Database: organized collection of related data.

MySQL – is a relational database

MySQL – Is open source.

ID	Name	Date of Birth	Phone
1	John	09/11/2000	0000000000000

Relational DB: data is store in tabular form. (row & column). To make it simple data is store in table.

DBMS – Database Management System – software which help users to create and maintain the database. DBMS is a software not DB

DBMS use SLQ – Structured Query Language as backend for manipulation (create, populated, update, delete, drop, and maintain a database) of a database.

DBMS Fulfils several purpose:

1. Creating
2. Defining database
3. Updating database
4. Deleting/ dropping database
5. Searching database
6. Handle security
7. Facilitate backup/ restore facility
8. Giving different privileges to users

CRUD – Create, Read/ retrieve, Update, Delete

Application Programme: Phyton, JAVA, C# - It send queries to database .

Database can be any size.

1. Amazon, YouTube – are large website
2. Database in an organization – medium and small

Relational DB – MySQL, Microsoft SQL Server, Oracle	**Non-Relational** DB -MongoDB, Cassandra
Data is stored into tables.	There is no table, data is not linked together.
Data of multiple tables is linking together through keys.	JSON, XML. Data is not structured.

SQL is the database language

Data is organized

Request made to DBMS for specific data

ID	Name	Course	Fee
1	John	DBMS	500
2	Michael	HTML	400
2	Matt	CSS	500

Give the details of students who are studying DBMS.

Meta Data – describe the actual data.

Valid DATA =

Keys:

1. **Primary key** – uniquely identify a row/ a record into the table – it can not null.

ID – Primary Key	Name	Course	Fee
1	John	DBMS	500
2	Michael	HTML	400
3	John	CSS	500
4	Zak	MySQL	600

2. Type of Primary key
    1. **Surrogate** – key that has no mapping to anything in the real world – random number assigned to an employee.
    2. **Natural** – key that has mapping to anything in the real world. The NI number act as primary key.
3. **Unique Key** – uniquely identify a row/ a record – but it can be null once.
4. **Foreign Key** – a primary key in another table is called foreign key.

Passport – Primary Key	Name
1	John
2	Kyle

3	
4	Zak

ID – Primary key	**Course**	**Std_Passport – Foreign Key**
01	JavaScript	1
02	DBMS	3
03	HTML	2
04	CSS	4

Name	**Course**
John	JavaScript

Composite Key: (Compound Key) – where primary key is made of two or more colums.

FName	**LName**	**Date_of_Birth**
Johan	Matt	1/1/1995
John	Orzel	10/1/1990
Zak	Matt	1/1/1995

None of the above column uniquely identify the row or record in the table.

We can make two column as primary key to uniquely identify the record in the above table.

SQL Basic – 4 Languages combine into one

1. **DQL** – Data Query Language – used to query the database for information. [select query].
2. **DDL** – Data Definition Language – used to define the database schema - what tables, columns, data types?
3. **DCL** – Data Control Language – used to control access/giving permissions to users.
4. **DML** – Data Manipulation Language – used to insert/update/ delete the data from the Database – **RUD**

5. **TCL –** Transaction Control Language – Rollback.

ID	**Name**
1	John
2	Zak

Show all the available databases?

Create two databases with Name DFE4, DFE5?

Delete the DFE5 database?

Select DFE4 Database?

List all the Available tables in Test database?

Show the definition of the Student table

Data types in MySQL

Int – whole number

Varchar (x) – for string text length of x;

Date – yyyy-mm-dd format

Timestamp - yyyy-mm-dd hh:mm:ss format

Decimal (m,n) – for decimal number – decimal (3,4) – 422.3330

1. m – is total number of digit before decimal point.
2. N – is total number of digit after decimal point

SQL Commands:

Clear the screen – Mac = Command + K – Windows= (System cls; / Ctrl+K)

SHOW DATABASES; display all the available database for us.

CREATE DATABASE <database_name>; create database.

USE <database_name> ; selecting a database

SHOW TABLES; list all tables within a database.

CREATE TABLE <tableName> (<columnName> <datatype>, <columnName> <datatype>,);

CREATE TABLE Student (ID int, Name varchar (10));

Describe <tableName>; - describe the structure of a table.

