ch. Mohana siva Balaji

23295-AI-010

Index

S. NO	Name of the Experiment	Date	signature
1.	Know installation of oracle.		
2.	Exercise on creating tables.		
3.	Exercise on inserting records.		
4.	Exercise on updating records.		
5.	Exercise on modifying the structure of the table.		
6.	Exercise on SELECT command.		
7.	Exercise on querying the table using clauses like WHERE, ORDER IN, AND, OR, NOT, IS NULL		
8.	Exercise on GROUP BY HAVING		
9.	Exercise on number functions, character functions, conversion functions, and date functions, group functions.		
10.	Exercise ON SET operators.		
11.	Exercise on sub queries.		
12.	Exercise on joins.		
13.	Exercise on various data and number from at models.		

14.	Exercise on creating tables with integrity constraints.		
15.	write programs using PLISQL control statements.		
16.	Exercise on procedure.		
17.	Exercise on Functions.		
18.	Exercise on cursors.		
19.	Exercise on Triggers.		
20	Exercise on Installation of MongoDB.		
21.	Exercise on creation and Dropping of database.		
22.	Exercise on creation and Dropping of collections.		
23	Exercise on commands of MongoDB - Insert, update, find, delete and sorting of Documents.		

generations a cloud cloud applications ouy company infra structure.

Autmous data base Enterprise resource careers
data management sales developer center

on-promise
infra structure
oracle data base
Java
software

Product Help
cloud Readlines
downloads
support
training.

Data base 19c enterprise / standard editions
Data base 18c enterprise / standard editions
Data base 12c enterprise / standard editions

Data base 11g enterprise / standard editions

Data base 18c express edition

Berkeley DB

big data connectors

Experiment - 1

Aim: know installation of circle.

step 1: go to oracle.com and click on options menu

step 2: click the download button and scroll docon to database section.

Name	Doconload	Note
Microsoft window (32-bit)	(2GB)	see all
Microsoft windows (x64)	file' file 2 (2GB)	see all
unix x86	file' $file_2$ (2GB)	sell all

you must accept the orcale license argreement to doconolad this software

I accept the oracle license argreement

Download usm64-11g R2-data base-101-22ip

All the content on this site

oracle account sign in

username ①

password ①

sign in

Need help?

Dont have an oracle account

create account

step 3: click
editions, af
versions
the files

step
will
nee
ha
-c

step 3: click data base 11g enterprise/ standard editions. after which you'll find different versions of orcale for different os download the files according to year os.

step 4: After clicking the download button the page will be directed to login screen where you'll need to sig in oracle account. If you don't have one, then you must sign up, because without that you won't be able to download the files.

name	Data modified	type	size
win 64 - 11g - R2 data base - 1 of 2	06 - 02 - 2020 00:47	winRAR zip archeive	11,85,061
win 64 - 11g - R2 data base - 2 of 2	06 - 02 - 2020 11:00	winRAR zip archeive	9,84,365

Name	Data modified	Type
database	07-02-2020 11:20	file floder
main database	31-03-2010 06:48	file floder

Name	Data modified	Type
orade.ctx	30-03-2010 19:21	file floder
orade.oowb2sf	30-03-2010 19:54	file floder
orade.rdbms install	30-03-2010 21:47	file floder
orade.sysman	30-03-2010 19:38	file floder
console.db		

step 6: Repeat the same steps for both the files and download them.

After downloading the files successfully you'll find both files in downloads folder where both of them will be in compressed form, so you'll need to extra them.

Extraction of downloaded files

step 1: Extra both files with the use of winRAR or way other extraction tool of your choice in the same directory.

step 2: Go to data base -> stage -> components the content of this folder.

step 3: Go to main database -> stage paste the copied content in components folder.

Name	Data modified	Type	Size
doc	24-03-2010 12:48	file folder	
Install	30-03-2010 20:39	file folder	
Response	30-03-2010 22:01	file folder	
Stage	30-03-2010 22:10	file folder	
set-up	12-03-2010 14:41	application	334 KB
welcome	17-03-2010 2:12	chrome HTML	6 KB

oracle data base 11g Release 2 Installer

[INS-13001] environment does not meet minimum requirement

Are you sure you want to continue?

[yes] [NO] [Details]

configure security updates

Installation option

grid Installation option

Install type

Typical Installation

provide your email address to be informed of security issues install the product & initiate configuration manager view details.

email: [ ]

[V] I wish to receive security

My oracle password: [ ]

Help [Back] [next >] [finish] [cancel]

Installation of oracle, DataBase

step 1: Go to main database folder cotore you'll find setup right click the stop exe file and choose Run as administrator.

step 2: click yes to continue this will stast oracle universal installer.

step 3: provide your e-mail address to receive all the notifications and news alerts from oracle.

select Installation option

- configure security updates
- Installation option
- grid Installation option
- Install type
- Typical Installation
- prequisite checks
- summaries
- Install product
- finish

select any of the following Install options.

- ⊖ create & configure a data base
- o Install database software only
- o update an existing database

Help | <Back | Next > | finish | cancel

step 4: sel
ins

option 1:

option 2:

option 3

step 5

system class

oracle database, 11g

- configure security updates
- Installation option
- system class
- Typical Installation
- Pre requistechdy

⊖ Desktop class

⊖ sever class

Help | <Back | Next > | finish | cancel

step 4:- select any of the three different installation options acording to your needs.

option 1:- If you want to install oracle server software and want to create data base also.

option 2:- If you want to install oracle server only.

option 3:- If you want to upgrade your exisisting data base.

step 5:- choose b/w server class and desktop class as per your requirement and click on next

Typical Install configuration — oracle Database 11g

- configure security updates
- Installation option
- system class
- Typical Installation
- prerequisite checks
- summary
- Install product

perform full database Installation with basic configuration

oracle base [ ] Browser

software location [ ] Browser

password Messages: [ ] Browser

Administrative password: [INS-30] The admin password entered doesn't confirm to the oracle recommand.

