CYCLE 1 : SQL QUERY

1] Write a program to implement database triggers in PL/SQL by using following schema -employee(e_id,e_name,e_doj,e_salary,e_age)

i. Create an employee table and insert any five records.

ii. Write row-level trigger for salary changes.(insert / update / delete operations on employee)

→

```
SQL>  CREATE TABLE EMPLOYEE(
  2   E_ID NUMBER,
  3   E_NAME VARCHAR(20),
  4   E_DOJ VARCHAR(20),
  5   E_SALARY NUMBER,
  6   E_AGE NUMBER,
  7   PRIMARY KEY (E_ID));
Table created.

SQL> insert into EMPLOYEE values (1,'tanu','2022-08-10',30000,22);
1 row created.
SQL> insert into EMPLOYEE values(2,'avi','2023-10-21',45000,34);
1 row created.
SQL> insert into EMPLOYEE values (3,'anu','2005-04-25',50000,39);
1 row created.
SQL> insert into EMPLOYEE values (4,'ash','2010-06-3',25000,42);
1 row created.
SQL> insert into EMPLOYEE values (5,'shyam','2021-11-2',46000,31);
1 row created.

SQL> CREATE OR REPLACE TRIGGER display_salary_changes
  2  BEFORE DELETE OR INSERT OR UPDATE ON employee
  3  FOR EACH ROW
  4  WHEN (NEW.E_ID > 0)
  5  DECLARE
```

```
 6  sal_diff number;
 7  BEGIN
 8  sal_diff := :NEW.E_SALARY - :OLD.E_SALARY;
 9  dbms_output.put_line('Old salary: ' || :OLD.E_SALARY);
10  dbms_output.put_line('New salary: ' || :NEW.E_SALARY);
11  dbms_output.put_line('Salary difference: ' || sal_diff);
12  END;
13  /
```

Trigger created.

```
SQL> SET SERVEROUTPUT ON;

SQL> insert into EMPLOYEE values (6,'siri','2023-11-21',50000,25);
```

Old salary:

New salary: 50000

Salary difference:

1 row created.

```
SQL> UPDATE EMPLOYEE
  2  SET E_SALARY=E_SALARY+500
  3  WHERE E_ID=2;
```

Old salary: 45000

New salary: 45500

Salary difference: 500

1 row updated.

```
SQL> select * from EMPLOYEE;
```

E_ID	E_NAME	E_DOJ	E_SALARY	E_AGE
1	tanu	2022-08-10	30000	22
2	avi	2023-10-21	45500	34
3	anu	2005-04-25	50000	39
4	ash	2010-06-3	25000	42
5	shyam	2021 -11-2	46000	31

```
     6 siri                 2023-11-21          50000               25

  6 rows selected.

SQL> DELETE EMPLOYEE
  2  WHERE E_ID=3;
1 row deleted.

SQL> select * from EMPLOYEE;
```

E_ID	E_NAME	E_DOJ	E_SALARY	E_AGE
1	tanu	2022-08-10	30000	22
2	avi	2023-10-21	45500	34
4	ash	2010-06-3	25000	42
5	shyam	2021-11-2	46000	31
6	siri	2023-11-21	50000	25

2]. Write a program to implement database triggers in PL/SQL by using following schema – employee2(e_id,e_name,e_age)

i. Create an employee table and insert any five records.

ii. Write a trigger to check the age of an employee is between 18 to 58, if not raise an error.(during insert / update / delete operations on employee)

