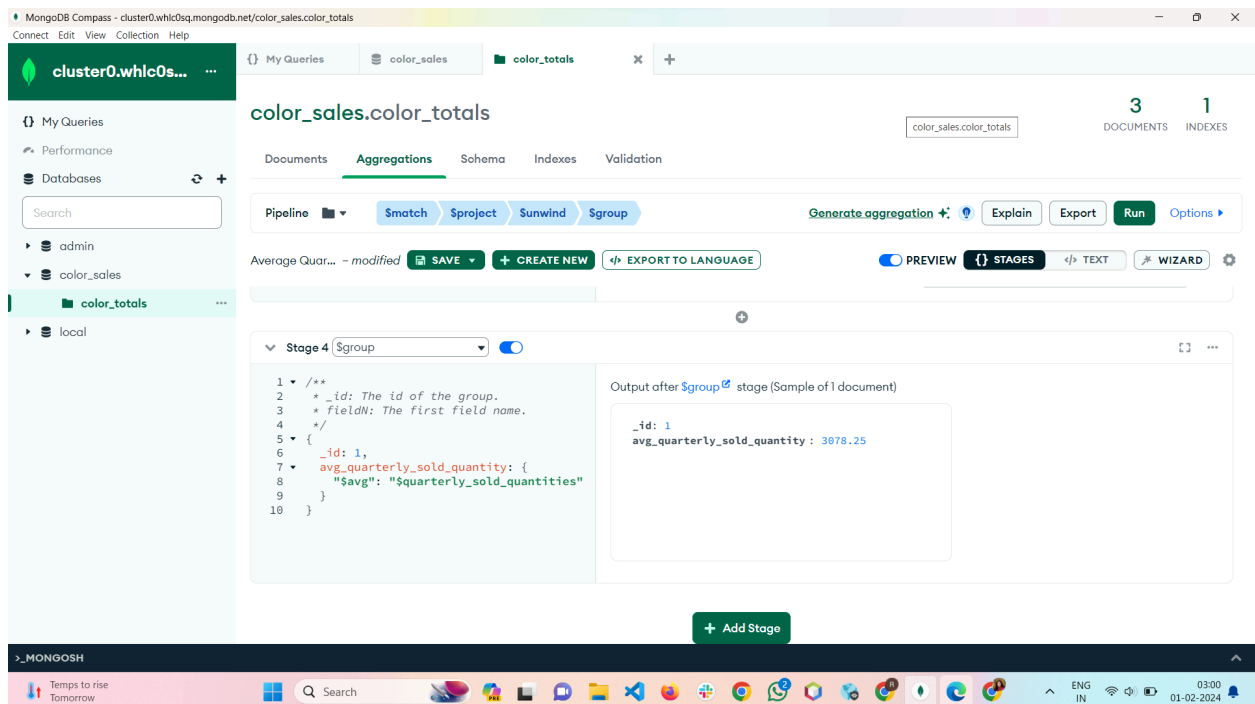
 The right pane shows the 'Output after \$unwind stage (Sample of 4 documents)' as two JSON objects:

```
{
  "_id": ObjectId('65bb08a5217083362c5f8538'),
  "quarterly_sold_quantities": 230
}
{
  "_id": ObjectId('65bb08a5217083362c5f8538'),
  "quarterly_sold_quantities": 452
}
```

 The bottom status bar shows the system time as 03:00 on 01-02-2024.



Stage 4: \$group - Separates documents into groups according to a "group key". The output is one document for each unique group key.



Final Output: Average Quarterly Sold Quantity of Blue Color = 3078.25

References

- [Professor's Normalization Playlist](#)
- [Professor's slides - ERD Example](#)
- [Professor's MongoDB Playlist](#)
- [MongoDB Documentation](#)