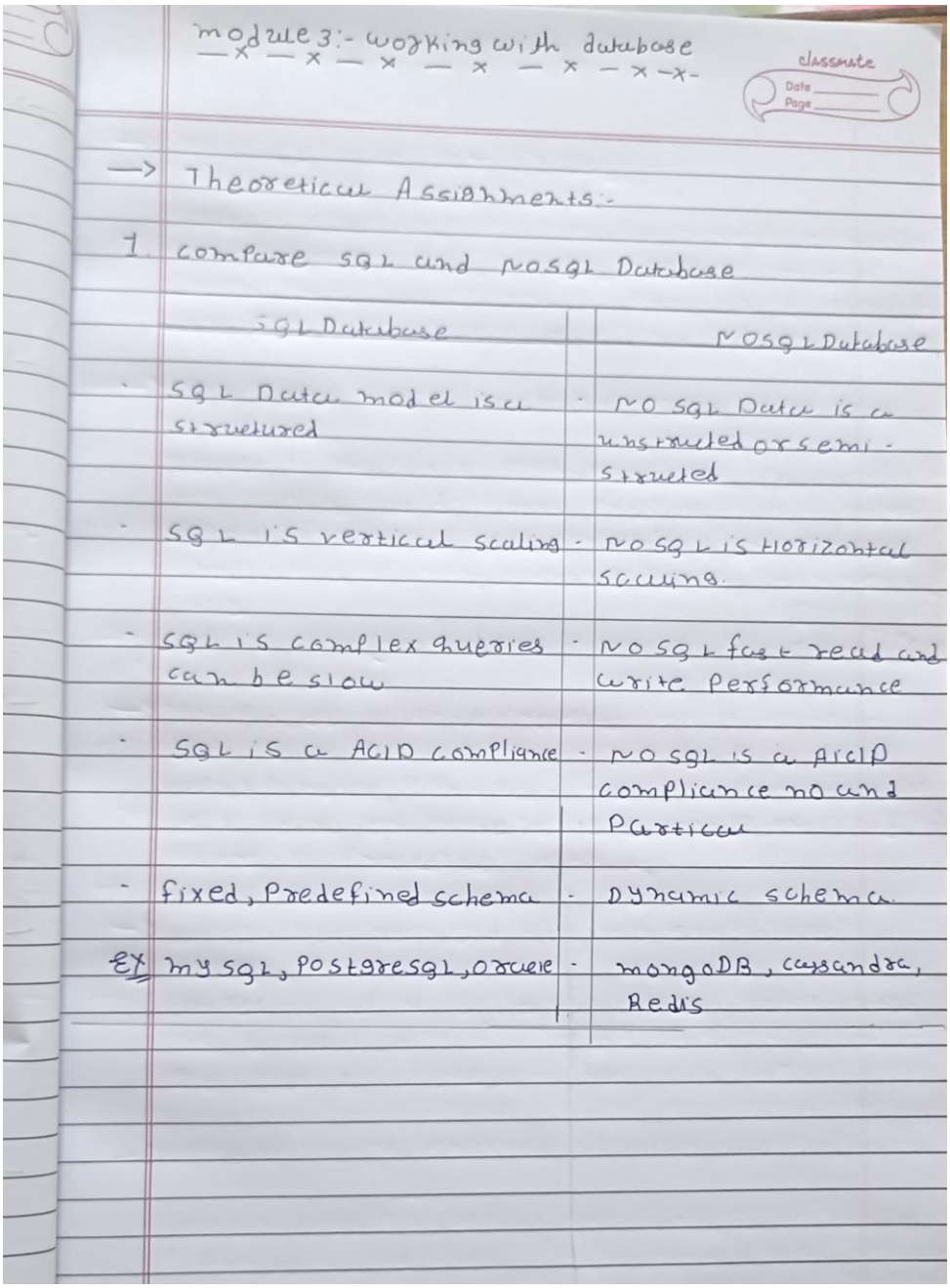
MODULE-3: Working with Databases

Theoretical Assignments: -

1. Compare SQL and NoSQL Databases:-



module 3:- working with database

→ Theoretical Assignments:-

1. compare SQL and NoSQL Database

SQL Database	NoSQL Database
SQL Data model is structured	NO SQL Data is unstructed or semi-structed
SQL is vertical scaling	NoSQL is Horizontal scaling.
SQL is complex queries can be slow	NO SQL fast read and write performance
SQL is a ACID compliance	NO SQL is a ACID compliance no and Partical
fixed, Predefined schema	Dynamic schema
Ex mysql, postgresql, oracle	mongoDB, cassandra, Redis

Practical Tasks for SQL:-

1. write a SQL query to find customers who are either from the city 'New York' or who do not have a grade greater than 100. Return customer_id, cust_name, city, grade, and salesman_id.