→

```
SQL> CREATE TABLE EMPLOYEE2(
 2  E_ID NUMBER,
 3  E_NAME VARCHAR(20),
 4  E_AGE NUMBER,
 5  PRIMARY KEY (E_ID));
Table created.

SQL> insert into EMPLOYEE2 values (1,'tanu',22);
1 row created.
SQL> insert into EMPLOYEE2 values (2,'sam',32);
1 row created.
SQL> insert into EMPLOYEE2 values (3,'shree',42);
1 row created.
SQL> insert into EMPLOYEE2 values (4,'ram',33);
1 row created.
SQL> insert into EMPLOYEE2 values (5,'dhru',45);
1 row created.

SQL> CREATE OR REPLACE TRIGGER DISPLAY_AGE_CHANGES
 2  BEFORE INSERT OR UPDATE OR DELETE ON EMPLOYEE2
 3  FOR EACH ROW
 4  WHEN(NEW.E_ID>0)
 5  BEGIN
 6  IF:NEW.E_AGE < 18
 7  THEN
 8  RAISE_APPLICATION_ERROR(-20001,'Employee age must be greater than or
```

```
 9  equal to 18.');
10  ELSIF:NEW.E_AGE > 58
 11  THEN
 12  RAISE_APPLICATION_ERROR(-20001,'Employee age must be lesser than or equal
 13  to 58.');
 14  END IF;
 15  END;
 16  /
Trigger created.

SQL> SET SERVEROUTPUT ON;

SQL> insert into EMPLOYEE2 values (6,'raksh',16);
insert into EMPLOYEE2 values (6,'raksh',16)
       *
ERROR at line 1:
ORA-20001: Employee age must be greater than or
equal to 18.
ORA-06512: at "SCOTT.DISPLAY_AGE_CHANGES", line 4
ORA-04088: error during execution of trigger 'SCOTT.DISPLAY_AGE_CHANGES'

SQL> insert into EMPLOYEE2 values(6,'raksh',62);
insert into EMPLOYEE2 values(6,'raksh',62)
       *
ERROR at line 1:
ORA-20001: Employee age must be lesser than or equal
to 58.
ORA-06512: at "SCOTT.DISPLAY_AGE_CHANGES", line 8
ORA-04088: error during execution of trigger 'SCOTT.DISPLAY_AGE_CHANGES'
```

3]. Write a program to implement cursor in PL/SQL to display the employee details from the following table -emp(eno,ename,designation,doj,salary) .

→

```
SQL> CREATE TABLE EMPLOYEE3(
  2  E_ID NUMBER,
  3  E_NAME VARCHAR(20),
  4  SALARY NUMBER,
  5  PRIMARY KEY (E_ID));
```

Table created.

```
SQL> insert into EMPLOYEE3 values (1,'rach',40000);
```

1 row created.

```
SQL> insert into EMPLOYEE3 values (2,'ammu',50000);
```

1 row created.

```
SQL> insert into EMPLOYEE3 values (3,'avi',24000);
```

1 row created.

```
SQL> insert into EMPLOYEE3 values (4,'tanu',52000);
```

1 row created.

```
SQL> insert into EMPLOYEE3 values (5,'ash',33000);
```

1 row created.

```
SQL> select * from EMPLOYEE3;
```

E_ID	E_NAME	SALARY
1	rach	40000
2	ammu	50000
3	avi	24000
4	tanu	52000
5	ash	33000

```
SQL> CREATE TABLE emp_temp AS
  2   SELECT * FROM employee3;
```

Table created.

```
SQL> DECLARE

 2   CURSOR employee_cur IS
 3    SELECT * FROM emp_temp
 4  FOR UPDATE;
 5  incr_sal NUMBER;
 6  BEGIN
 7   FOR employee_rec IN employee_cur LOOP
 8   IF employee_rec.salary< 25000 THEN
 9  incr_sal := .20;
10   ELSE
11  incr_sal := .10;
12   END IF;
13   UPDATE emp_temp
14   SET salary = salary + salary * incr_sal
15  WHERE CURRENT OF employee_cur;
16   END LOOP;
17  END;
18  /
PL/SQL procedure successfully completed.

SQL> select * from emp_temp;
    E_ID    E_NAME            SALARY
---------- --------------------      ----------
         1        rach                   44000
         2        ammu                   55000
         3        avi                    28800
         4        tanu                   57200
         5        ash                    36300
```

4]. Write a program to implement Procedure in PL/SQL to update the salary of the employee from the following table – employee5(eno,ename,salary) .

→

```
SQL> CREATE TABLE EMPLOYEE5(
  2  E_ID NUMBER,
  3  E_NAME VARCHAR(20),
  4  SALARY NUMBER,
  5  PRIMARY KEY (E_ID));
Table created.

SQL> insert into EMPLOYEE5 values (1,'ash',13000);
1 row created.
SQL> insert into EMPLOYEE5 values (2,'anu',35000);
1 row created.
SQL> insert into EMPLOYEE5 values (3,'shree',53000);
1 row created.
SQL> insert into EMPLOYEE5 values (4,'sam',16000);
1 row created.
SQL> insert into EMPLOYEE5 values (5,'jhon',26000);
1 row created.

SQL> select * from EMPLOYEE5;
    E_ID  E_NAME          SALARY
---------- -------------------- ----------
         1  ash                  13000
         2  anu                  35000
         3  shree                53000
         4  sam                  16000
         5  jhon                 26000

SQL> CREATE OR REPLACE PROCEDURE adjust_salary
  2  IS
  3  BEGIN
```

```
 4  UPDATE EMPLOYEE5 set salary = salary * 1.1 WHERE salary>25000;
 5  UPDATE EMPLOYEE5 set salary = salary * 1.2 WHERE salary<25000;


6  END;
 7  /
```