Help | <Back | Next> | Finish | Cancel

---

summary — oracle

- configure security
- Installation option
- system class
- Typical Installation

- oracle Database 11 grelease 21
  - Installing
  - global settings
  - Inventory information
  - Database information

Help | <Back | next> | finish | cancel

step 6:- configure the basic settings and create a password for your data base once the configuration is done click on next to continue

step 7:- Here, orcale universal installer (out) will check for the pre requisties such as hardware compatibility. If there will be any errors then out will show them here and will ie command the changes.

uction of DBM
aye management
is a collectio
in a way
ed. managed
for string
tively whe
Proper
the da
be
elational
SQL, s
of
in

Introduction of

Data baye management

DBMS is a collection of ste

an a way that data can b

and updated.

and retrevis

Install product

oracle database 11g

configure security updates

Installation option

systems class typical

Installation

pre requisite checks

summary

• Install product

finish

progress

15%.

Extracting the replay 184 jsp from '/app/admin/product\11.20

status

->oracle DB installation In process

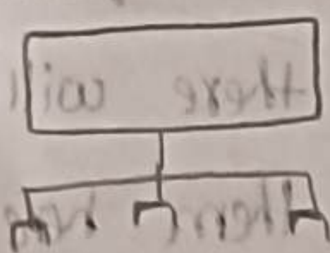
U. prepare successed

-> copy files In progress

• setup files pending

oracle OB pending

configuration



Data base configuration Assistant - □ X

Management

Automation

• ADDM for RAC

• Automatic SQL tuning and memory management

• advisiorpor stream

copying database files

creating and starting oracle instance

completing Database execution

c lore database creation in Progress

100%.

long files for the current are locate.

step 8:- click

process this

depending

step 8:- click on finish to start the Installation process this installation might take some times depending on your hardwork.

Intro
Data base
DBMS is a ... called
... a way that ... and update
... and retrieve

database configuration Assistant

database creation complete for details check log files:
o:\app\admin\cfg\toallogs\dbcall oracle
database information
global data base Name: oracle
system identifier (SID): oracle
server parameter file name:
The data base control urli's https://local host
(ok) 1158/em password management

finish

oracle
Database 11g

configure security
Installation option
system class
Typical installation
pre equistechecks
summary
Install product
finish

Help

The installation oracle data base was successful note:

Enterprise manager DB control - (orcale):

https://local host:1158/em

<

>

step 9: cli

step 10:

open

updated.

and relativis

step 9: click ok to finish the installation.

step 10: copy the level local host link provided to

open your enterprise manager.

oracle enterprise manager 11g

database control

Login

username

Password

connect As Normal

Login

click the close button and you are done with the installation process.

* GO to start menu and
* search oracle folder
* click on data base control oracle (your global data base name)
* This will take you to the login screen of your oracle enterprise manager.

Experiment - 2

Aim :- Exercise on creating tabel.

The SQL create Tabel statement.

the create table statement is used to create a new table in a data base.

syntaxi

create table table name (fields, desc(p) SQL create table example.

the following example create a table called persons that contains file columns: person ID, last name, First name, Address and city;

Ext

Create table person (person ID, int lastname varchar (30), First name varchar (20), Address varchar (20), city varchar (10));

Experiment -3

Aim:

Exercise on inserting Records.

SQL insert statement:-

This statement is used to insert records in a table.

syntax:

insert into table_name values (value-1, value-2, value-3, value-4);

Ex:

insert into student ('Balaji', 010, 500, 'Nandarada');

insert into student ('Ram', 002, 570, 'korukonda');

insert into student ('sai', 027, 550, 'Mandapeta');

insert into student ('Raj', 017, 550, 'kothapeta');

Experiment -4

Aim:

Exercise on updating records.

SQL update statement:

update is used to modify the existing records in a table.

Syntax:

update table name set col-name = value where condition;

Ex:

update student set name = 'sai' where pin = 021;

Experiment -5

Aim:

Exercise on modifying the structure of the table.

SQL Alter statements

Alter table table_name add col_name datatype;

Ex:

Alter table student add grade varchar(20);

Experiment -6

Aim:

Exercise on select command.

SQL select statement:

select statement is used to select data / display data from a database.

- To select for specific columns from a table.

Syntax:

select col_name1, col_name2, ... from table_name;

Ex:

select name, marks from student;

- To select all data from a table.

Syntax:

select * from table_name;

Ex:

select * from student;

Experiment -7

Aim: Exercise on querying the table using clauses like WHERE, ORDER BY, IN, AND, OR, NOT, ISNULL

First create a table with the given details table name 'emp' and fields: Emp_id, name, salary, department.

Ex: CREATE table Emp(Emp-id int, name varchar(10), salary int, department varchar(10));

The SQL where clause: The where clause is used to filter records.

where syntax:

```
SELECT column1, column2 ...
FROM table_name
WHERE condition;
```

where clause example: The following SQL statements select all the employees from the department 'IT' in the Emp table:

Ex:

```
SELECT * FROM Emp
WHERE department = 'IT';
```

The SQL IN operator:

- The IN operator allows you to specify multiple values in a WHERE clause.
- The IN operator is a statement for multiple

or conditions.

IN syntax: SELECT column-name
FROM table-name
WHERE column-name IN(value1, ..);

IN operator example:
. The following SQL statement select all employees that are seleted from Emp-id.

Ext
SELECT * FROM Emp
WHERE Emp-id IN(2,3,5);

The SQL ORDER BY keyboard:
- The ORDER BY keyboard is used to sort the result set in ascending or descending order.

ORDER BY syntax
SELECT column1, column2....
FROM table-name
ORDER BY column1, column2, ... ASC|DESC;

ORDER BY Example1;
SELECT * FROM Emp
ORDER BY salary asc;

ORDER BY Example 2;
SELECT * FROM Emp
ORDER BY salary DESC;

The SQL AND, OR & NOT operators:

AND syntax
SELECT column 1, column 2...

FROM table_name
WHERE condition 1 AND condition 2 AND...

NOT syntax:

SELECT column1, column2
FROM table_name
WHERE NOT condition;

OR syntax:

SELECT column1, column2...
FROM table_name
WHERE condition1 OR condition2 ...