INSERT INTO <tableName> (column, column) Values (value1, ‘value2’);

SELECT * FROM <tableName>;

ID	firstName	lastName
int	Varchar (10)	Varchar (30)

Home Work

1. Create databases with the name of Company
2. Create below two tables.
3. Insert 10 records in each table
    1. Use single insertion and multi insertion

Database – Company

Student-

ID – int (5)	firstName – varchar (20)	lastName - varchar (50)	Gender

Course

ID – int (10)   courseName   startDate - timestamp

MySQL Constraint:

1. Not NULL - Make sure the column have a value. It cannot be null.
2. UNIQUE -
3. PRIMARY KEY
4. FOREIGN KEY
5. CHECK - The value in the columns satisfy a specific condition.
6. DEFAULT – set the default value for a column.
7. Auto_increment – not constraint. But extra feature. Allow unique number to be generated automatically when record gets inserted into the table.

Comparison Operators

1.	=	equal to
2.	>	greater than
3.	<	less than
4.	>=	greater than or equal
5.	<=	less than or equal
6.	<>	not equal to

Arithmetic Operators

1.	+	Add
2.	-	Subtract

3.	*	Multiply
4.	/	Divide
5.	%	Reminder

Logical Operators

1. AND – all the condition should be true

1. OR – either of the condition need to be true;

Updating records

1. Updating one column
    1. Update <tableName> Set <columnName> = <value> Where <condition>
2. Updating multiple columns
    1. UPDATE Employee SET firstName='Elyas', Salary=35000, Gender='Male' WHERE ID=3;

ALTER TABLE COMMAND: Used to add/delete or modify the columns in an existing table.

1. Add a column to an existing table
    1. Alter table <tableName> add <columnName> <datatype>;
2. Add a column at a specific position in the table
    1. Alter table <tableName> add <columnName> <datatype> after <existingColumnName>;
3. Drop a column from a table
    1. Alter table <tableName> drop column <columnName>;
4. Changing the data type of the Column
    1. Alter table <tableName> modify column <columnName> <newDataType>;
5. Renaming a table
    1. Alter table <tableName> rename to <newTableName>;
6. Renaming an existing column
    1. Alter table <tableName> rename column <columnName> to <newColumnName>;
7. Add primary key
    1. Alter table <tableName> add primary key (<columnName>);
8. Drop primary key
    1. Alter table <tableName> drop primary key;
9. Adding foreign Key
    1. Alter table <tableName> add foreign key (<columnName>) references <secondTableName> (<columnName-PrimaryKey);

Assignment

CREATE two tables named (Student and Courses) with below columns:

Student:

1. passportNumber: datatype = int
2. studentName: datatype= varchar (10)
3. studetnAge: datatype=int

Course:

1. courseID: datatype = int (10)
2. courseName: datatype = varchar (12)
3. sutdentPassport: datatype= int

Constraint for Student table:

1. the passportNumber should be primary key and automatically generating unique numbers.
2. The studentName can't be null.
3. The studentAge can't be null. It will be taking 18 as a default value.

Constraint for Course Table:

1. The courseID should be primary key and can not be null.
2. The courseName should only take (HTML, CSS, JavaScript, Database, Phyton) values.
3. The studentPassport should refer to the passportNumber of the Student table.

Note:

1. Insert at least 10 records in each table.
2. Update single column and multiple columns.
3. Rename the student table to Learners
4. Rename courseName column of Course table to Course.
5. Add another studentCountry column in Learners table through Alter Table command
6. Select details of students who are over 18.

3rd Day – 16/11/2022

1. Creating individual breakdown room and asking everyone to search about this for about 10 minutes.

Delete vs Truncate vs Drop

Delete: DML: Command
it used to deleted a record from a table. It only delete the rows/data from a table. The structure of the table remains unaffected. We can add where clause with this.

Truncate: DDL Command:
is used to delete all the data from a table. Here only the data is deleted. The structure of the table remains unaffected. There is no option to add the where clause with Truncate.

Drop: DDL Command:
It delete the whole table schema. (structure and data).
Everything is gone.
Dangerous command. Limited authorization.

We need to first delete the Foreign Key record then the Primary Key record.

BASIC QUERIES:

1. **Order by**: filters the results [ascending/ descending] based on the order by column. Order by Col1, Col2 -> it will order the result by col1 first, and if there are any records that have the same col1 value, it will then order by col2.
2. **Limit**: limit the number of records that you are getting from the database.
3. **IN**: if you want to include multiple records [with conditions] it is best to use in records.