   Ans:-

   ```
   select customer_id,cust_name,city,grade,salesman_id
   from customer where city = 'newyork' or grade <= 100;
   ```

   Result Grid | Filter Rows: | Edit:

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3009	Geoff Cameron	Berlin	100	5003
NULL	NULL	NULL	NULL	NULL

2. write a SQL query to find all the customers in 'New York' city who have a grade value above 100. Return customer_id, cust_name, city, grade, and salesman_id.

   ANS:-

   ```
   select customer_id, cust_name, city, grade,salesman_id
    from customer
    where city = 'New York' and grade>100;
   ```

   Result Grid | Filter Rows: | Edit:

customer_id	cust_name	city	grade	salesman_id
3007	Brad Davis	New York	200	5001
NULL	NULL	NULL	NULL	NULL

3. Write a SQL query that displays order number, purchase amount, and the achieved and unachieved percentage (%) for those orders that exceed 50%of the target value of 6000.

ANS:-

```
select ord_no,purch_amt,round((purch_amt / 6000) * 100, 2) as
achieved_percentage,
round(100 - (purch_amt / 6000) * 100, 2) as unachieved_percentage
from orders where (purch_amt / 6000) * 100 > 50;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ord_no	purch_amt	achieved_percentage	unachieved_percentage
70008	5760.00	96.00	4.00
70013	3045.60	50.76	49.24

4. write a SQL query to calculate the total purchase amount of all orders. Return total purchase amount.

ANS:-

```
select sum(purch_amt) as total_purchase_amount from orders;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_purchase_amount
17541.18

5. write a SQL query to find the highest purchase amount ordered by each customer. Return customer ID, maximum purchase amount.

ANS:-

select customer_id,max(purch_amt) as maximum_purchase_amount from orders group by customer_id order by maximum_purchase_amount;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	maximum_purchase_amount
3003	75.29
3008	250.45
3001	270.65
3005	948.50
3004	1983.43
3007	2400.60
3009	2480.40
3002	5760.00

6. write a SQL query to calculate the average product price. Return average product price.

   ANS:-

   select round(avg(pro_price),2) as avgprice from item_mast;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avgprice
1435.00

7. write a SQL query to find those employees whose department is located at 'Toronto'. Return first name, last name, employee ID, job ID.

   ANS:-

   select e.first_name,e.last_name,e.employee_id,e.job_id
   from employees e

```
join departments d on e.department_id = d.department_id
join locations l on d.location_id = l.location_id
where l.city = 'Toronto';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	employee_id	job_id
Michael	Hartstein	201	MK_MAN
Pat	Fay	202	MK_REP

8. write a SQL query to find those employees whose salary is lower than that of employees whose job title is "MK_MAN". Exclude employees of the Job title ‘MK_MAN’. Return employee ID, first name, last name, job ID.

ANS:-

```
select employee_id, first_name, last_name, job_id
from employees
where salary < (select max(salary)
from employees where job_id = 'MK_MAN')and job_id <>
'MK_MAN';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employee_id	first_name	last_name	job_id
103	Alexander	Hunold	IT_PROG
104	Bruce	Ernst	IT_PROG
105	David	Austin	IT_PROG
106	Valli	Pataballa	IT_PROG
107	Diana	Lorentz	IT_PROG
108	Nancy	Greenberg	FI_MGR
109	Daniel	Faviet	FI_ACCOUNT
110	John	Chen	FI_ACCOUNT
111	Ismael	Sciarra	FI_ACCOUNT
112	Jose Manuel	Urman	FI_ACCOUNT
113	Luis	Popp	FI_ACCOUNT
114	Den	Raphaely	PU_MAN
115	Alexander	Khoo	PU_CLERK
116	Shelli	Baida	PU_CLERK
117	Sigal	Tobias	PU_CLERK
118	Guy	Himuro	PU_CLERK

employees 68

9. write a SQL query to find all those employees who work in department ID80or40. Return first name, last name, department number and department name.

