

Week 5 Exercise: MongoDB Data Modelling & ERD Design

Objective: Design a MongoDB schema for the ride-hailing system by identifying entities, relationships, and document structures. Create an ERD using crow's foot notation to visualize the database architecture.

Lab Overview

Key Topics

1. **Data Modelling:** Identify collections, document structures, and relationships.
2. **Embedding vs. Referencing:** Decide when to nest documents or use references.
3. **ERD Design:** Visualize entities and relationships with crow's foot notation.

Deliverables

1. ERD diagram for the ride-hailing system.
 2. MongoDB schema definitions (collections, sample documents).
 3. Answers to schema design questions.
-

Lab Procedures

Part 1: Identify Entities & Relationships

Task 1: List Core Entities

Based on the e-hailing system's use cases (Week 4), identify entities (example):

- **User** (Customers, Drivers, Admins)
- **Ride**
- **Vehicle** (linked to Drivers)
- **etc**

Task 2: Define Attributes

For each entity, list fields (example):

User:

- `_id` (ObjectId)
- `email` (String, unique)
- `password` (String)
- `role` (String: "customer", "driver", "admin")
- `createdAt` (Date)

Ride:

- `_id` (ObjectId)
- `customerId` (ObjectId → User)
- `driverId` (ObjectId → User)
- `pickupLocation` (Object: { `lat`: Number, `lng`: Number })
- `destination` (Object: { `lat`: Number, `lng`: Number })
- `status` (String: "requested", "in_progress", "completed")
- `fare` (Number)

Task 3: Model Relationships

Decide how entities relate (example):

- **User** → **Ride**: One-to-Many (1 user can have many rides).
- **Driver** → **Vehicle**: One-to-One (1 driver has 1 vehicle).

Part 2: Design the ERD

Task 4: Draw the ERD

Use draw.io or Lucidchart to create an ERD with crow's foot notation.

Submission Requirements

1. **ERD Diagram**: Image file or link.
2. **Schema Definitions**:
 - List of collections, fields, and data types.