AND Example:

The following SQL statement select all fields from 'emp' where department is 'HR' and salary = 50000.

Example:

SELECT * FROM emp
WHERE department = 'HR' AND salary = 50,000;

OR Example:

The following SQL statement selects all fields from "emp" where department is "sales" or "IT".

Example:

SELECT *FROM Emp
WHERE department = 'sales' OR 'IT';

NOT Example:

The following SQL statements selects all fields from "emp" where salary is Not '65,000'.

Example:

```
SELECT * FROM Emp
WHERE salary NOT 65,000;
```

IS NULL :-

. The ISNULL command is used to test for empty values (NULL). The following SQL lists all emp with a NULL value in the Emp_id field.

Example :-

```
SELECT * FROM emp
WHERE emp_id is NULL;
```

Experiment -8

Aim: Exercise on GROUP BY, HAVING.

Group by:
The Group by command is used to group the result set.

syntax:
```
SELECT column1, aggregate_function (column)
FROM table-name
GROUP BY column1;
```

Example:
```
SELECT Department, AVG (salary) As Avg salary
FROM employees
GROUP BY Department;
```

Having:
The having command is used instead of where with aggregate function.

syntax:
```
SELECT column1, aggregate_function (column2)
FROM table-name
GROUP BY column1
HAVING condition;
```

Example:
```
SELECT Department, AVG (salary) As Avg salary
FROM employee
GROUP BY Department
HAVING AVG (salary) > 5000;
```

Experiment -9

Aim: Exercise on number function, character function, conversion function and date function, group function.

1. Number function:

a. ABS():

. Absolute is the measure of magnitude of a value this function returns absolute value ie is always positive number.

syntax: SELECT ABS(number);

Ex: SELECT ABS(-10);

b. CEILING():

. This function returns the smallest integer value and it is not smaller than the value.

syntax: SELECT CEILING(value);

Ex: SELECT CEILING(3.4);

c. FLOOR():

. This function generates the output after rounding of the decimal. which is equal to or less than the expression value.

syntax: SELECT FLOOR(value);

Ex: SELECT FLOOR(3.6);

d. SQRT():

. This function returns the square root of given value.

syntax: SELECT 'SQRT(value);

Ex: SELECT SQRT(49);

2. character functions:-

a. Initcap():

. This function converts first character to capital letter.

syntax:- SELECT Initcap('name');

Ex:- SELECT Initcap('hello');

b. lower():

. It converts string into lower.

syntax:- SELECT LOWER ('name');

Ex:- SELECT LOWER ('HELLO');

c. upper():

. This function converts string into uppercase.

syntax:- SELECT UPPER ('name');

Ex:- SELECT UPPER ('hello');

3. Date functions:-

a. CURDATE():

. This function returns current date as a value in yyyy-MM-DD format.

syntax:- SELECT CURDATE();

Ex:- SELECT CURDATE();

b. Next-Day():

syntax:- SELECT next-day ('date', 'day');

Ex:- SELECT next-day ('17-sep-24', 'tuesday');

c. last-day():
. this function returns date of last day.
syntax: SELECT last-day('date');
Ex: SELECT last-day('2024-9-17');

4. conversion function():-
. SQL provides three functions to convert value from one data to another.

5. Group function:-

a. AVG():
. This function is used to get the average value of the numeric value.
syntax: SELECT AVG(column-name) FROM table-name;
Ex:- SELECT AVG(E-Sal) FROM EMP;

b. Max():
. This function is used to get maximum value from a column.
syntax: SELECT MAX(column-name) FROM table-name;
Ex: SELECT MAX(E-sal) FROM EMP;

Experiment -10

Aim: Exercise on set operators.

set operator:

- set operator which results of two queries into a single result.
- First create two tables Employees1, Employee2, with given fields: Employee ID, Name, Department, salary. respectively.

UNION:

- combine two result sets and remove duplicates.

syntax

```
SELECT columns-list
FROM table-1

UNION
SELECT columns-list
FROM table-2;
```

Excamplet

```
SELECT Name, Department
FROM Employees1

UNION
SELECT Name, Department
FROM Employees2;
```

UNION ALL:

- combine two results set without removing duplicates.

syntax:

```
SELECT column-list
FROM table-1
```

UNION ALL
SELECT column-list
FROM table_2;
Example:-
SELECT Name, Department
FROM Employees 1
UNION ALL
SELECT Name, Department
FROM Employees 2;

INTERSET :-
. Find common rows between two result sets.

syntaxt
SELECT * FROM table_name1

INTERSET
SELECT * FROM table-name2;

Example :-
SELECT * FROM Employees 1
INTERSET
SELECT * FROM Employees 2;

MINUS;
. Return rows from the first result set that are not in the second result set.

syntaxt
SELECT * FROM table-name1

MINUS
SELECT * FROM table-name2;

Example:

```
SELECT * FROM Employees1
MINUS
SELECT * FROM Employees2;
```

Experiment -11

Aim: Exercise on sub queries.

sub queries:-

A sub query is a queries with in another SQL query and embedded with in the where clause.

1. sub queries with the select statements:-

SQL sub queries are most frequently used with the select statement.

syntax:-

SELECT column-name FROM table_name WHERE column-name expression operator (SELECT column-name from table-name WHERE ....)

Example:-

SELECT * FROM employee WHERE ID IN (SELECT ID FROM employee WHERT salary>4500);

2. sub queries with the insert statements:-

SQL subquery can also be used with the insert statement. In the insert statement, data returned from the subquery is used to insert into another table.

- In the subquery, the selected data can be modifies with only of the characters date functions.

syntax:-

INSERT INTO table-name(column1, column2,...)