Procedure created.

SQL> Exec adjust_salary;

PL/SQL procedure successfully completed.

SQL> select * from EMPLOYEE5;

E_ID	E_NAME	SALARY
1	ash	15600
2	anu	38500
3	shree	58300
4	sam	19200
5	jhon	28600

5]. Write a program to implement packages in PL/SQL by using following schema – EMPLOYEE12(ID NUMBER,NAME VARCHAR(20),AGE NUMBER,ADDRESS VARCHAR(20),SALARY NUMBER, PRIMARY KEY (ID)); i. Create the package for adding, removing and listing a customer. ii. Display suitable output.

→

```
SQL> CREATE TABLE EMPLOYEE12(
  2  ID NUMBER,
  3  NAME VARCHAR(20),
  4  AGE NUMBER,
  5  ADDRESS VARCHAR(20),
  6  SALARY NUMBER,
  7  PRIMARY KEY (ID));
Table created.
SQL>  insert into EMPLOYEE12 values (1,'tanu',21,'davangere',30000);
1 row created.
SQL>  insert into EMPLOYEE12 values (2,'ram',22,'tumkur',40000);
1 row created.
SQL> insert into EMPLOYEE12 values (3,'shru',33,'hubli',34000);
1 row created.
SQL> insert into EMPLOYEE12 values (4,'priya',55,'davangere',42000);
1 row created.
SQL> insert into EMPLOYEE12 values (5,'ash',24,'hubli',53000);
1 row created.
SQL>  create or replace package e_pack as
  2  procedure addemp
  3  (
  4   e_id employee12.id%type,
  5    e_name employee12.name%type,
  6    e_age employee12.age%type,
  7    e_addr employee12.address%type,
  8    e_sal employee12.salary%type);
  9  procedure delemp(e_id employee12.id%type);
 10  procedure listemp;
 11  end e_pack;
```

```
12 /
```

Package created.

```
SQL>  create or replace package body e_pack as
 2  procedure addemp(e_id employee12.id%type,
 3  e_name employee12.name%type,
 4  e_age employee12.age%type,
 5  e_addr employee12.address%type,
 6  e_sal employee12.salary%type)
 7  is
 8  begin
 9  insert into employee12(id, name,age,address,salary)
10   values(e_id, e_name,e_age,e_addr,e_sal);
11   end addemp;
12   procedure delemp(e_id employee12.id%type) is
13   begin
14   delete from employee12 where id=e_id;
15   end delemp;
16   procedure listemp is
17    cursor e_emp is
18    select name from employee12;
19    TYPE e_list is table of employee12.name%type;
20  name_list e_list := e_list();
21  counter integer := 0;
22  begin
23  for n in e_emp loop
24  counter := counter +1;
25  name_list.extend;
26  name_list(counter) := n.name;
27  dbms_output.put_line('employee('||counter||')'||name_list(counter));
28  end loop;
29  end listemp;
30  end e_pack;
31  /
```

Package body created.

```
SQL> Set Serveroutput on;
SQL> DECLARE
  2   code EMPLOYEE12.id%type:=1;
  3  BEGIN
  4   e_pack.addemp(6,'anu',39,'tumkur',3600);
  5   e_pack.listemp;
  6   e_pack.delemp(code);
  7   e_pack.listemp;
  8  END;
  9  /
employee(1)tanu
employee(2)ram
employee(3)shru
employee(4)priya
employee(5)ash
employee(6)anu
employee(1)ram
employee(2)shru
employee(3)priya
employee(4)ash
employee(5)anu
PL/SQL procedure successfully created
```

CYCLE 2: MONGODB

1]. Querying Data using MongoDB

1. Create a collection named "products" with fields: "name", "category", "price", and "stock".
2. Insert multiple documents into the "products" collection.
3. Write queries to find all products in a specific category.
4. Retrieve products with a price less than a certain value.
5. Find products that are out of stock.
6. Count the number of products in a specific category.