   ANS:-

```
select e.first_name, e.last_name, e.department_id, d.department_name
from employees e
join departments d on e.department_id = d.department_id
where e.department_id in (80, 40);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	department_id	department_name
Susan	Mavris	40	Human Resources
John	Russell	80	Sales
Karen	Partners	80	Sales
Alberto	Errazuriz	80	Sales
Gerald	Cambrault	80	Sales
Eleni	Zlotkey	80	Sales
Peter	Tucker	80	Sales
David	Bernstein	80	Sales
Peter	Hall	80	Sales
Christopher	Olsen	80	Sales
Nanette	Cambrault	80	Sales
Oliver	Tuvault	80	Sales
Janette	King	80	Sales
Patrick	Sully	80	Sales
Allan	McEwen	80	Sales
Lindsey	Smith	80	Sales

Result 69

10. write a SQL query to calculate the average salary, the number of employees receiving commissions in that department. Return department name, average salary and number of employees.

ANS:-

```
select d.department_name , round(avg(e.salary),2)
avgsalary,count(e.commission_pct) numofemployee
from employees e
join departments d
on e.department_id = d.department_id
group by d.department_name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

department_name	avgsalary	numofemployee
Executive	19333.33	0
IT	5760.00	0
Finance	8600.00	0
Purchasing	4150.00	0
Shipping	3475.56	0
Sales	8955.88	34
Administration	4400.00	0
Marketing	9500.00	0
Human Resources	6500.00	0
Public Relations	10000.00	0
Accounting	10150.00	0

11. write a SQL query to find out which employees have the same designation as the employee whose ID is 169. Return first name, last name, department ID and job ID.

ANS:-

```
select first_name, last_name, department_id, job_id
from employees
where job_id = (select job_id from employees where employee_id = 169);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	department_id	job_id
Peter	Tucker	80	SA_REP
David	Bernstein	80	SA_REP
Peter	Hall	80	SA_REP
Christopher	Olsen	80	SA_REP
Nanette	Cambrault	80	SA_REP
Oliver	Tuvault	80	SA_REP
Janette	King	80	SA_REP
Patrick	Sully	80	SA_REP
Allan	McEwen	80	SA_REP
Lindsey	Smith	80	SA_REP
Louise	Doran	80	SA_REP
Sarath	Sewall	80	SA_REP
Clara	Vishney	80	SA_REP
Danielle	Greene	80	SA_REP
Mattea	Marvins	80	SA_REP
David	Lee	80	SA_REP
Sundar	Ande	80	SA_REP

employees 71

12. write a SQL query to find those employees who earn more than the average salary. Return employee ID, first name, last name.

ANS:-

```
select employee_id, first_name, last_name
from employees
where salary > (select avg(salary) from employees);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employee_id	first_name	last_name
100	Steven	King
101	Neena	Kochhar
102	Lex	De Haan
103	Alexander	Hunold
108	Nancy	Greenberg
109	Daniel	Faviet
110	John	Chen
111	Ismael	Sciarra
112	Jose Manuel	Urman
113	Luis	Popp
114	Den	Raphaely
120	Matthew	Weiss
121	Adam	Fripp
122	Payam	Kaufling
123	Shanta	Vollman
145	John	Russell

employees 72

13. write a SQL query to find all those employees who work in the Finance department. Return department ID, name (first), job ID and department name.

ANS:-

```
select e.department_id,e.first_name,e.job_id,d.department_name
from employees e
join departments d ON e.department_id = d.department_id
where d.department_name = 'Finance';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

department_id	first_name	job_id	department_name
100	Nancy	FI_MGR	Finance
100	Daniel	FI_ACCOUNT	Finance
100	John	FI_ACCOUNT	Finance
100	Ismael	FI_ACCOUNT	Finance
100	Jose Manuel	FI_ACCOUNT	Finance
100	Luis	FI_ACCOUNT	Finance

14. From the following table, write a SQL query to find the employees who earn less than the employee of ID 182. Return first name, last name and salary.

ANS:-

```
select first_name, last_name, salary
from employees
where salary < (select salary from employees where employee_id = 182);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	salary
James	Landry	2400.00
Steven	Markle	2200.00
TJ	Olson	2100.00
Ki	Gee	2400.00
Hazel	Philtanker	2200.00

