<table>
<tr><th>Sr.</th><th>Practical</th></tr>
<tr><td>Lab-1
[Trigger]</td><td>
Create Database with Name: Person_Info
Create following tables under Person_Info database.

<table>
<tr><th colspan="3">Person</th></tr>
<tr><th>Column_Name</th><th>DataType</th><th>Constraints</th></tr>
<tr><td>PersonID</td><td>Int</td><td>Primary Key</td></tr>
<tr><td>PersonName</td><td>Varchar (50)</td><td>Not Null</td></tr>
<tr><td>Salary</td><td>Decimal (8,2)</td><td>Not Null</td></tr>
<tr><td>JoiningDate</td><td>Datetime</td><td>Not Null</td></tr>
<tr><td>City</td><td>Varchar (100)</td><td>Not Null</td></tr>
<tr><td>Age</td><td>Int</td><td>Null</td></tr>
<tr><td>BirthDate</td><td>Datetime</td><td>Not Null</td></tr>
</table>

<table>
<tr><th colspan="3">PersonLog</th></tr>
<tr><th>Column_Name</th><th>DataType</th><th>Constraints</th></tr>
<tr><td>PLogID</td><td>Int</td><td>Primary Key, Auto Increment (Identity)</td></tr>
<tr><td>PersonID</td><td>Int</td><td>Not Null</td></tr>
<tr><td>PersonName</td><td>Varchar(250)</td><td>Not Null</td></tr>
<tr><td>Operation</td><td>Varchar (50)</td><td>Not Null</td></tr>
<tr><td>UpdatedDate</td><td>Datetime</td><td>Not Null</td></tr>
</table>

From the above given table perform the following queries:

Create INSERT, UPDATE & DELETE Stored Procedures for Person table.
Create a trigger that fires on INSERT, UPDATE and DELETE operation on the Person table. For that, create a new table PersonLog to log (enter) all operations performed on the Person table.
Create an INSTEAD OF trigger that fires on INSERT, UPDATE and DELETE operation on the Person table. For that, log all operation performed on the Person table into PersonLog.
Create DELETE trigger on PersonLog table, when we delete any record of PersonLog table it prints ‘Record deleted successfully from PersonLog’.
Create INSERT trigger on person table, which calculates the age and update that age in Person table.

</td></tr>
<tr><td>Lab-2
[Cursor
&
Exception
Handling]</td><td>
Create Database with Name: Person_Info
Create following tables under PersonLog database.

<table>
<tr><th colspan="3">PersonLog</th></tr>
<tr><th>Column_Name</th><th>DataType</th><th>Constraints</th></tr>
<tr><td>PLogID</td><td>Int</td><td>Primary Key</td></tr>
<tr><td>PersonName</td><td>Varchar(250)</td><td>Not Null</td></tr>
<tr><td>Operation</td><td>Varchar (50)</td><td>Not Null</td></tr>
<tr><td>UpdatedDate</td><td>Datetime</td><td>Not Null</td></tr>
</table>

From the above given table perform the following queries:

Print message like - Error Occur that is: Divide by zero error encountered.
Print error message in insert statement using Error_Message () function: Conversion failed when converting datetime from character string.

</td></tr>
</table>

	3. Create procedure which prints the error message that "The PLogID is already taken. Try another one". 4. Create procedure that print the sum of two numbers: take both number as integer & handle exception with all error functions if any one enters string value in numbers otherwise print result. 5. Throw custom exception using stored procedure which accepts PLogID as input & that throws Error like no plogid is available in database. 6. Create cursor with name per_cursor which takes PLogID & PersonName as variable and produce combine output with PLogID & Person Name. 7. Use Table Student (Id, Rno, EnrollmentNo, Name, Branch, University) - Create cursor that updates enrollment column as combination of branch & Roll No. like SOE22CE0001 and so on. (22 is admission year)

Lab-3 [MongoDB]

Create Database with Name: **BANK_INFO**
Insert below data into the Collection.

Deposite

ACTNO	CNAME	BNAME	AMOUNT	ADATE
101	ANIL	VRCE	1000.00	1-3-95
102	SUNIL	AJNI	5000.00	4-1-96
103	MEHUL	KAROLBAGH	3500.00	17-11-95
104	MADHURI	CHANDI	1200.00	17-12-95
105	PRMOD	M.G. ROAD	3000.00	27-3-96
106	SANDIP	ANDHERI	2000.00	31-3-96
107	SHIVANI	VIRAR	1000.00	5-9-95
108	KRANTI	NEHRU PLACE	5000.00	2-7-95