→

```
test> db.createCollection("products")
   { ok: 1 }


test> db.products.insertOne([{name: "Products1",category:"Electronics",price:499.99,stock:10}])
 {
 acknowledged: true,
 insertedId: ObjectId('658ea5abfff6c4b3250daccc')
}


test>db.products.insertMany([{name:"Products2",category:"Clothings",price:299.99,stock:20},{name:"products3",cat
egory:"Electronics",price:899.99,stock:5},{name:"products4",category:"Books",price:300.0,stock:8}])
{
 acknowledged: true,
 insertedIds: {
   '0': ObjectId('658ea694fff6c4b3250daccd'),
   '1': ObjectId('658ea694fff6c4b3250dacce'),
   '2': ObjectId('658ea694fff6c4b3250daccf')
 }
}


test> db.products.find({category:"Electronics"})
[
 {
   _id: ObjectId('658ea59cfff6c4b3250daccb'),
   name: 'Products1',
   category: 'Electronics',
   price: 499.99,
   stock: 10
 },
 {
   _id: ObjectId('658ea694fff6c4b3250dacce'),
```

```
    name: 'products3',
    category: 'Electronics',

price: 899.99,
    stock: 5
  }
]

  test> db.products.find({category:'Electronics'})
   [
  {
    _id: ObjectId('658ea59cfff6c4b3250daccb'),
    name: 'Products1',
    category: 'Electronics',
    price: 499.99,
    stock: 10
  },
  {
    _id: ObjectId('658ea694fff6c4b3250dacce'),
    name: 'products3',
    category: 'Electronics',
    price: 899.99,
    stock: 5
  }
  ]

test> db.products.find({price:{$lt:300.0}})
[
  {
    _id: ObjectId('658ea694fff6c4b3250daccd'),
    name: 'Products2',
    category: 'Clothings',
    price: 299.99,
    stock: 20
  }
]

test> db.products.find({price:{$lt:50.0}})

test> db.products.find({stock:0})

test> db.products.find({stock:10})
```

```
[
  {
    _id: ObjectId('658ea59cfff6c4b3250daccb'),

   name: 'Products1',
    category: 'Electronics',
    price: 499.99,
    stock: 10
  }
]

test> db.products.find({price:{$lt:301.0}})
[
  {
    _id: ObjectId('658ea694fff6c4b3250daccd'),
    name: 'Products2',
    category: 'Clothings',
    price: 299.99,
    stock: 20
  },
  {
    _id: ObjectId('658ea694fff6c4b3250daccf'),
    name: 'products4',
    category: 'Books',
    price: 300,
    stock: 8
  }
]

test> db.products.countDocuments({category:'Electronics'})
2

test> db.products.countDocuments({category:'Books'})
1
```