15. Create a stored procedure Count Employees By Dept that returns the number of employees in each department

Ans:-

```
delimiter //
create procedure countemployee()
begin
  select department_id,count(*) as totalemployee from employees
  group by department_id;
end//
```

```
delimiter ;

call countemployee();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

department_id	totalemployee
NULL	1
10	1
20	2
30	6
40	1
50	45
60	5
70	2
80	34

Result 10

16. Create a stored procedure Add New Employee that adds a new employee to the database.

ANS:-

```
delimiter //
create procedure addnewemployee(
  in employee_id INT UNSIGNED ,
   in first_name VARCHAR(20),
    in last_name VARCHAR(25) ,
   in email VARCHAR(25) ,
   in phone_number VARCHAR(20),
  in  hire_date DATE ,
  in  job_id VARCHAR(10) ,
  in  salary DECIMAL(8, 2),
  in  commission_pct DECIMAL(5, 2),
  in  manager_id INT UNSIGNED,
  in  department_id INT UNSIGNED
```

```
)
begin
  INSERT INTO employees
VALUES
(employee_id, first_name, last_name, email, phone_number, hire_date,
job_id, salary, commission_pct, manager_id, department_id);
end //
delimiter ;
SET FOREIGN_KEY_CHECKS = 0;
call addNEWemployee(209, 'gargi', 'Baer', 'HBAER', '515.123.8888', '1994-06-
08', 'PR_REP', 10000, NULL, 101, 70);
SET FOREIGN_KEY_CHECKS = 1;
select*from employees;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	2003-06-17	AD_PRES	24000.00	NULL	NULL	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21	AD_VP	17000.00	NULL	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13	AD_VP	17000.00	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03	IT_PROG	9000.00	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21	IT_PROG	6000.00	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	2005-06-25	IT_PROG	4800.00	NULL	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05	IT_PROG	4800.00	NULL	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07	IT_PROG	4200.00	NULL	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	2002-08-17	FI_MGR	12000.00	NULL	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	2002-08-16	FI_ACCOUNT	9000.00	NULL	108	100
110	John	Chen	JCHEN	515.124.4269	2005-09-28	FI_ACCOUNT	8200.00	NULL	108	100
111	Ismael	Sciarra	ISCIARRA	515.124.4369	2005-09-30	FI_ACCOUNT	7700.00	NULL	108	100
112	Jose Manuel	Urman	JMURMAN	515.124.4469	2006-03-07	FI_ACCOUNT	7800.00	NULL	108	100
113	Luis	Popp	LPOPP	515.124.4567	2007-12-07	FI_ACCOUNT	6900.00	NULL	108	100
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

17. Create a stored procedure Delete Employees By Dept that removes all employees from a specific department

ANS:-

```
delimiter //
create procedure DeleteEmployeesByDept(in dept_id int unsigned)
begin
```

```
    delete
    from employees
    where  department_id = dept_id;
end //
delimiter ;
SET FOREIGN_KEY_CHECKS = 0;
call DeleteEmployeesByDept(100);
SET FOREIGN_KEY_CHECKS = 1;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	2003-06-17	AD_PRES	24000.00	NULL	NULL	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21	AD_VP	17000.00	NULL	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13	AD_VP	17000.00	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03	IT_PROG	9000.00	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21	IT_PROG	6000.00	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	2005-06-25	IT_PROG	4800.00	NULL	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05	IT_PROG	4800.00	NULL	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07	IT_PROG	4200.00	NULL	103	60
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

18. Create a stored procedure Get Top Paid Employees that retrieves the highest-paid employee in each department.

ANS:-

```
DELIMITER //
CREATE PROCEDURE GetTopPaidEmployees()
BEGIN
  SELECT e.employee_id, e.first_name, e.last_name,
e.department_id, e.salary
   FROM employees e
   WHERE e.salary = (
      SELECT MAX(salary)
      FROM employees
      WHERE department_id = e.department_id
      );
END //
```

DELIMITER ;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employee_id	first_name	last_name	department_id	salary
100	Steven	King	90	24000.00
103	Alexander	Hunold	60	9000.00

19. Create a stored procedure Promote Employee that increases an employee's salary and changes their job role.

ANS:-