SELECT * FROM table_name
WHERE value operator;

Example:

INSERT INTO Employee_BKP
SELECT * FROM employee
WHERE ID IN (SELECT ID FROM employee);

3. sub queries with the update statement:

- The subquery of SQL can be used in conjunction with the update statement. when a sub query is used with the update statement, then either single or multiple columns in a table can be updated.

syntaxt

UPDATE column-name_value
WHERE value operator
(SELECT column-name FROM table-name WHERE condition);

Example:

UPDATE employee set salary = salary * 0.25
WHERE age IN
(SELECT age FROM customers_BKP WHERE age >=29);

4. sub queries with the delete statement:

- The subqueries of SQL can be used in conjunction within the delete statement. Just like any other statements mention above.

syntaxt

DELETE FROM table_name

WHERE value operator

(SELECT column_name FROM table_name WHERE condition);

Example:

DELETE FROM EMP

WHERE ID IN

(SELECT ename from Emp WHERE Eid = 110);

Experiment - 12

Aim: Exercise on joins.

Joins:

Returns rows where there is a match in both tables.

1. Equi join
2. Non-equi join
3. self join
4. Cross join

1. Equi join:

. A join which is based on equality then it is called as equi join.

Example:

```
SELECT * FROM emp, dept
WHERE emp.deptno = deptno;
```

2. Non-Equi join:

. A join which is based on unequality (other than =) then it is called as non-equi join.

Example:

```
SELECT * FROM emp, dept
WHERE emp.deptno = deptno;
```

3. self Join:

. Joining of a table itself then it is called as self join.

Syntax:

```
SELECT * FROM emp1, emp2
```

WHERE e1.dept no = e2.dept no;

4. CROSS JOIN:-

. Cross joins are used to combine each row of one table with each row of another table and return the cartesion product of the sets of rows from the tables that are joined.

Example:-

SELECT * FROM EMP1
CROSS JOIN EMP 2;

Left Join:-

. Returns all rows from the table and matched rows from the right table.

Example:-

SELECT ename
FROM EMP1
Left JOIN Emp1.ename = EMP2.Ename)

Right joint

. Returns all rows from the right table and the matched rows from the left value.

Example:-

SELECT ename FROM EMP1 right JOIN EMP2 on Emp1.ename = Emp2.ename;

FULL JOINT

. Returns all rows when there is a match in one of the tables.

syntax:-

SELECT ename FROM EMP1 FULL JOIN EMP2 on Emp1.ename = emp2.ename;

Experiment -13

Aim: exercise on various date and number format models.

procedure:

- First login into oracle database using username and password.
- Here click on the SQL option then click on SQL command.
- Type the command as shown below.

TO_CHAR: converting character data type.

NUMBER FORMAT ON '999' FORMAT WIDTH SET MY NUMBER ON NINES.

syntax:

select to_char (number, format) from dual;

Ex:

select to_char (2829.20, '9999.99') from dual;

NUMERIC FUNCTION ON '09999' FORMAT ZERO FILLED.

Ex:

select to_char (28, '09999') from dual;

FORMAT PREFIX VALUE WITH A DOLLAR SIGN:

- Numeric function on '$9999' format prefix value with a dollar sign.

Ex:

select to_char (2829, '$9999') from dual;

FORMAT WITH COMMA DELIMETER:

- Numeric function on '9,999' format comma delimeter where indicated.

Ex:-
select to-char(2829, '9,9999') from dual;

SELENTIFIC NOTATION ON '9.99EEEE' FORMAT;

. scientific notation with four format.

Ex:-
select to-char (2829, '9.99eeee') from dual;

DATE FUNCTIONS;

Ex:-
select TO-CHAR (sysdate, 'DD DAY MON YYYY
HH:MI:SS AM DDD') from dual;

Ex:-
select TO-DATE ('2018/02/15 9:45:34', 'YYYY/MM/
DD HH:MI:SS') from dual;

Experiment - 14

Aim: Exercise on creating tables with integrity constraints.

Procedure:

- First login into oracle database using username and password.
- Here click on the SQL option then click on SQL command.
- Then create the table using following integrity constraints.

1. **NOTNULL:**

- when a column is defined as NOTNULL, that column becomes a mandatory column.

syntax:

<columnnames> <datatype> NOTNULL.

Ex:

```
create table student (sid varchar(20) NOT NULL, name
varchar(15), marks number(3));
```

Inserting 'NOTNULL' value into student table.

```
insert into student values ('18030-cm-168', 'josh',
874);
```

2. **UNIQUE:**

- The purpose of unique constraint it to ensure that information in the column in unique.

syntax:

```
(reate table tablename (columnname datatype (size)
unique, columnname datatype (size));
```

Ex:

Create table student (pinno varchar(10) unique, name varchar(16), marks number(3));

Inserting two different pinno columns in student table then only it takes value.

Insert into student values ('18030-cm-321','raju',840);

Insert into student values ('18030-cm-322','raju',850);

3. PRIMARY KEY:

. A primary key is one or more columns in a table used to identify uniquely each row in the table.

syntax:

create table tablename (columnname datatype (size) primary key columnname datatype (size));

ex:

create table student (pinno varchar(20) primary key name varchar(16), marks number(3));

Inserting two same pinno columns into student table.

Insert into student values ('18030-cm-321','raju',840);

Insert into student values (18030-cm-321','raju',880);

Insert into student values (null,'raju',870);

4. FOREIGN KEY:

. Forein key represents realationship between tables.

A foregin key is a column whose values are derived from the primary key or unique key of other table.

syntax:
create table tablename (columnname datatype (size
references tablename, columnname datatype (size));

ex:
create table student (pinno varchar(20) references
student2, college varchar(20));
Insert into student values ('121', 'mic', '14-01');

CHECK:-
• check constraint must be specified as a logical expression that evaluate either true or false.

syntax:
create table tablename (columnname datatype(size)
check (logical exp), columnname datatype (size));

ex:
create table client (clientno varchar(20) check (clientno
like 'c%'), name varchar(20), city varchar(20));
Insert into client values ('12', 'ravi', 'mumbai');
Insert into client values ('c-01', 'rajeev', 'banglore');

Experiment -15

Aim:- write programs using PL/SQL control statements.

procedure:-

selection statements:-

Ex-1:-

. write a pl/sql program to find the given number even or odd.

```
DECLARE
   a number;

BEGIN
   a:=&a;
   if a mod 2=0 then
    dbms_output.put_line ('a is even number');
   else
    dbms_output.put_line ('a is odd number');
   end if;

END;
```

Looping statements:-

Ex:-3

. write a pl/sql program to perform multiplication of a given number upto 10 using for loop.

```
DECLARE
   n1 number :=10;
   n2 number :=1;

BEGIN
   for n2 in 1..10
   loop
```

```
    dbms_output.put_line (n1*n2);
    end loop;
END;
```

Experiment -16

Aim:- Exercise on Procedures.

syntax for creating Procedures:-

CREATE [OR REPLACE] PROCEDURE procedure_name
(parameter_name IN | OUT | IN OUT datatype, ...)
{IS|AS}

BEGIN
< procedure_body >
END produre_name;

Procedure:-

- To get output write the command set server outputon.
- To open or edit a file type edit filename.