1. To find all the employees in the employee table.
2. To find all the employees ordered by salary.
3. To find all the employee ordered by Sex and then by first Name
4. To find the first two employees.
5. To find the first Name and Last Name for all employee.
6. To find all the different Address from the Employee tables.

SQL Aggregate Functions:

Asking all to search about this for 5 minutes in the main room:

Count(), min(), max(), avg(), sum()

1. How many records are in Employee table.
2. To find the minimum salary an employee is receiving.
3. To find the total of all salaries in Employee table.
4. To find the total of salary for those Employees who's SEX is M.
5. To find minimum and maximum salary in Employee table.

Group by: where we can use the functions and can display the data in a more meaningful way.

1. To find out how many males and females are in Employee table.

--------- **Assignment for Thursday** ----------

--Display all the records from Employee Table

-- Display the first 5 record from Employee table.

-- Display the first 3 highest Salaries.

-- Display record of the 5 Employees who received the lowest salary.

-- Display number of all F and M employees from Employee table.

-- Display the AVG Salary for all employees separately based on their address.

-- Display the FNAME as First Name and LNAM as Last Name and other columns for all the employees from Employee table.

-- Display the full name for all employees from Employee table.

-- Display the different salaries of Employees from the Employee table.

Having Clause:

1. It is used in the select statement to specify filter conditions for a group of rows or aggregates.
2. Is often used with the GROUP BY clause to filter the groups based on a specific condition.
3. If you omit the group by clause, having clause behaves like a where clause.
4. The HAVING clause applies a filter to each group of rows, while the WHERE clause applies the filter to each individual row.

Syntax:

SELECT <columnList> FROM <tableName> WHERE <searchCondition> GROUP BY <groupByCondition> HAVING <groupCondition>

1. The having clause is only useful when we use it with the GROUP BY clause to generate the output for higher level reports. Finding the number of orders for this month/ quarterly/ half year.

WILDCARDS: A way of defining patterns that we want to match the specific data to.

LIKE: Special SQL Keyword used as wild card.

% (Percentage) = any number of character
_ (underscore) = one character.

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

Like Operator	Descriptions
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

2. SELECT all the Employees who's Name start with 'J'
   1. `SELECT * FROM Employee WHERE FNAME LIKE 'J%';`
3. SELECT Employees Name who has 'me' in the middle of thier name.
   1. `SELECT FNAME FROM Employee WHERE FNAME LIKE '%me%';`
4. Select employees who's first Name can be anything but should have 'oh' after firt character.
   1. `SELECT FNAME from employee where FNAME LIKE '_oh%';`

5. Display Name of employee which has 'Rames' and the last character can be anything.
    1. `SELECT FNAME FROM employee where FNAME LIKE 'Rames_';`
6. Select record of Employee who is born in 1965.
    1. `SELECT * FROM employee WHERE BDATE LIKE '1965-01%';`
7. Display country name which start with A and ends with N.
    1. `SELECT Name from Country where Name LIKE 'A%n';`
8. Display country name which start with A and ends with N.
    1. `SELECT Name from Country where Name LIKE 'A%n';`

UNION: is a special SQL operator which is used to combine the result of multiple select statement into one.

9. There could be an error if they don’t have similar data type.
10. When joining two tables they should have same number of columns;

The UNION will only select distinct values or records.
UNION ALL – Will select the duplicate value or record.