```
delimiter //
create procedure PromoteEmployee(IN emp_id INT,
   IN new_salary DECIMAL(8,2),
   IN new_job_id VARCHAR(10)
)
begin
update employees
set salary = new_salary,
      job_id = new_job_id
   WHERE employee_id = emp_id;
end //
delimiter ;
call PromoteEmployee(108, 6500.00, 'AC_ACCOUNT');
select*from employees;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	2003-06-17	AD_PRES	24000.00	NULL	NULL	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21	AD_VP	17000.00	NULL	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13	AD_VP	17000.00	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03	IT_PROG	9000.00	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21	IT_PROG	6000.00	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	2005-06-25	IT_PROG	4800.00	NULL	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05	IT_PROG	4800.00	NULL	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07	IT_PROG	4200.00	NULL	103	60
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

20. Create a stored procedure Assign Manager To Department that assigns a new manager to all employees in a specific department.

ANS:-

```
delimiter //
create procedure AssignManagerToDepartment (in dept_id INT UNSIGNED,
in new_manager_id INT UNSIGNED)
begin
update employees
set manager_id = new_manager_id
where department_id = dept_id;
end //
delimiter ;
call AssignManagerToDepartment(100,112);
select*from employees;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	2003-06-17	AD_PRES	24000.00	NULL	NULL	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21	AD_VP	17000.00	NULL	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13	AD_VP	17000.00	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03	IT_PROG	9000.00	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21	IT_PROG	6000.00	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	2005-06-25	IT_PROG	4800.00	NULL	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05	IT_PROG	4800.00	NULL	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07	IT_PROG	4200.00	NULL	103	60
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

NO SQL:-

1. Retrieve all employee records:-

Ans:-

db.employees.find().pretty()

```
{
  _id: ObjectId('685e1edf867d2c85e8748a5f'),
  name: 'Alice Johnson',
  age: 30,
  department: 'HR',
  salary: 60000,
  joining_date: ISODate('2019-05-15T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a60'),
  name: 'Bob Smith',
  age: 40,
  department: 'IT',
  salary: 80000,
  joining_date: ISODate('2015-08-20T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a61'),
  name: 'Charlie Brown',
  age: 35,
  department: 'Finance',
  salary: 75000,
  joining_date: ISODate('2018-11-30T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a62'),
  name: 'David White',
  age: 28,
  department: 'IT',
  salary: 72000,
  joining_date: ISODate('2021-01-10T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a63'),
  name: 'Emma Wilson',
  age: 32,
  department: 'Marketing',
  salary: 65000,
  joining_date: ISODate('2017-03-25T00:00:00.000Z')
},
```

```
},
{
  _id: ObjectId('685e1edf867d2c85e8748a64'),
  name: 'Franklin Adams',
  age: 45,
  department: 'Finance',
  salary: 90000,
  joining_date: ISODate('2010-07-12T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a65'),
  name: 'Grace Lee',
  age: 29,
  department: 'HR',
  salary: 58000,
  joining_date: ISODate('2020-06-05T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a66'),
  name: 'Henry Ford',
  age: 50,
  department: 'IT',
  salary: 95000,
  joining_date: ISODate('2008-12-15T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a67'),
  name: 'Isabella Martinez',
  age: 38,
  department: 'Marketing',
  salary: 70000,
  joining_date: ISODate('2016-09-18T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a68'),
  name: 'Jack Carter',
  age: 27,
  department: 'Finance',
  salary: 68000,
  joining_date: ISODate('2022-04-10T00:00:00.000Z')
}
```

2. Find employees who work in the IT department

Ans:-

db.employees.find({ department: "IT" })

```
mydata> db.employees.find({ department: "IT" })
[
  {
    _id: ObjectId('685e1edf867d2c85e8748a60'),
    name: 'Bob Smith',
    age: 40,
    department: 'IT',
    salary: 80000,
    joining_date: ISODate('2015-08-20T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a62'),
    name: 'David White',
    age: 28,
    department: 'IT',
    salary: 72000,
    joining_date: ISODate('2021-01-10T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a66'),
    name: 'Henry Ford',
    age: 50,
    department: 'IT',
    salary: 95000,
    joining_date: ISODate('2008-12-15T00:00:00.000Z')
  }
]
```

3. Find employees who have a salary greater than 70,000

ANS:-

db.employees.find({ salary: { $gt: 70000 } })