2]. Aggregation Framework

1. Create a collection named "orders" with fields: "order_date", "total_amount", and "customer_id".

2. Insert orders into the "orders" collection.

3. Write an aggregation pipeline to calculate the total sales per month.

4. Calculate the average order amount for each customer.

5. Find the customer with the highest total order amount.

→

```
test> db.createCollection("orders")

{ ok: 1 }

test>db.orders.insertMany([{ord_date:ISODate("2023-01-
15T12:30:00Z"),total_amount:100.50,cust_id:1},{ord_date:ISODate("2023-01-
16T01:30:00Z"),total_amount:75.00,cust_id:2},{ord_date:ISODate("2023-01-
17T02:30:00Z"),total_amount:120.00,cust_id:3},])

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('658ea911fff6c4b3250dacd0'),
    '1': ObjectId('658ea911fff6c4b3250dacd1'),
    '2': ObjectId('658ea911fff6c4b3250dacd2')
  }
}

test> db.orders.aggregate([])

[
  {
    _id: ObjectId('658ea911fff6c4b3250dacd0'),
    ord_date: ISODate('2023-01-15T12:30:00.000Z'),
    total_amount: 100.5,
    cust_id: 1
  },
  {
    _id: ObjectId('658ea911fff6c4b3250dacd1'),
    ord_date: ISODate('2023-01-16T01:30:00.000Z'),
    total_amount: 75,
    cust_id: 2
  },
  {
    _id: ObjectId('658ea911fff6c4b3250dacd2'),
    ord_date: ISODate('2023-01-17T02:30:00.000Z'),
    total_amount: 120,
    cust_id: 3
  }
]

test>db.orders.aggregate([{$group:{_id:{$month:"$ord_date"},totalSales:{$sum:"$total_amount"}}},{$project:{_id:0
,month:"$_id",totalSales:1}},{$sort:{month:1}}])

[ { totalSales: 295.5, month: 1 } ]
```

```
test>db.orders.aggregate([{$group:{_id:"$cust_id",averageOrderAmount:{$avg:"$total_amount"}}},{$project:{_id:0,
cust_id:"$_id",averageOrderAmount:1}}])

[
  { averageOrderAmount: 75, cust_id: 2 },
  { averageOrderAmount: 100.5, cust_id: 1 },
  { averageOrderAmount: 120, cust_id: 3 }
]

test>db.orders.aggregate([{$group:{_id:"$cust_id",totalOrderAmount:{$sum:"$total_amount"}}},{$sort:{totalOrderA
mount:-1}},{$limit:1}])

[ { _id: 3, totalOrderAmount: 120 } ]
```

3].Implement MongoDB queries using CRUD operations for a restaurants collection
For the structure of "Restaurants" collection
1. Write a MongoDB query I. to display all the documents in the collection restaurants
2. To display the fields restaurant_id, name, borough and cuisine for all the documents in the Collection restaurant.
3. To display the fields restaurant_id, name, borough and cuisine, but exclude the field _id for all the documents in the collection restaurant
4. To display all the restaurant which is in the borough Bronx
5. To display the first 5 restaurant which is in the borough Bronx.
6. To find the restaurants who achieved a score more than 90
7. To find the restaurants that achieved a score, more than 80 but less than 10

→

```
test> db.createCollection("restaurants")
{ ok: 1 }

test>db.restaurants.insertMany([{rest_id:1,name:"rev",borough:"gobi",cuisine:"eggrice"},{rest_id:2,borough:"rome",cuisine:"dosa"},{rest_id:3,name:"vince",borough:"spain",cuisine:"idli"},])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('658eaca2fff6c4b3250dacd3'),
    '1': ObjectId('658eaca2fff6c4b3250dacd4'),
    '2': ObjectId('658eaca2fff6c4b3250dacd5')
  }
}

test> db.resturants.find({},{rest_id:1,name:1,borough:1,cuisine:1})

test> db.resturants.find({},{rest_id:1,name:1,borough:1,cuisine:1,_id:0})

test> db.restaurants.find({borough:"gobi"})
[
  {
    _id: ObjectId('658eaca2fff6c4b3250dacd3'),
    rest_id: 1,
    name: 'rev',
    borough: 'gobi',
```

```
    cuisine: 'eggrice'
  }

]
test> db.restaurants.find({borough:"gobi"}).limit(5)
[
  {
    _id: ObjectId('658eaca2fff6c4b3250dacd3'),
    rest_id: 1,
    name: 'rev',
    borough: 'gobi',
    cuisine: 'eggrice'
  }
]

test> db.restaurants.find({"grades.score":{$gt:90}})

test> db.restaurants.find({"grades.score":{$lt:90}})

test>db.restaurants.insertMany([{rest_id:1,name:"rev",borough:"gobi",cuisine:"eggrice",score:50},{rest_id:2,borough:
"rome",cuisine:"dosa",score:91},{rest_id:3,name:"vince",borough:"spain",cuisine:"idli",score:88},])

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('658eade2fff6c4b3250dacd6'),
    '1': ObjectId('658eade2fff6c4b3250dacd7'),
    '2': ObjectId('658eade2fff6c4b3250dacd8')
  }
}

test> db.restaurants.find({"grades.score":{$lt:90}})

test>db.restaurants.insertMany([{rest_id:1,name:"rev",borough:"gobi",cuisine:"eggrice",grades:50},{rest_id:2,boroug
h:"rome",cuisine:"dosa",grades:91},{rest_id:3,name:"vince",borough:"spain",cuisine:"idli",grades:88},])
```

```
{
  acknowledged: true,

  insertedIds: {
    '0': ObjectId('658eae04fff6c4b3250dacd9'),
    '1': ObjectId('658eae04fff6c4b3250dacda'),
    '2': ObjectId('658eae04fff6c4b3250dacdb')
  }
}

test> db.restaurants.find({"grades.score":{$lt:90}})

test> db.restaurants.find({"grades":{$lt:90}})
[
  {
    _id: ObjectId('658eae04fff6c4b3250dacd9'),
    rest_id: 1,
    name: 'rev',
    borough: 'gobi',
    cuisine: 'eggrice',
    grades: 50
  },
  {
    _id: ObjectId('658eae04fff6c4b3250dacdb'),
    rest_id: 3,
    name: 'vince',
    borough: 'spain',
    cuisine: 'idli',
    grades: 88
  }
]

test> db.restaurants.find({"grades":{$gt:90}})
[
```

```
  {
    _id: ObjectId('658eae04fff6c4b3250dacda'),
    rest_id: 2,

 borough: 'rome',
    cuisine: 'dosa',
    grades: 91
  }
]

test> db.restaurants.find({"grades":{$gt:80,$lt:100}})
[
  {
    _id: ObjectId('658eae04fff6c4b3250dacda'),
    rest_id: 2,
    borough: 'rome',
    cuisine: 'dosa',
    grades: 91
  },
  {
    _id: ObjectId('658eae04fff6c4b3250dacdb'),
    rest_id: 3,
    name: 'vince',
    borough: 'spain',
    cuisine: 'idli',
    grades: 88
  }
]
```

