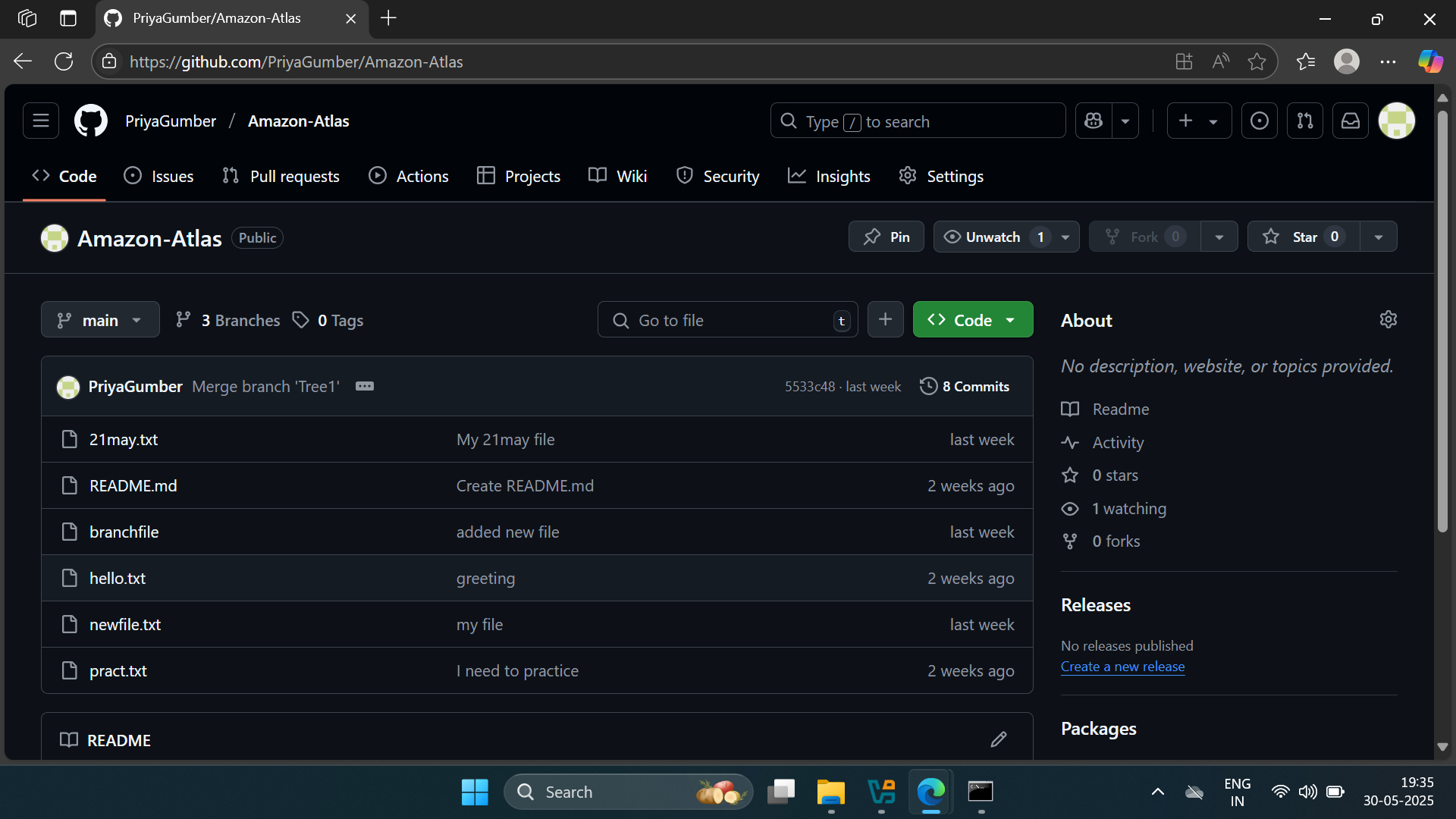
Day 3 , Date : 30/05/2025 , Emp Id : 112916146

**Task to commit to git remote file named may 21:**



Task 2 : ACID Properties of database systems.

### 🔹 Atomicity

Definition: Ensures that all operations within a transaction are completed; if any fail, the entire transaction is rolled back.