Example:-

- write a PL/SQL program to find the square root of given number.

DECLARE
a number;
PROCEDURE squareNum (x IN OUT number) IS
BEGIN
greetings
x := x * x;
END;
BEGIN
a := 23;
squareNum(a);

```
dbms_output.put_line ('square of (75):' ||la);
END;
```

Experiment -17

Aim:- exercise on functions.

syntax:-

CREATE [OR REPLACE] FUNCTION function_name

[(parameter_name [IN | OUT | IN OUT] type [,...])]

RETURN return_datatype;

{IS | AS}

BEGIN

<function_body>

END [function_name];

Procedure:-

- To get output write the command set server-output on.

- To open or edit a file type edit filename.

Example-1:-

- write a pl/sql function for addition of two numbers.

Function creation:-

- create function addition (x number, y number)

return number

is

z number;

begin

z := x+y;

return(z);

end;

program 1-

```
DECLARE
   a number;
   b number;
   c number;
BEGIN
   a := 10;
   b := 20;
   c := addition(a, b);
   dbms_output.put_line('addition is' ||c);
END;
```

Experiment -18

Aim:- Exercise on cursors.

procedure:-

- To get output write the command set serveroutput on.
- To open or edit a file type edit filename.

Implicit cursor:-

- write a plsql program to implement cursors.

```
DECLARE
  vid employee.empid%type;
  vname employee.name%type;
  vphoneno employee.phoneno%type;
  vsalary employee.salary%type;
BEGIN
  vid := &vid;
  select * into vid, name, vphoneno, vsalary from emp
  where empid = vid;
  if sql%found then
  dbms_output.put_line (sql%rowcount || ' rows is
                        deleted');
  else
  dbms_output.put_line ( no rows deleted);
  END IF;
END;
```

Explict cursor:-

syntax

declaring the cursor:

CURSOR cursor_name IS select_statement;

opening the cursor:

OPEN Cursor_Name;

Fetching the cursor:

FETCH cursor_name INTO Columnnames, column...

close the cursor:

close cursorname;

Example :-

DECLARE

cid customers.id%type;

c_name customers.name%type;

c_addr customers.address%type;

CURSOR c_customers is

SELECT id, name, address FROM customers;

BEGIN

open c_customes;

LOOP

FETCH c_customes into c_id, c_name, c_addr;

EXIT WHEN c_customers%notfound;

dbms_output.put_line (c_id||' '||c_name||' '||

c_addr);

END LOOP;

CLOSE C_customers;

END;

Experiment -19

Aim:- Exercise on triggers.

procedure:-

. To get output write the command set server-output on.

. To open or edit a file type edit filename.

syntax for TRIGGER:-

```
CREATE [OR REPLACE] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF}
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
Declaration-statements
BEGIN
Executable-statements
EXCEPTION
Exception-handling-statements
END;
```

Trigger program:-

```
CREATE OR REPLACE TRIGGER display-salary_changes
AFTER DELETE OR INSERT OR UPDATE ON customers
FOR EACH ROW
```

```
WHEN (NEW.ID >0)
DECLARE
sal_diff  number;
BEGIN
sal_diff := :NEW.salary - :OLD.salary;
dbms_output.put_line('old salary :' ||:OLD.salary);
dbms_output.put_line('new salary :' ||:NEW.salary);
dbms_output.put_line('salary difference :' ||sal_diff);
END;
```

Experiment -20

Aim: Exercise on Installation of mongo DB.

Procedure:-

INSTALLATION OF MangoDB.

step 1:- Go to Mongodb Doconload center to download mongoDB community server.

step 2:- when the download is complete open the msi file and click the next button in the strart up.

MongoDB object software

Q signin | Try free

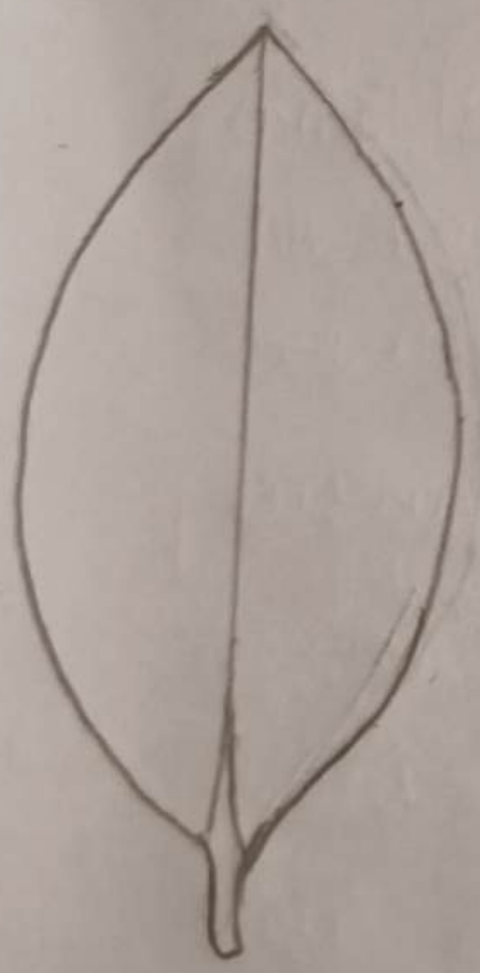
MongoDB Enterprise server

MongoDB Community server
The MongoDB Enterprise server gives you Reviews the Enterprise server too to available

Available Downloads
window
AA3 (current)
platform
windows
packages
MSY

Download | copy link

MongoDB 4.4.2 2000BRPlus SS (64 bits)



welcome to the MongoDB 4.4.2 2008R2Plus SS2 (64 bits) setup wizard

The setup wizard will install mongodB 4.4.2 2008R2Plus SS2 (64 bit) on your computer.