ID	Name - **varchar (10)**
1	Matt
2	Zak
2	Jeanette

ESS	**STD_NAME – varchar (10)**
4	Jeanette
5	Narayan

ID	**Name**
1	Matt
2	Zak
3	Jeanette
4	Narayan

On ess=id

UNION ALL

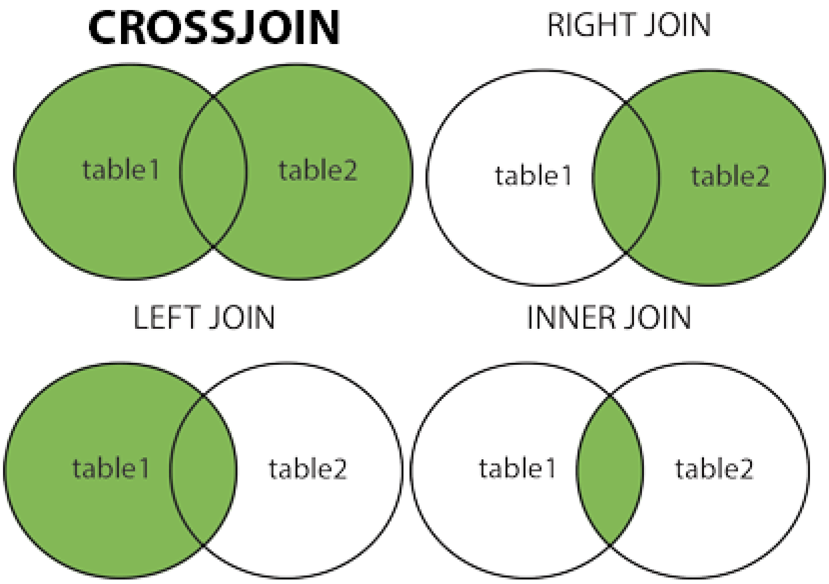
Name
Matt
Zak
Jeanette
Jeanette
Narayan

UNION

Name
Matt
Zak
Jeanette
Narayan

JOIN: Use to combine rows from two or more tables, based on a related columns between them.

1. **INNER JOIN** – Return records that have matching values in both table.
2. **LEF JOIN** – Returns all records from the left table, and the matched records from the right table.
3. **RIGHT JOIN** – Return all the records from the right table, and the matched records from the left table.
4. **CROSS JOIN** – Return all the records from both tables.



Self Join: join a table with itself.

1. The table must contain a column X [employeeID] that is a PK and different column Y [supervisorID] that stores values that can be matched up with the values in column Y.

Between: select records with in a given range: the value can be number, text and date;
The between command is inclusive (the begin and end values are included).

2. Display the record of all employees who has born between 1965-01-09 till 1995-01-01.

IN: With the in keyword we can specify multiple values in the where clause.

3. Display record of those employees who FNAME is Franklin, Ahmad or Alicia;

22/11/2022

String Function:

1. -- Concat(); --- used to concatnate two strings, columns.
2. select concat("Zak", " ","PARDIS");

3. USE Company;
4. select concat(FNAME, " ", LNAME) FROM Employee;
5. -- TRIM (); ---- used to remove extra spaces.
6. SELECT TRIM("   Zak   PARDIS   ") AS Name;
7. SELECT LTRIM("   Zak PARDIS   ") AS Name;
8. SELECT RTRIM("   Zak PARDIS   ") AS Name;
9. -- LOWER (); / LCASE (); -- Used to return the string in lower case;
10. SELECT LCASE("Zak PARDIS");
11. SELECT Lower("Zak PARDIS");
12. USE Company;
13. SELECT lower(concat(FNAME, " ",LNAME)) FROM employee;
14. -- Upper (); /Ucase (); -- used to return the string in Upper case;
15. SELECT upper("zak");
16. SELECT upper(concat(FNAME, " ",LNAME)) FROM employee;
17. SELECT upper(lower("Zak"));
18. -- now(); --- Return the current date and time
19. SELECT now();
20. -- curdate(); -- Return the current date
21. SELECT curdate();
22. -- CURRENT_TIME(); -- Return the current time
23. SELECT CURRENT_TIME();
24. SELECT MONTHNAME(CURDATE()); -- will return current month.
25. SELECT DAYNAME(CURDATE());
26. SELECT year(curdate());
27. SELECT day(curdate());
28.
29. -- reverse (); -- retrun the reverse of a string
30.
31. SELECT reverse("Google"); --
32.
33. -- substring(); -- extract a sub string from a string
34. SELECT substring("PARDIS",2);
35.
36. -- current_user() -- return the current user for use
37. SELECT current_user();

INSERT IGNORE: When you use the insert statement to add multiple row to a table, and if an error occurs during the processing, MySQL will terminate the whole statement and return an error. As a result now rows are inserted in the table.

38. If you use the INSERT IGNORE, the row with the invalid data are ignored and the rows with valid data gets inserted in the table.

ON DELETE CASCADE constraint is used in MySQL to delete the rows from the child table automatically, when the rows from the parent table are deleted.

ON DELETE SET NULL set the foreign key column or columns in the child table to NULL.

39. If you specify a SET NULL action, make sure that you have not declared the columns in the child table as NOT NULL

Store Procedure: pre-compiled MySQL statement stored into the database similar like table stored in the database.

Reduce Network Traffic

Making database secure;

Resource Usage;

Delimiter – the default delimiter in MySQL is semicolon.