Example: Transferring ₹1,000 from Account A to Account B will either debit and credit both accounts or neither, preventing partial updates.

### 🔹 Consistency

Definition: Guarantees that a transaction brings the database from one valid state to another, maintaining all predefined rules.

Example: A transaction that violates a rule—like overdrawing an account beyond its limit—is rejected to preserve data integrity.

### 🔹 Isolation

Definition: Ensures that concurrent transactions execute independently without interference, leading to the same state as if run sequentially.

Example: Two users booking the last available train seat simultaneously will have their transactions isolated to prevent double booking.

### 🔹 Durability

Definition: Once a transaction is committed, its results are permanent, even in the event of a system failure.

Example: After booking a hotel room and receiving confirmation, the reservation remains intact despite any subsequent system crash.

These ACID properties are fundamental to ensuring reliable and consistent database transactions, particularly in critical applications like banking, reservations, and e-commerce.

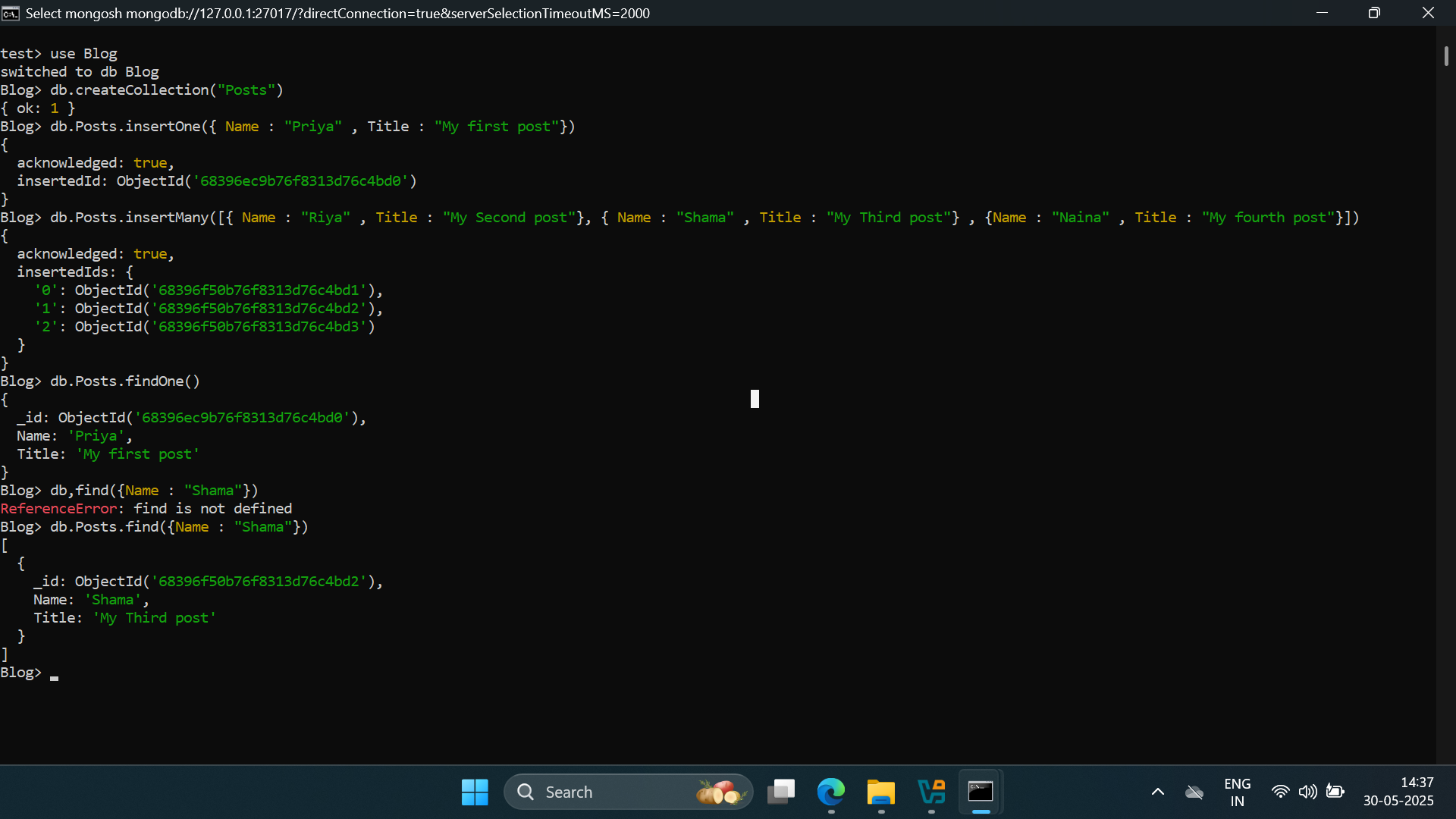
MonGo DB CRUD Operations .