CYCLE 3 : HADOOP AND HIVE

1] Execute the following commands in HADOOP.

A. To get the list of directories and files at the root of HDFS.

B. To get the list of complete directories and files of HDFS.

C. To create a directory (say, sample) in HDFS.

D. To copy a file from local file system to HDFS.

E. To copy a file from HDFS to local file system

F. To copy a file from local file system to HDFS via copy From Local command

G. To copy a file from Hadoop file system to local file system via copy To Local

H. To display the contents of an HDFS file on console

I. To copy a file from one directory to another directory

J. To remove a directory HDFS.

→

hadoop@hadoopvm-virtual-machine:~$ start-dfs.sh

Starting namenodes on [localhost]

localhost: namenode is running as process 2936. Stop it first.

Starting datanodes

localhost: datanode is running as process 3092. Stop it first.

Starting secondary namenodes [hadoopvm-virtual-machine]

hadoopvm-virtual-machine: secondarynamenode is running as process 3328. Stop it first.

hadoop@hadoopvm-virtual-machine:~$ start-yarn.sh

Starting resourcemanager

resourcemanager is running as process 3531. Stop it first.

Starting nodemanagers

localhost: nodemanager is running as process 3671. Stop it first.

hadoop@hadoopvm-virtual-machine:~$ jps

3328 SecondaryNameNode

3092 DataNode

3671 NodeManager

2936 NameNode

3531 ResourceManager

7871 Jps

hadoop@hadoopvm-virtual-machine:~$ hadoop fs -ls /

Found 6 items

drwxr-xr-x - hadoop supergroup 0 2023-12-06 03:12 /20CS042

drwxr-xr-x - hadoop supergroup 0 2023-12-06 03:31 /20cs042

drwxr-xr-x - hadoop supergroup 0 2023-12-06 04:39 /Lingaraju

drwxr-xr-x - hadoop supergroup 0 2023-12-06 22:43 /anu

drwxr-xr-x - hadoop supergroup 0 2023-12-06 21:54 /cse

drwxr-xr-x - hadoop supergroup 0 2023-12-06 22:09 /csse

hadoop@hadoopvm-virtual-machine:~$ hadoop fs -ls -R /

drwxr-xr-x - hadoop supergroup 0 2023-12-06 03:12 /20CS042

-rw-r--r-- 1 hadoop supergroup 12 2023-12-06 03:08 /20CS042/lingu1.txt

drwxr-xr-x - hadoop supergroup 0 2023-12-06 03:31 /20cs042

-rw-r--r-- 1 hadoop supergroup 7 2023-12-06 03:29 /20cs042/Rohit1.txt

drwxr-xr-x - hadoop supergroup 0 2023-12-06 04:39 /Lingaraju

drwxr-xr-x - hadoop supergroup 0 2023-12-06 04:24 /Lingu

-rw-r--r-- 1 hadoop supergroup 8 2023-12-06 04:24 /Lingu/hitman3.txt

drwxr-xr-x - hadoop supergroup 0 2023-12-06 22:43 /anu

-rw-r--r-- 1 hadoop supergroup 19 2023-12-06 22:43 /anu/file.txt

drwxr-xr-x - hadoop supergroup 0 2023-12-06 21:54 /cse

drwxr-xr-x - hadoop supergroup 0 2023-12-06 22:09 /csse

drwx------ - hadoop supergroup 0 2021-11-30 22:24 /tmp/hadoop-yarn/staging/hadoop