Back | Next | cancel

step 3:- Now accept the end-user license agreement and click the next button.

step 4:- Now select the complete option to install all the program features. Here, if you can want to install only selected program features and want to select the location of the instalation, then use, the customoption.

MongoDB 4.4.2 2008R2plus SSL (64 bit) setup

END-user License Agreement

please read the following license carefully.

SERVER SIDE PUBLIC LICENCESE

VERSION 1, OCTOBER 16, 2018

Copyright 2018 MongoDB, Inc,

Everyone is permitted to copy and distribute verbatim changine is allocued.

TERMS AND CONDITIONS

I accept the terms

Print | Back | Next | Cancel

MongoDB 4.4.2 2008R2plus SSL (64 bit) setup

choose setup TYPE

choose the setup that best needs.

Complettee

All program features will be installed. Requires the most disk space.

custom

Allows users to choose which program features will be installed and where they will be installed.

Back | Next | Cancel

step 3: Now

agreement an

step 4: N

all the

to instal

want

then u

step 5:- Select "Run service as Network service user" and copy the path of the data directory. click Next.

step 6:- click the install button to start the installation process.

step 7:- After clicking on the install button install-ation of MongoDB begins.

MongoDB 4.4.2 2008 R2 Plus SSL (64bits)

service configuration
specify optional setting to configue as service

☑ install mongodb as service
◉ Run service as network service domain.
○ Run service as

Account Domain:
Account name: MongoDB
Account password:

service Name: MongoDB
data Directory: C:\Program Files\MongoDB\Server\4.4\data\
log Directory: C:\Program Files\MongoDB\Server\4.4\log\

<Back | Next > | Cancel

MongoDB 4.4.2 2008R2Plus SSL (64 bit) setup

Ready to install MongoDB 4.4.2 2008R2Plus

click install to begin the installation. click Back to review or change any of your installation settings.

Back | Install | cancel

MongoDB 4.4.2 2008R2Plus SSL (64 bit) setup

Installing MongoDB 4.4.2 2008R2Plus SSL (64 bit)

please wait while the setup wizard to installs MongoDB 4.4.2 2008R2Plus SSL (64 bit).