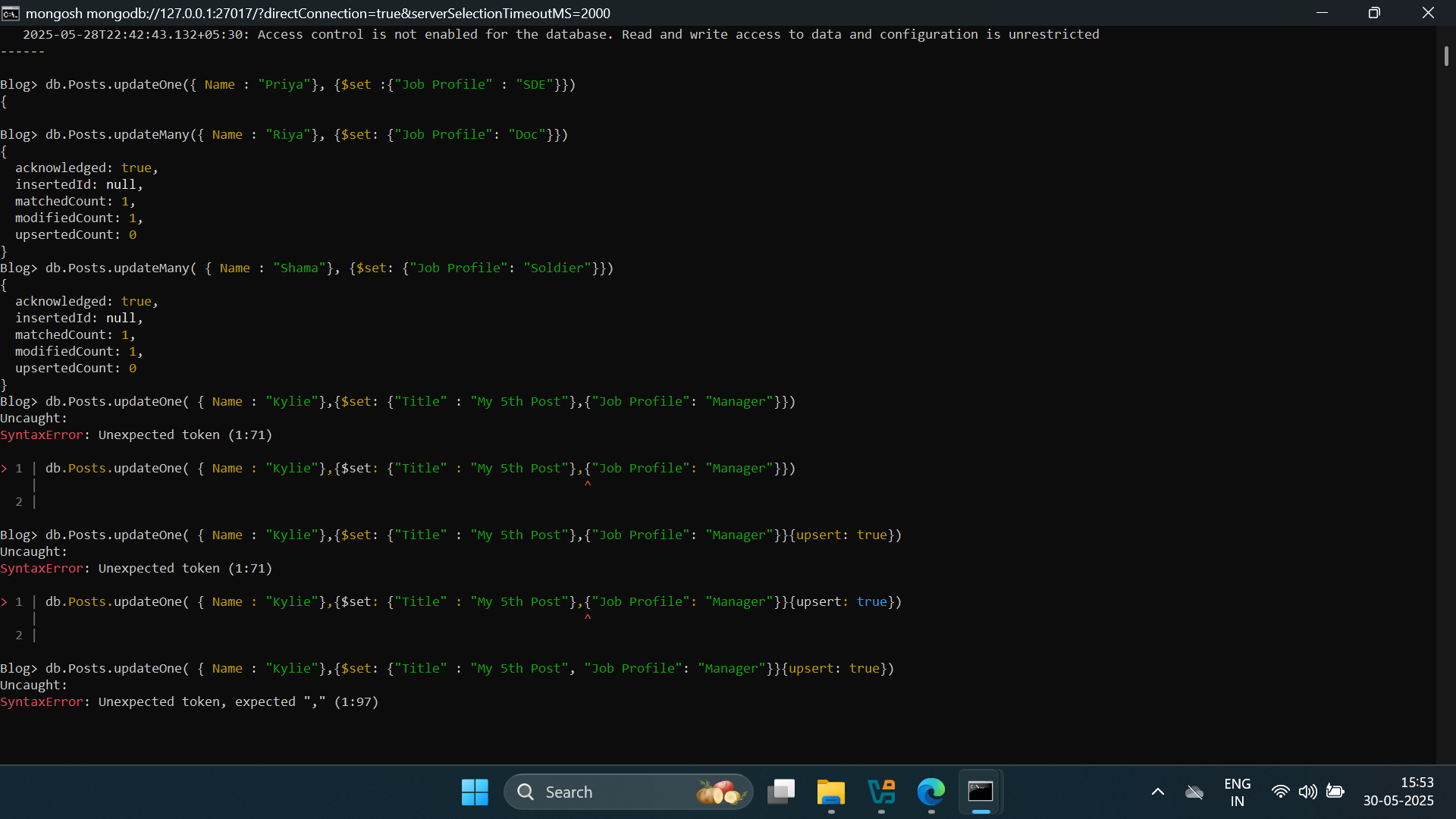
1. Step 1: Create and Use a Database
2. Step 2: Create a Collection