drwx------ - hadoop supergroup 0 2021-11-30 22:55 /tmp/hadoop-yarn/staging/hadoop/.staging

drwx------ - hadoop supergroup 0 2021-11-30 22:24 /tmp/hadoop-yarn/staging/hadoop/.staging/job_1638279301336_0001

-rw-r--r-- 10 hadoop supergroup 40623961 2021-11-30 22:24 /tmp/hadoop-yarn/staging/hadoop/.staging/job_1638279301336_0001/job.jar

-rw-r--r-- 10 hadoop supergroup 315 2021-11-30 22:24 /tmp/hadoop-yarn/staging/hadoop/.staging/job_1638279301336_0001/job.split

-rw-r--r-- 1 hadoop supergroup 38 2021-11-30 22:24 /tmp/hadoop-yarn/staging/hadoop/.staging/job_1638279301336_0001/job.splitmetainfo

-rw-r--r-- 1 hadoop supergroup 366892 2021-11-30 22:24 /tmp/hadoop-yarn/staging/hadoop/.staging/job_1638279301336_0001/job.xml

drwx------ - hadoop supergroup 0 2021-11-30 22:24 /tmp/hadoop-yarn/staging/hadoop/.staging/job_1638279301336_0002

-rw-r--r-- 10 hadoop supergroup 40623961 2021-11-30 22:24 /tmp/hadoop-yarn/staging/hadoop/.staging/job_1638279301336_0002/job.jar

-rwxrwx---1 hadoop supergroup26759 2021-11-30 22:55 /tmp/hadoop-yarn/staging/history/done_intermediate/hadoop/job_1638292941265_0001-1638293104988-hadoop-INSERT+INTO+TABLE+stu...6%27%29%2C%28%27Dhruv%27-1638293135338-1-1-SUCCEEDED-default-1638293115975.jhist

-rwxrwx--- 1 hadoop supergroup 488 2021-11-30 22:55 /tmp/hadoop-yarn/staging/history/done_intermediate/hadoop/job_1638292941265_0001.summary

-rwxrwx--- 1 hadoop supergroup 424571 2021-11-30 22:55 /tmp/hadoop-yarn/staging/history/done_intermediate/hadoop/job_1638292941265_0001_conf.xml

drwx-wx-wx - hadoop supergroup 0 2021-11-30 22:59 /tmp/hive

drwx-wx-wx - hadoop supergroup 0 2021-11-30 22:59 /tmp/hive/resultscache

drwx------ - hadoop supergroup 0 2023-12-06 23:30 /tmp/hive/hadoop

drwxr-xr-x - hadoop supergroup 0 2021-11-30 20:36 /user

drwxr-xr-x - hadoop supergroup 0 2021-11-30 20:36 /user/hive

drwxrwxr-x - hadoop supergroup 0 2021-11-30 22:23 /user/hive/warehouse

drwxr-xr-x - hadoop supergroup 0 2021-11-30 22:55 /user/hive/warehouse/student

-rw-r--r-- 1 hadoop supergroup 43 2021-11-30 22:55 /user/hive/warehouse/student/000000_0

hadoop@hadoopvm-virtual-machine:~$ hadoop fs -mkdir /20cs089

hadoop@hadoopvm-virtual-machine:~$ cd /home/hadoop/Desktop

hadoop@hadoopvm-virtual-machine:~/Desktop$ mkdir 20cs089

hadoop@hadoopvm-virtual-machine:~/Desktop$ cd /home/hadoop/Desktop/20cs089

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ touch tanu.txt

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ hadoop fs -put /home/hadoop/Desktop/20cs089/tanu.txt /20cs089/ex.txt