Status: copying new files

step 6: selec
and copy
click Next

step6's c
installat

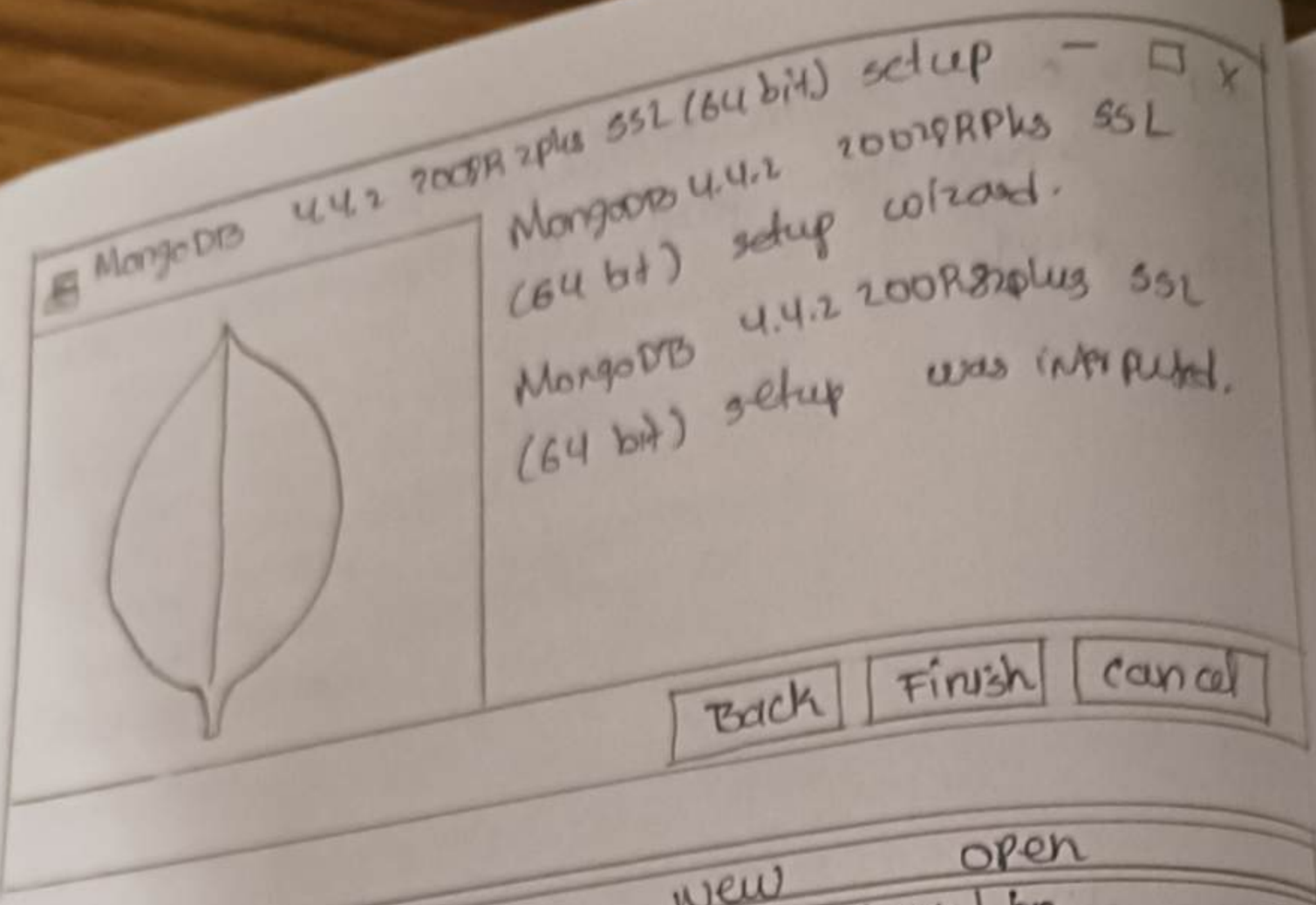
step

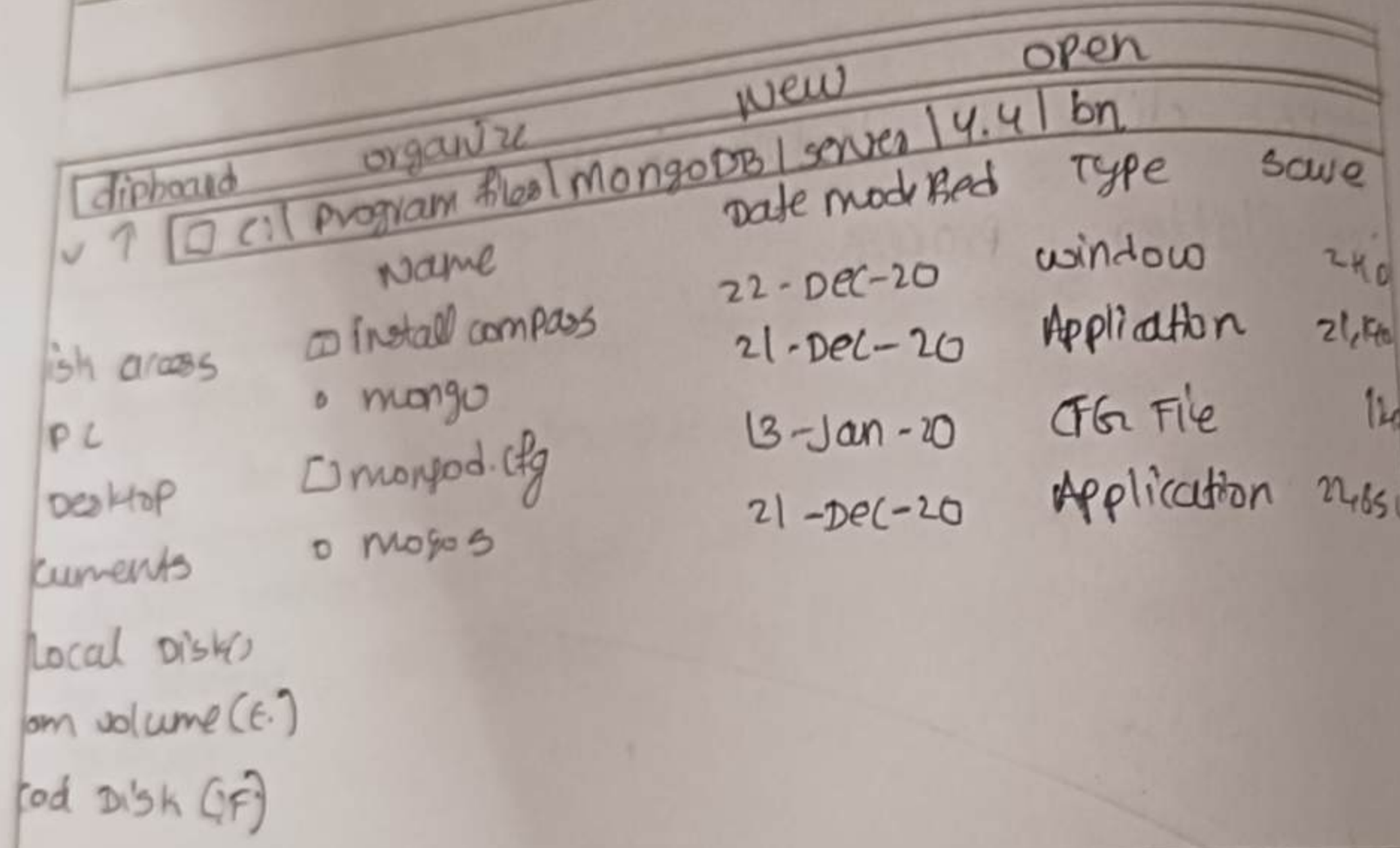
-ati

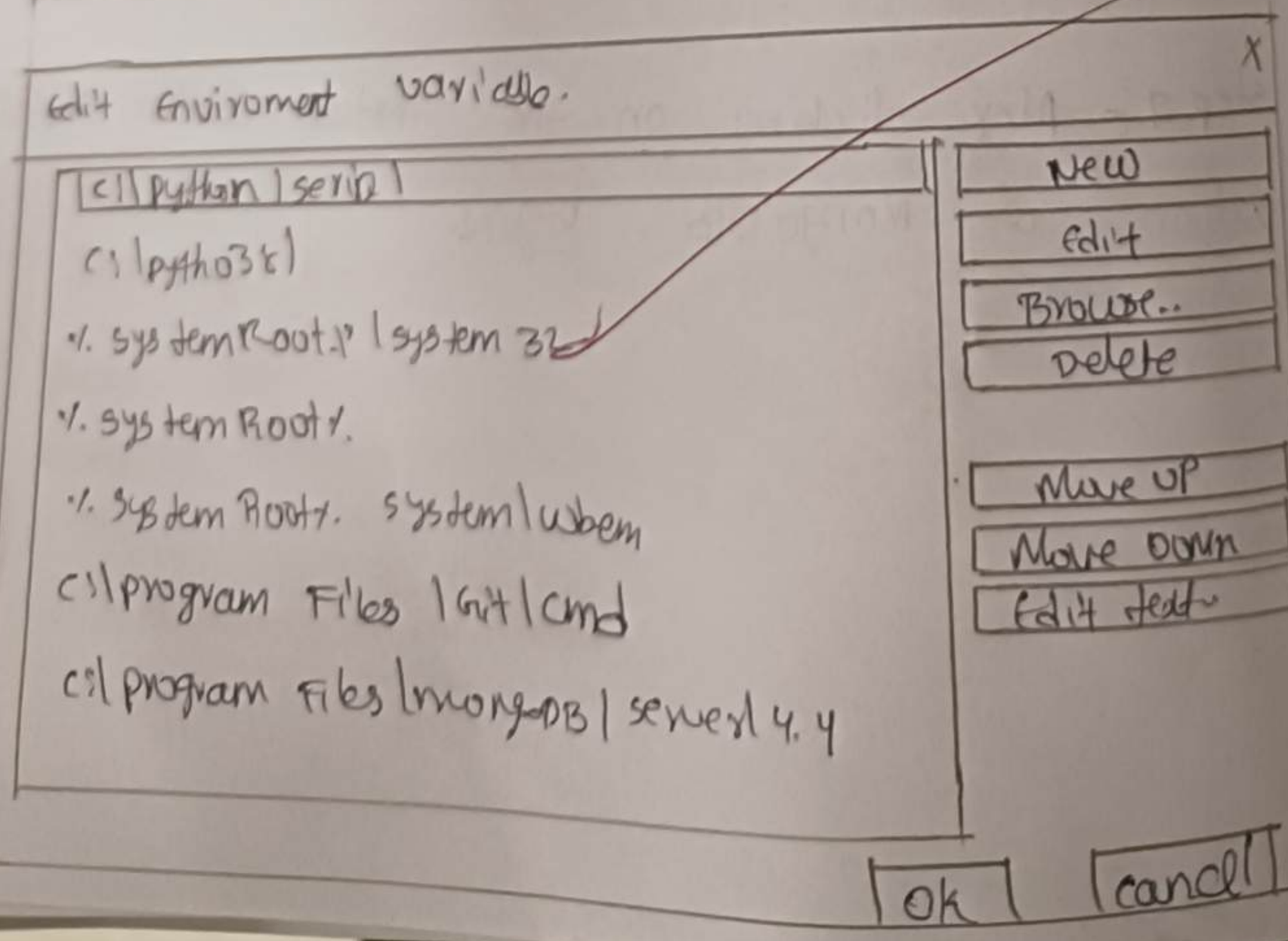
step 8:- Now click the finish button to complete the installation process

step 9:- No we go to the location where Mongo DB installed in your system and copy the path upto bin.

step 10:- now, to create an enviroment variable open system properties. go to system variable and then go to path, edit environment variable and paste the copied link to your environment system and click ok.







step 8:- Now click the installation

step 9:- Now we DB installed in path upto bi

step 10:- Now open syste and then variable a environment

Experiment -21

Aim:- Exercise on creation and Dropping of database.

Procedure:-

creating of database:-

. create a MongoDB

syntax use databasename

ex:- use hemanth

. see the currently created bb

syntax db

. To show the list of data bases in the system.

syntax:- show dbs.

Dropping of database:-

. list out all the available database by using the show dbs command.

syntax:- show dbs

. select the particular database which want delete, if you already in selected database so we switched to other database. Then write the command as shoco beloco.

ex:- db.dropDatabase()

. check the list of databases again now using the show command to confirm the deletion of the database

show dbs

Experiment -22

Aim:- Exercise on creating and dropping of collections.

Procedure:-

creating a mongodb collection:-

- collection belong to the single database so first we will have to create a database for creating a collection.

step 1:-

First run the following command into your mongo shell or command prompt mongo.

step 2:-

check all the existing databases by using the following command.

step 3:-

creating database.

use HEMANTH

If the database name HEMANTH is present it will switch to it, else it will create a new database name as akhildb1 and then switch to it as shown below.

step 4:-

If a collection does not exist, MongoDB creates the collection when you first store data for that collection.

db. HEMANTH. insertone ({id:13})

step 5:-

