Slide Deck: MongoDB Aggregation Pipeline - Careem Analytics Project

---

Slide 1: Title

Aggregation Pipeline in MongoDB\ *Careem Analytics API Example*\ **Instructor**: [Your Name]\ **Date**: August 2025

Slide 2: What is Aggregation?

Aggregation is a powerful feature in MongoDB used to process and transform data in stages.\ Each stage performs an operation on the input documents and passes the results to the next stage, like a pipeline.

Use cases:

- Analytics dashboards
- Reporting
- Summarizing data

Slide 3: Why Use Aggregation?

- Allows complex data transformations
- Replaces multiple queries with one pipeline
- Useful for joins, statistics, and real-time reporting

Slide 4: Aggregation Pipeline Syntax

```
db.collection.aggregate([
  { stage1 },
  { stage2 },
  ...
]);
```

Each `{ stage }` is an object with a key like `$match`, `$group`, etc.

Slide 5: \$match

Definition: Filters documents by given condition (similar to `find`)

```
{ $match: { city: "Karachi" } }
```

Use Case: Get only rides in Karachi.

Slide 6: \$project

Definition: Reshapes the document, includes/excludes fields

```
{ $project: { _id: 0, riderId: 1, fare: 1, city: 1 } }
```

Use Case: Return only selected fields.

Slide 7: \$group

Definition: Groups documents by a field and performs aggregation

```
{
  $group: {
    _id: "$captainId",
    totalEarnings: { $sum: "$fare" },
    averageRating: { $avg: "$rating" }
  }
}
```

Use Case: Calculate captain earnings and average rating.

Slide 8: \$sort

Definition: Sorts results by a field (1 = ASC, -1 = DESC)

```
{ $sort: { totalEarnings: -1 } }
```

Use Case: Show highest earning captains first.

Slide 9: \$limit & \$skip

Definition: Limit the number of documents and skip a given number

```
{ $skip: 10 }, { $limit: 5 }
```

Use Case: Implement pagination.

Slide 10: \$lookup

Definition: Performs a join between two collections

```
{
  $lookup: {
    from: "captains",
    localField: "captainId",
    foreignField: "_id",
    as: "captain"
  }
}
```

Use Case: Enrich ride with captain info.

Slide 11: \$unwind

Definition: Deconstructs an array into multiple documents

```
{ $unwind: "$deliveryStops" }
```

Use Case: Flatten delivery route stops.

Slide 12: \$facet

Definition: Runs multiple pipelines in parallel and returns combined output

```
{
  $facet: {
    stats: [ { $group: { _id: "$city", total: { $sum: 1 } } } ],
    data: [ { $skip: 0 }, { $limit: 5 } ]
  }
}
```

Use Case: Pagination + summary in one query.

Slide 17: Summary

You learned:

- Aggregation pipeline structure
- Each core stage and its usage
- Practical use in a rides/delivery system

Slide 18: Practice Task

"Find the top 3 busiest cities in terms of completed rides."

- Use `$match`, `$group`, `$sort`, `$limit`