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ Hadoop fs -get /20cs089/ex.txt /home/hadoop/Desktop/20cs089/test.txt

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ hadoop fs -copyFromLocal /home/hadoop/Desktop/20cs089/tanu.txt /20cs089/t1.txt

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ hadoop fs -copyToLocal /20cs089/t1.txt /home/hadoop/Desktop/20cs089/t2.txt

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ hadoop fs -put /home/hadoop/Desktop/20cs089/tanu.txt /20cs089/t4.txt

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ hadoop fs -cat /20cs089/t4.txt

Hii ,Welcome To SSIT

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ hadoop fs -mkdir /20cs088

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ hadoop fs -cp /20cs089/tanu.txt /20cs088

hadoop@hadoopvm-virtual-machine:~/Desktop/20cs089$ hadoop fs -rm -r /20cs088

Deleted /20cs088

2] Execute the following commands in HIVE

A. To create a database named "STUDENTS" with comments and database pro

B. To display the list of all databases

C. To describe the database

D. To describe the extended database

E. To alter the database properties

F. To make the database as current working database

G. To drop database

H. To create managed table named 'STUDENT'.

I. To describe the "STUDENT" table

J. To create external table name "EXT_STUDENT".

K. To load data into the table from file named student.tsv

→

hadoop@hadoopvm-virtual-machine:~$ start-dfs.sh

Starting namenodes on [localhost]

localhost: namenode is running as process 4854. Stop it first.

Starting datanodes

localhost: datanode is running as process 4998. Stop it first.

Starting secondary namenodes [hadoopvm-virtual-machine]

hadoopvm-virtual-machine: secondarynamenode is running as process 5184. Stop it first.

hadoop@hadoopvm-virtual-machine:~$ start-yarn.sh

Starting resourcemanager

resourcemanager is running as process 5450. Stop it first.

Starting nodemanagers

localhost: nodemanager is running as process 5590. Stop it first.

hadoop@hadoopvm-virtual-machine:~$ hive

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.

SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Hive Session ID = 589c0af1-42e6-4003-8642-a4f9c0203ad3

Logging initialized using configuration in jar:file:/home/hadoop/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true

Hive Session ID = ae967405-9358-43b7-b515-d9b756a95489

Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.

hive> show databases;

OK

ssit_student

student1

student20

students

Time taken: 0.608 seconds, Fetched: 5 row(s)

hive> create database if not exists ssit comment'Student details' with dbproperties('creator'='Tanusha');

OK

Time taken: 0.272 seconds

hive> show databases;

OK

default

ssit

ssit_student

student1

student20

students

Time taken: 0.034 seconds, Fetched: 6 row(s)

hive> describe database ssit;

OK

ssit Student details hdfs://localhost:9000/user/hive/warehouse/ssit.db hadoop USER

Time taken: 0.047 seconds, Fetched: 1 row(s)

hive> describe database extended ssit;

OK

ssit Student details hdfs://localhost:9000/user/hive/warehouse/ssit.db hadoop USER {creator=Tanusha}

Time taken: 0.033 seconds, Fetched: 1 row(s)

hive> alter database ssit set dbproperties('Edited-By'='tan');

OK

Time taken: 0.234 seconds

```
hive> describe database extended ssit;
OK
ssit	Student details	hdfs://localhost:9000/user/hive/warehouse/ssit.db Hadoop USER {creator=Tanusha, Edited-By=tan}
Time taken: 0.026 seconds, Fetched: 1 row(s)
hive> use ssit;
OK
Time taken: 0.023 seconds
hive> create table if not exists cse(rollno int,name string) row format delimited fields terminated by '\t';
OK
Time taken: 0.519 seconds
hive> describe cse;
OK
rollno		int
name		string
Time taken: 0.085 seconds, Fetched: 2 row(s)
hive> create external table if not exists ise(rollno int,name string) row format delimited fields terminated by '\t';
OK
Time taken: 0.093 seconds
hive> desc ise;
OK
rollno		int
name		string
Time taken: 0.057 seconds, Fetched: 2 row(s)
hive> load data local inpath '/home/hadoop/Desktop/stude.tsv' overwrite into table ise;
Loading data to table ssit.ise
OK
Time taken: 0.232 seconds
hive> select * from ise;
OK
100	tanu
101	yashu
102	avi
Time taken: 0.139 seconds, Fetched: 3 row(s)
```