Store Procedure Parameter

1. **IN** – Giving values to the Store Procedure – parameter passes a value into a procedure. The procedure might modify the value, but the modification is not visible to the caller when the procedure returns.

2. **OUT** – procedure is giving data/value to the caller – An OUT parameter passes a value from the procedure back to the caller. Its initial value is NULL within the procedure, and its value is visible to the caller when the procedure returns.

3. **INOUT** – it's doing the work of both IN and Out Parameter.

Variables in Store Procedure

1. Storing data in programming.

TRIGGERS:

A block of SQL code which will define a certain action that should happen when certain operation is performed on the table.

2. Row/recorded is added/deleted/updated in the table

Automate things

3. Event Logging

Command	Type	Keyword
Insert	Before | After	new
Update	Before | After	Old (before change) New (after change)
Delete	Before | After	Old

Narayn staff has been added

4. Before – the operation is executed before <insert the data into the table>
5. After – execution takes place. AFTER the data has been inserted/updated/deleted from a table.
6. Show triggers ();
7. Drop trigger <triggerName>;

Syntax:

```
Delimiter $$
Create trigger myTrigger Before | After  Insert | Delete | Update
On  tableName
For each row
Begin
        Code statement;

End$$
```

8. Write a trigger that will add a message “A new Employee is added” whenever we insert a row in the employee table.

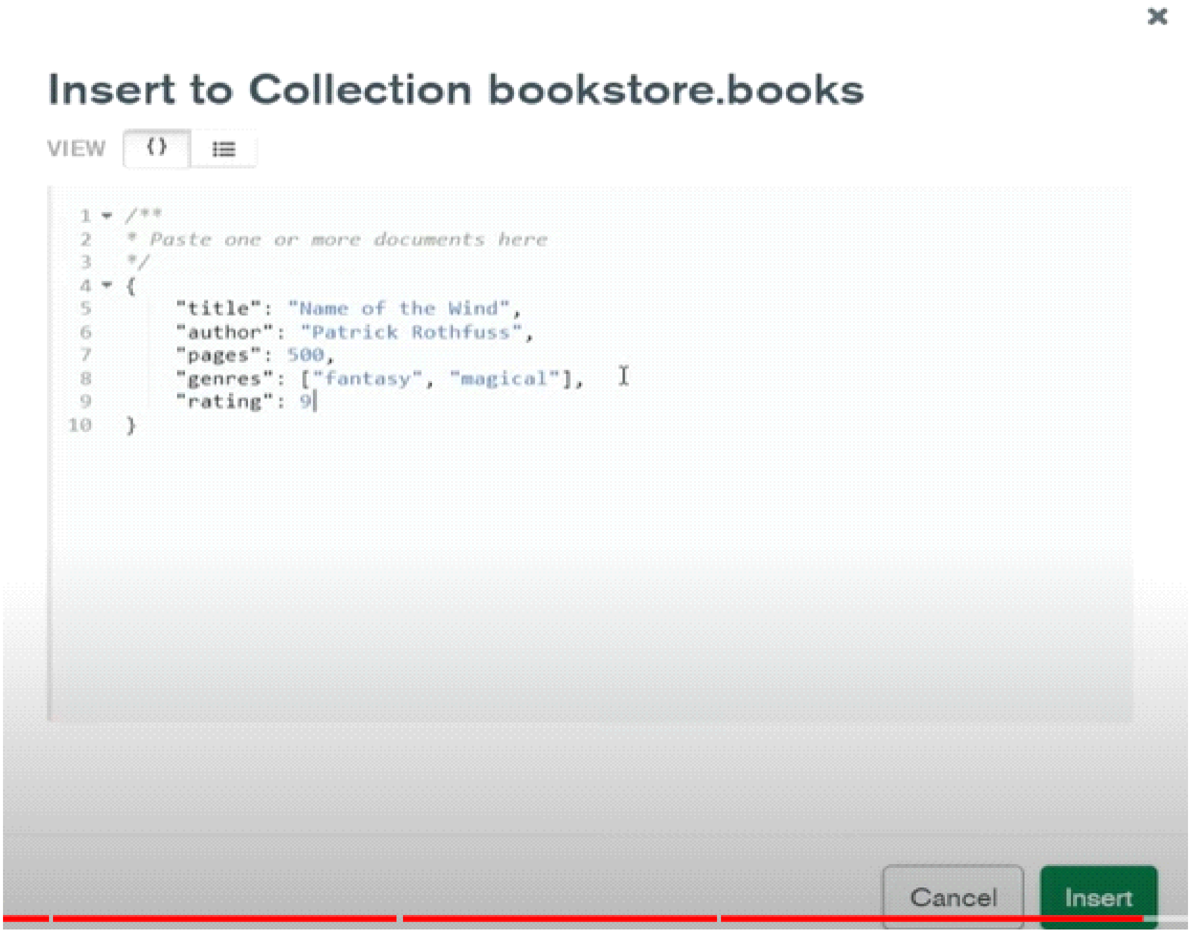
Mongo DB: NoSQL –

NoSQL: MongoDB:
It does not use SQL for interaction
Use JSON
There is no primary key or foreign key and no relation between them.
Data is not organized.
More flexible
faster
the data is store in the form of documents

SQL DB:
Use SQL for interaction with DB.
Store data in row/column (tabular)
Tables has relation with each other by PK, FK.
Data is organized.

Mango DB: it is a database which store data.

It is known as no SQL data base mean we don’t use SQL to interact with the database.



Insert to Collection bookstore.books

```
/**
 * Paste one or more documents here
 */
{
    "title": "Name of the Wind",
    "author": "Patrick Rothfuss",
    "pages": 500,
    "genres": ["fantasy", "magical"],
    "rating": 9
}
```

In the command prompt – type mongosh and then enter

Show dbs – shows all database.

Use <databaseName> -- to switch to database

In Mongo DBs it doesn't matter whether a database exist or not. Even if a database is not exist still we can type use and then the database name.

Show collections – show all the collection in the current database;

db – show the current database

db.dropDatabase() => dropping a database