## Step 3: Insert Documents

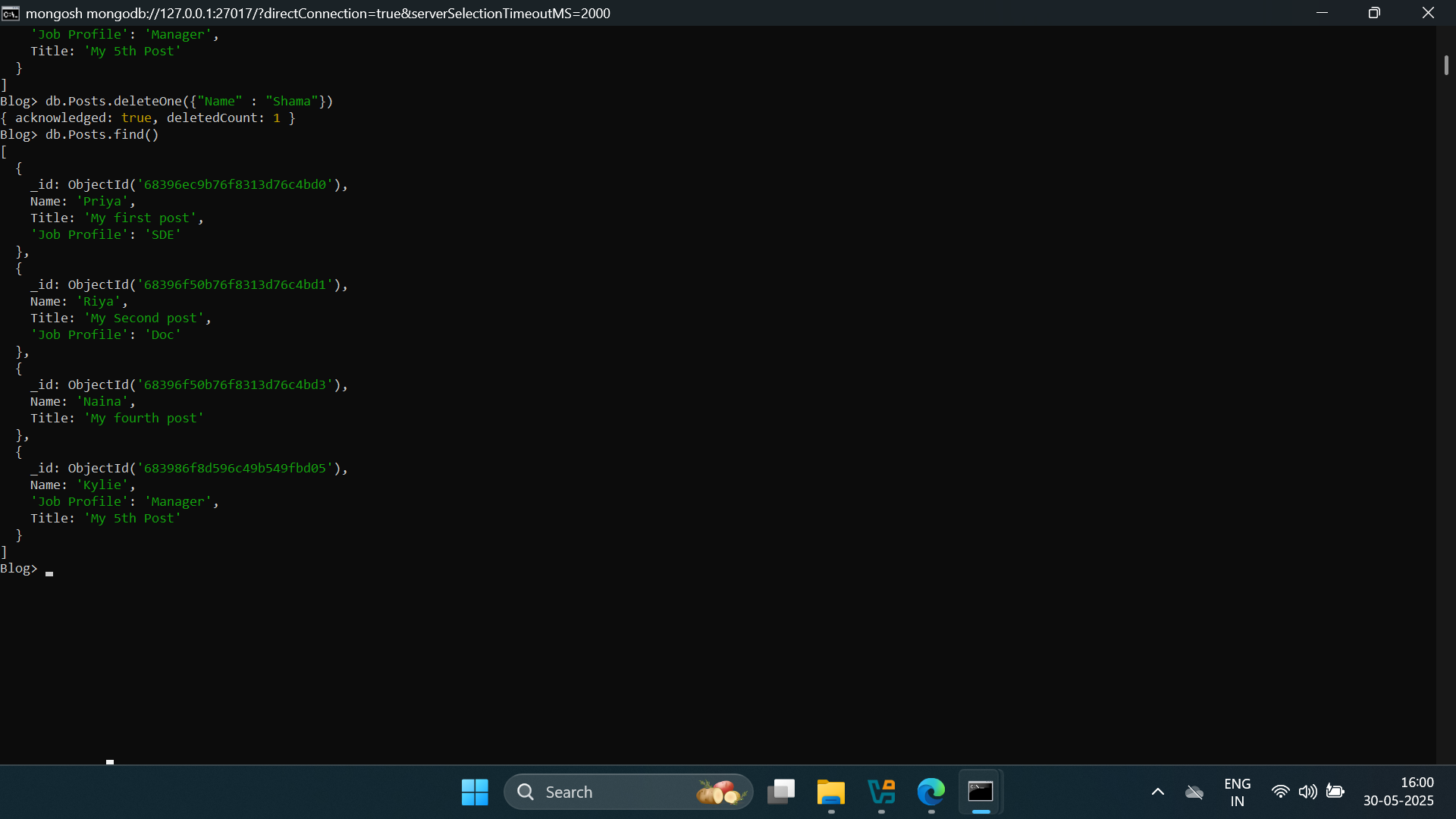
### Insert a Single Document

### Insert Multiple Document

1. Use FindOne() and Find() with arguments 