```
]
mydata> db.employees.find({ salary: { $gt: 70000 } })
[
  {
    _id: ObjectId('685e1edf867d2c85e8748a60'),
    name: 'Bob Smith',
    age: 40,
    department: 'IT',
    salary: 80000,
    joining_date: ISODate('2015-08-20T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a61'),
    name: 'Charlie Brown',
    age: 35,
    department: 'Finance',
    salary: 75000,
    joining_date: ISODate('2018-11-30T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a62'),
    name: 'David White',
    age: 28,
    department: 'IT',
    salary: 72000,
    joining_date: ISODate('2021-01-10T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a64'),
    name: 'Franklin Adams',
    age: 45,
    department: 'Finance',
    salary: 90000,
    joining_date: ISODate('2010-07-12T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a66'),
    name: 'Henry Ford',
    age: 50,
    department: 'IT',
    salary: 95000,
    joining_date: ISODate('2008-12-15T00:00:00.000Z')
  }
```

4. Find employees who joined after 2018
   ANS:-

```
db.employees.find({ joining_date: { $gt: ISODate("2018-12-31") }});
```

```
{
  _id: ObjectId('685e1edf867d2c85e8748a5f'),
  name: 'Alice Johnson',
  age: 30,
  department: 'HR',
  salary: 60000,
  joining_date: ISODate('2019-05-15T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a62'),
  name: 'David White',
  age: 28,
  department: 'IT',
  salary: 72000,
  joining_date: ISODate('2021-01-10T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a65'),
  name: 'Grace Lee',
  age: 29,
  department: 'HR',
  salary: 58000,
  joining_date: ISODate('2020-06-05T00:00:00.000Z')
},
{
  _id: ObjectId('685e1edf867d2c85e8748a68'),
  name: 'Jack Carter',
  age: 27,
  department: 'Finance',
  salary: 68000,
  joining_date: ISODate('2022-04-10T00:00:00.000Z')
}
```

5. Find employees between the ages of 30 and 40

ANS:-

```
db.employees.find({ age: { $gte: 30, $lte: 40 } })
```

```
mydata> db.employees.find({ age: { $gte: 30, $lte: 40 } })
[
  {
    _id: ObjectId('685e1edf867d2c85e8748a5f'),
    name: 'Alice Johnson',
    age: 30,
    department: 'HR',
    salary: 60000,
    joining_date: ISODate('2019-05-15T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a60'),
    name: 'Bob Smith',
    age: 40,
    department: 'IT',
    salary: 80000,
    joining_date: ISODate('2015-08-20T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a61'),
    name: 'Charlie Brown',
    age: 35,
    department: 'Finance',
    salary: 75000,
    joining_date: ISODate('2018-11-30T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a63'),
    name: 'Emma Wilson',
    age: 32,
    department: 'Marketing',
    salary: 65000,
    joining_date: ISODate('2017-03-25T00:00:00.000Z')
  },
  {
    _id: ObjectId('685e1edf867d2c85e8748a67'),
    name: 'Isabella Martinez',
    age: 38,
    department: 'Marketing',
    salary: 70000,
    joining_date: ISODate('2016-09-18T00:00:00.000Z')
  }
```

6. Increase the salary of all employees in the Finance department by 5%

   ANS:-

db.your_collection_name.updateMany({ department: "Finance" }, { $mul: { salary: 1.05 })

```
mydata> db.your_collection_name.updateMany(
...     { department: "Finance" },
...     { $mul: { salary: 1.05 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
```

7. Delete employees who joined before 2010

   ANS:-

   db.employees.deleteMany({joining_date: { $lt: ISODate("2010-01-01") }});

```
mydata> db.employees.deleteMany({joining_date: { $lt: ISODate("2010-01-01") }});
{ acknowledged: true, deletedCount: 1 }
mydata>
```

8. Find the highest-paid employee

   ANS:-

   db.employees.find().sort({ salary: -1 }).limit(1)

```
mydata> db.employees.find().sort({ salary: -1 }).limit(1)
[
  {
    _id: ObjectId('685e1edf867d2c85e8748a64'),
    name: 'Franklin Adams',
    age: 45,
    department: 'Finance',
    salary: 90000,
    joining_date: ISODate('2010-07-12T00:00:00.000Z')
  }
]
mydata>
```