After running the command, if we get {ok:"1"}, that means the command has run successfully, and that db has been dropped.

COLLECTION RELATED COMMANDS
db.createCollection("<collectionName>") => It will create a collection with the given collectionName.

show collections; => will bring up all the collections for that database.
db. <collectionName>.drop() –> for dropping a connection.

CRUD operations on MongoDB
Create – Inserting the records into the collections.
Read – Searching/querying the records from the collection.
Update – Updating the records in the collection.
Delete – Deletion of records from the collection.

In Mongodb, you don't have to follow a particular schema. The document(s) can have different key/value pairs, and they will get inserted into the collection without any problem.

To uniquely identify the row, MongoDB assigned an Object ID [like a PK]

Insertion:

db.book.insertOne ({title: "Mongo Database", author: "Zak PARDIS", pages: 50, rating: 5, genres: ["Software", "Database"]})

db.book.insertMany([{title: "JAVA Script", author: "Narayan", pages: 30, rating: 5, genres: ["Software Development", "Web Development"]},{ title: "CSS", author: "Tim", pages: 20, rating: 3, genres: ["Software Development", "Web Development", "Web Design"]}])

Searching Data:

db.books.find() – will show the first 20 documents in book collections

db.books.find({author: "Zak"}) – will find all data who's author is Zak

db.books.find({author: "Zak"}, rating:3) – it will only display the document where both match.

db.books.find({author: "Tim"}, {title: 1, author: 1}) -- this will only display the title and author of documents, who's author is Tim.

db.books.find({}, { title: 1, author: 1}) it will display the title and author for all the documents.

db.books.findOne({_id: need to past the id here})

Sorting data:

db.books.find().sort({title: 1}) – will do ascending sort by title for descending we need to type -1

db.books.find().count() -- count the data/documents

db.books.find({author: "Zak"}).count() -- will count all data/documents who's writer is Zak

db.books.find().limit(2) – will only display 3 data/documents

Delete Data

Db.books.deleteOne({_id: need to copy the id here});

Db.books.deleteMany({author: "Zak"}) -- will delete all the data who's author is Zak

Update Data

db.books.updateOne({_id: we need to copy and paste the id here}, {$set: {rating: 20, pages: 40}}) – it will update the rating to 8 and pages to 40 – ObjectId("622b746cd28eddc86c808d40")

db.books.updateMany({author: "Zak"}, {$set: {author: "Narayan"}}) -- it will update the author to Narayan who's author is Zak

--- optional ---

db.books.updateMany({author: "Zak"}, {$inc: {pages: 2}}) -- will increase the number of pages by 2.

lete many record/document