you can use db.(reate collection() method for creating of collections.

Ex:- db. (reate collection ("aanmvvrsr")

step 6:-

we can also write collections as shown below

db. (reate Collection (name, options).

Experiment -23

Aim: Exercise on commands on mongodb - insert, update, find, Delete, and sorting of sorting of Documents.

procedure:

1. How to insert a document into a collection.

syntax: db.name of collection.insert({})

ex:

db.customer.insert([

{"name": "Theodore", "gender": "M"},

{"name": "Jane Doe", "gender": "F"},

{"name": "John Doe", "gender": "M"}

])

2. How to show all documents in a collection.

syntax: db.collectionName.~~find~~()

ex:

db.customer.find().pretty()

[

{

"_id": ObjectId("659c294d8e2c164066bbeca9"),

"name": "Theodore",

"gender": "M"

},

{

"_id": ObjectId("659cda1f8e2c164066bbeca"),

"name": "Theodore",

"gender" : "M"

}.

3. How to update a document with $set

syntax db.[name of collection].update({},{$set:{}})

ex: db.customer.update({"name":"dbms"},
{$set:{"name":"rdbms"}})

4. How to remove a document from a collection

syntax: db.[name of collections].remove({})

ex: db.customer.remove({"name":"Theodore"})

db.customer.remove({"name":"Theodore"})

writeResult({"nRowe~~ved~~":2})

5. How to sort the given data in ascending order.

- The sorting order in mongodB is defined by either by a one (1) or a minus (-1). Here the positive one represents the ascending order, while the negative one represents one represents the descending order.

syntax: db.collection-name.find().sort({field-name:
sor order})

ex: db.customer.find().sort({name:1})

IN Ascending order:

db.customer.find().sort({name:1})

{"_id" : objectId ("659f59c82cedadafcef1611"), "name":
"jaggu", "gender" : "m"}

{"_id": objectId ("659fa112c8edabfcef1616"),
"name" : "vasanth", "gender" : "f"}

{"_id": objectId ("6559d92c8edabcef1616"),
"name" : "vishu", "gender" : "m"}

In decsending order:-

db.customer.find().sort({name:-1}).pretty()

{ "_id" : objectId ("65f59d928edaafcef1615"),
"name": "vishu",
"gender": "m"
}

{ "_id" : objectId ("659f5a112c8edadfcef1616"),
"name": "vasantha",
"gender": "f"
}

{ "_id" : objectId ("659f5982cedabfcef1614"),
"name": "jaggu",
"gender": "m"
}