2. Update a Single Document



8.Delete Documents



## Comparison Operators

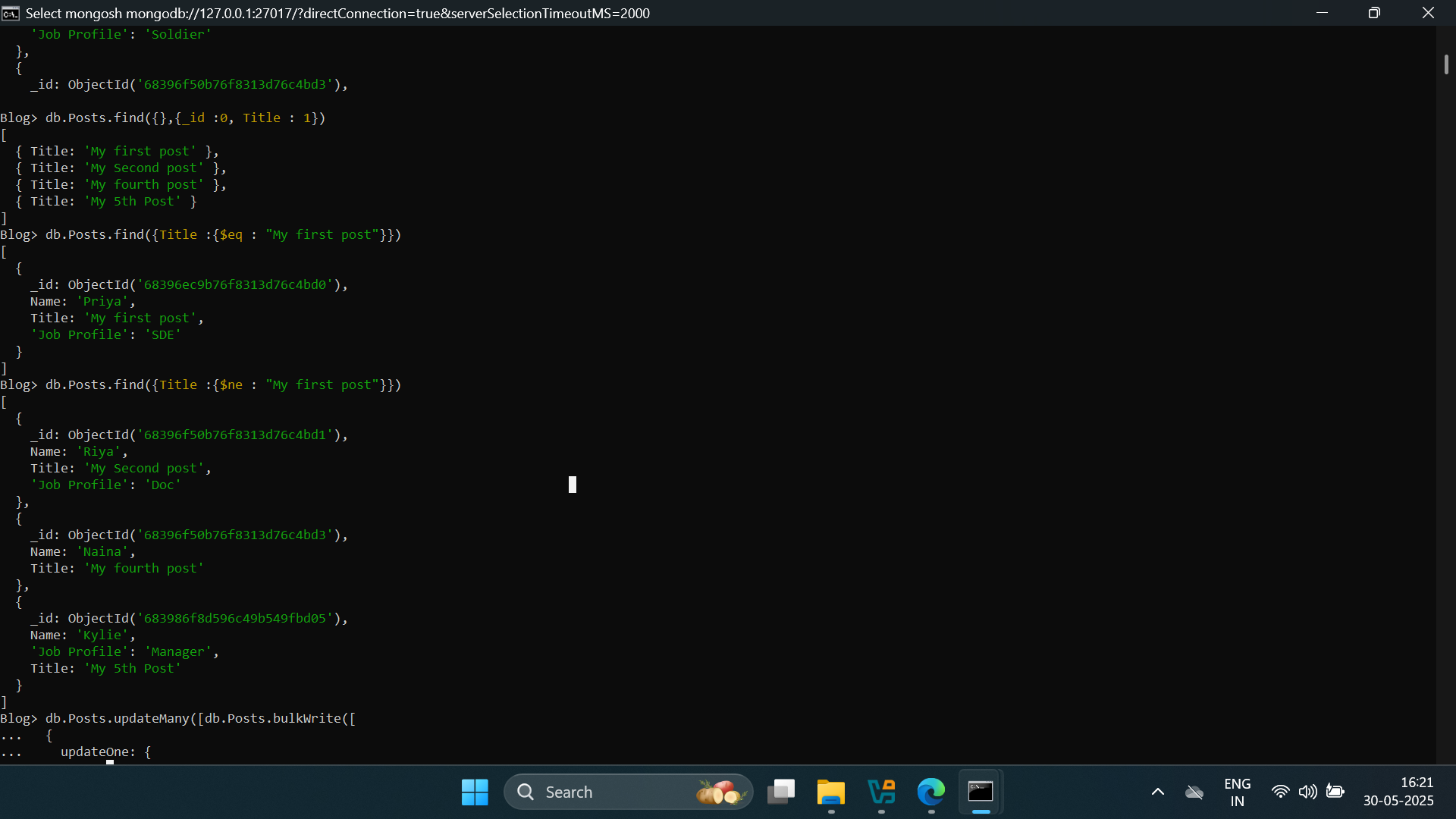
These operators compare field values to specified criteria.