Branch

BNAME	CITY
VRCE	NAGPUR
AJNI	NAGPUR
KAROLBAGH	DELHI
DHARAMPETH	NAGPUR
VIRAR	BOMBAY
NEHRU PLACE	DELHI
POWAI	BOMBAY

Customer

CNAME	CITY
ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
SHIVANI	BOMBAY

Borrow

LOANNO	CNAME	LBNAME	AMOUNT
201	ANIL	VRCE	1000.00
206	MEHUL	AJNI	5000.00
311	SUNIL	DHARAMPETH	3000.00
321	MADHURI	ANDHERI	2000.00
375	PRMOD	VIRAR	8000.00
205	ANIL	NEHRU PLACE	3000.00

From the above given collection perform the following queries:

1. Retrieve/Display every document of your collection.
2. Retrieve/Display every document of your collection. (Use option pretty)
3. Display only one documents of your collection. (Use findone)
4. Display documents whose Account Number is 101.
5. Display documents whose Account Number is less than 103.
6. Display documents whose Account Number is greater than 102 and Customer Name is Arjun.
7. Display documents whose Account Number is 105 or 108 using IN.
8. Display documents whose Account Number is not greater than 105.
9. Display documents with CNAME, CCITY, BNAME and AMOUNT fields.
10. Display Nagpur city branch's documents with CNAME, CCITY, BNAME and AMOUNT fields.
11. Display every document of your collection on ascending order by CNAME and descending order by AMOUNT.
12. Display only two documents of your collection. (Use LIMIT | Use Customer Collection)
13. Display from 3rd documents of your collection. (Use SKIP | Use Borrow Collection)
14. Display the count of documents in your collection. (Use Deposite Collection)
15. Display the documents whose name starts with S in your collection.
16. Display the documents whose name starts with S or M in your collection.
17. Display the documents whose name starts with A and having 5 characters in your collection.
18. Display the documents whose name starts with A to M in your collection.
19. Display the sum of amount in your collection. (Use Deposite Collection)
20. Display branch wise sum of amount in your collection. (Use Deposite Collection)
21. Update name of Anil to Arjun and also Branch Name to "DPS".
22. Delete the document whose Branch Name is DPS.
23. Drop BANK_INFO database.

Lab-4 [MongoDB]

Create Database with Name: **Employee_Info**
Create following Collection under Employee_Info database.

Employee

EID	EName	Gender	JoiningDate	Salary	City
1	Nick	Male	01-JAN-13	4000	London
2	Julian	Female	01-OCT-14	3000	New York
3	Roy	Male	01-JUN-16	3500	London
4	Tom	Male	*NULL*	4500	London
5	Jerry	Male	01-FEB-13	2800	Sydney
6	Philip	Male	01-JAN-15	7000	New York
7	Sara	Female	01-AUG-17	4800	Sydney
8	Emily	Female	01-JAN-15	5500	New York

9	Michael	Male	*NULL*	6500	London
10	John	Male	01-JAN-15	8800	London

From the above given table perform the following queries:

1. Write a MongoDB query to display all the documents in the collection Employee.
2. Write a MongoDB query to display the fields EID, Name, Gender, and salary for all the documents in the collection employee.
3. Write a MongoDB query to display the fields EID, Name, Gender, and City, but exclude the field _id for all the documents in the collection employee.
4. Write a MongoDB query to display the fields salary, but exclude the field _id for all the documents in the collection employee.
5. Write a MongoDB query to display all the Employees which are in the city London.
6. Write a MongoDB query to display the first 5 EID which are in the city Sydney.
7. Write a MongoDB query to display the next 2 Employees after skipping the first 2 which are in the city New York.
8. Write a MongoDB query to display the count of documents in your collection.
9. Write a MongoDB query to display the sum of salary in your collection.
10. Write a MongoDB query to display the documents whose employee name starts with S or M in your collection.
11. Write a MongoDB query to find the employee Id, name, city, and salary for those employees which contain 'Phi' as the first three letters of their name.
12. Write a MongoDB query to find the employee Id, name, city, and gender for those employees which contain 'ael' as the last three letters of their name.
13. Write a MongoDB query to find the name, joining date, and city for those restaurants which contain 'dne' as three letters somewhere in their city name.
14. Write a MongoDB query to find the employee Id, name, city, and joining date for those employees which do not belong to the city London or Sydney.
15. Write a MongoDB query to find the name and city for those employees which salary is more than 10000.
16. Write a MongoDB query to arrange the name of the employees in ascending order along with all the columns.
17. Write a MongoDB query to arrange the city of the employees in descending order along with all the columns.
18. Write a MongoDB query to arrange the name of the employees in ascending order and, the city should be in descending order.
19. Write a MongoDB query to display city wise sum of salary from employee collection.
20. Write a MongoDB query to delete the document whose city name is London.