Session 3

Where, AND, OR & CRUD

Experiment 1

WHERE

- Given a Collection you want to FILTER a subset based on a condition. That is the place WHERE is used.

```
// Find all students with GPA greater than 3.5
db.students.find({ gpa: { $gt: 3.5 } });

// Find all students from "City 3"
db.students.find({ home_city: "City 3" });
```

AND

- Given a Collection you want to FILTER a subset based on multiple conditions

```
// Find all students who live in "City 5" AND have a blood group of "A+"
db.students.find({
  $and: [
    { home_city: "City 5" },
    { blood_group: "A+" }
  ]
});
```

mongoDB

OR

- Given a Collection you want to FILTER a subset based on multiple conditions but Any One is Sufficient

```javascript
// Find all students who are hotel residents OR have a GPA less than 3.
db.students.find({
  $or: [
    { is_hotel_resident: true },
    { gpa: { $lt: 3.0 } }
  ]
});
```

CRUD

- C - Create / Insert
- R - Remove
- U - update
- D - Delete

This is applicable for a Collection (Table) or a Document (Row)

Insert

```javascript
// Define the student data as a JSON document
const studentData = {
  "name": "Alice Smith",
  "age": 22,
  "courses": ["Mathematics", "Computer Science", "English"],
  "gpa": 3.8,
  "home_city": "New York",
  "blood_group": "A+",
  "is_hotel_resident": false
};

// Insert the student document into the "students" collection
db.students.insertOne(studentData);
```

Update

```javascript
// Find a student by name and update their GPA
db.students.updateOne({ name: "Alice Smith" }, { $set: { gpa: 3.8 } });
```

Delete

```javascript
// Delete a student by name
db.students.deleteOne({ name: "John Doe" });
```

Update Many

```javascript
// Update all students with a GPA less than 3.0 by increasing it by 0.5
db.students.updateMany({ gpa: { $lt: 3.0 } }, { $inc: { gpa: 0.5 } });
```

mongoDB

Delete Many

```
// Delete all students who are not hotel residents
db.students.deleteMany({ is_hotel_resident: false });
```

Projection

This is used when we don't need all columns/attributes

```
// Get only the name and gpa for all students
db.students.find({}, { name: 1, gpa: 1 });

// Exclude the "_id" field from all queries by default
db.students.find({}, { _id: 0 });
```