### 1. $eq – Equal To

Syntax: { field: { $eq: value } }

2.2. $ne – Not Equal To

Syntax: { field: { $ne: value } }

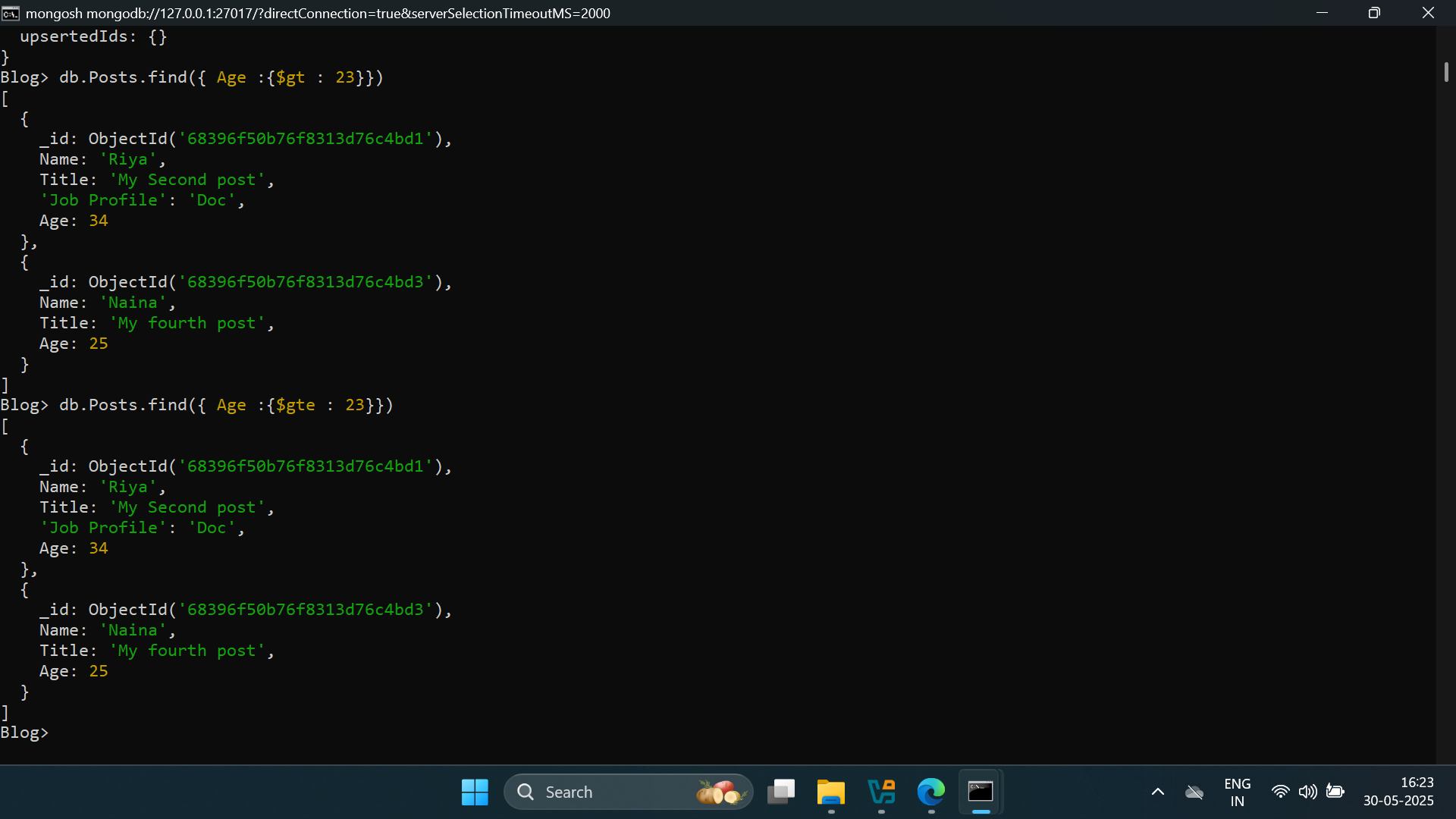


### 3. $gt – Greater Than

Syntax: { field: { $gt: value } }

### 4. $gte – Greater Than or Equal To

Syntax: { field: { $gte: value } }

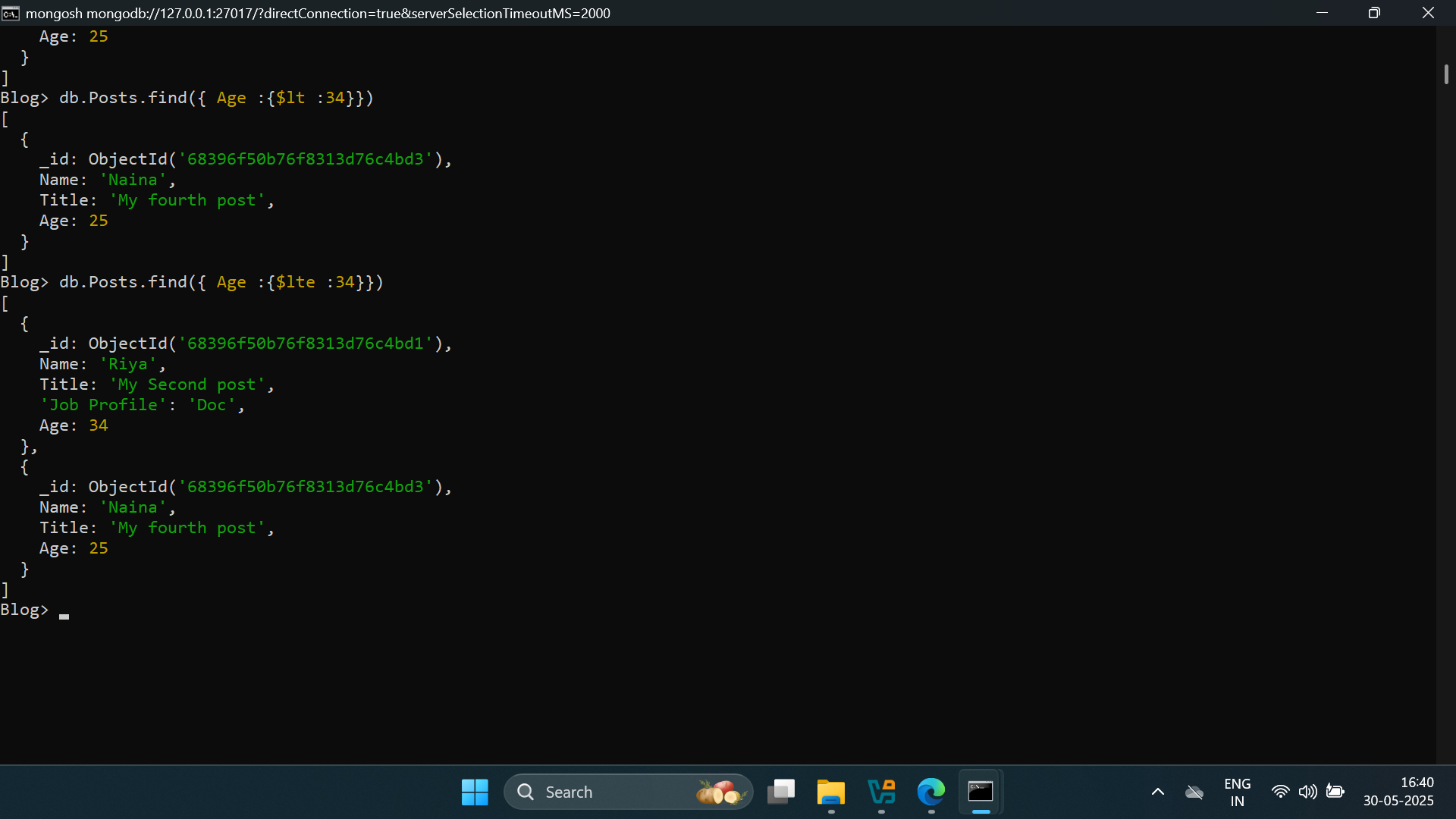


### 5. $lt – Less Than

Syntax: { field: { $lt: value } }

### 6. $lte – Less Than or Equal To

Syntax: { field: { $lte: value } }



## Capped Collections in MongoDB

### What Are Capped Collections?

Capped collections are fixed-size collections that maintain the insertion order of documents and automatically overwrite the oldest documents when the specified storage limit is reached.

