GIT LINK:- https://github.com/nlvmadhav/Drought\_Info.git

**PROBLEM STATEMENT: -**

Though shortage of water triggers drought, it has been classified into various types viz., meteorological, hydrological, agricultural for better understanding, monitoring and management. Drought is characterized by its creeping onset and withdrawal and there can't be any definitive beginning or end of its occurrence. What is considered as a drought like condition in a particular climatic setting may not be considered to have the same impact in another climatic setting. Effective management and mitigation of drought impacts are not easy due to such ambiguities. Drought Management (2016) of the Ministry of Agriculture & Farmers' Welfare, Govt of India has desired that declaration of drought by the State Governments should be supported by the matrix-based analysis based on rainfall indices, hydrological indices, vegetative indices, progression of crop sowing and other parameters. Further, National Crisis Management Plan for Drought (2019) suggested a framework of crisis analysis with vulnerability magnitude, on a scale of 0-10, as area specific measure with appropriate strategic response matrix. With the available information and forecast by agencies on rainfall, soil moisture, surface and groundwater levels, color coded areas may be generated using google maps for display of vulnerability magnitude for the whole country at district/sub-district level. It may also be designed as easily accessible stand-alone App which can generate location specific notifications to the common user. It will aid in better monitoring and management of portents of drought by inter-ministerial Crop Weather Watch Group (CWWG) and State Drought Monitoring Centers (DMCs) apart from disseminating information to the general public.

**Reference websites**: -

1. Ref:http://agricoop.nic.in/sites/default/files/Manual%20Drought%202016.pdf
2. Ref: [http://agricoop.nic.in/sites/default/files/CRISISMANAGEMENT- PLAN 2019.pdf](http://agricoop.nic.in/sites/default/files/CRISISMANAGEMENT-%20PLAN%202019.pdf)

**Abstract: -**

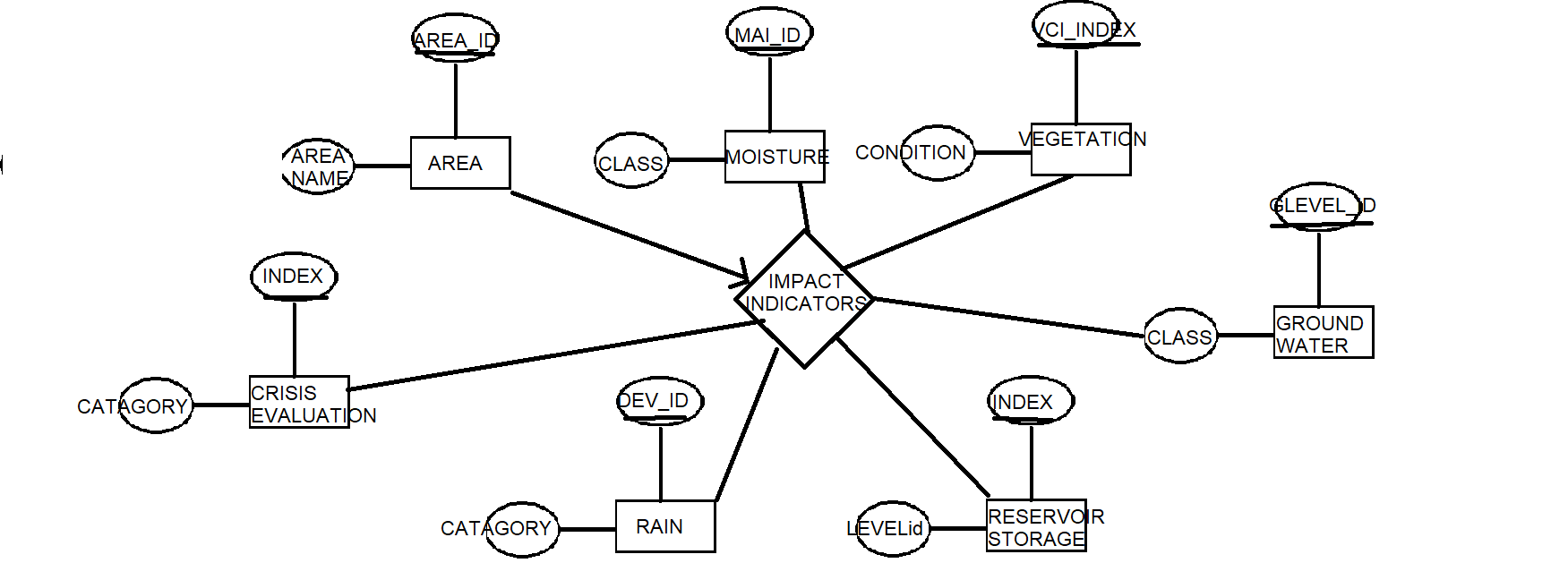
Drought is **a prolonged dry period in the natural climate cycle.** They have the potential to cause severe economic and social impacts. In recent years, with the increasing frequency and severity of droughts India, the need for effective drought information and management is very much need of the day. Through this project we can effectively develop an integrated approach to drought management that combines the data like rain patterns, vegetation, reservoir levels, ground water levels, and provide the category of the drought in that area.

**Number of tables: -**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name** | **Number of columns** | **Primary key** | **Foreign key** |
| Rain | 2 | Dev\_id | - |
| Vegetation | 2 | VCI\_id | - |
| Moisture | 2 | MAI\_id | - |
| Reservoir\_Storage\_Index | 2 | Storage\_level\_id | - |
| Groundwater | 2 | Groundwater\_level\_id | - |
| Area | 2 | Area\_id | - |
| Crisis\_evaluation\_index | 2 | Category\_of\_drought | - |
| Matrix\_impact\_indicators | 6 | Area\_id | primary keys of tables mentioned above |
| Area\_rainfall | 3 | Area\_id,Month | Area\_id |
| Area\_vegetation | 3 | Area\_id,Month | Area\_id |
| Area\_moisture | 3 | Area\_id,Month | Area\_id |
| Area\_groundwater | 3 | Area\_id,Month | Area\_id |
| Area\_reservoir | 3 | Area\_id,Month | Area\_id |

**ENTITY-REALATIONSHIP DIAGRAM: -**

IMPACT\_INDIACATORS table E-R diagram:



**DDL commands: -**

**DDL command for creation of rain table**

SQL> create table rain(

2 dev\_id number,

3 rainfall\_dev number,

4 category varchar2(20));

**Output: -**

SQL> desc rain

Name Null? Type

----------------------------------------- -------- ----------------------------

DEV\_ID NOT NULL NUMBER

CATEGORY VARCHAR2(20)

**DDL command for creation of vegetation table**

SQL> create table vegetation(

2 vci\_index number,

3 vegetation\_condiation varchar2(20));

Table created.

**Output: -**

**SQL> desc vegetation**

Name Null? Type

----------------------------------------- -------- ----------------------------

VCI\_INDEX NOT NULL NUMBER

VEGETATION\_CONDIATION VARCHAR2(20)

**DDL command for creation of moisture table**

SQL> create table moisture(

2 MAI\_ID number,

3 agriculture\_drought\_class varchar2(30));

Table created.

**Output**

SQL> desc moisture

Name Null? Type

----------------------------------------- -------- ----------------------------

MAI\_ID NOT NULL NUMBER

AGRICULTURE\_DROUGHT\_CLASS VARCHAR2(30)

**DDL command for creation of reservoir\_storage\_index table**

SQL> create table Reservoir\_Storage\_Index(

2 storage\_level\_id number,

3 category varchar2(30));

Table created.

**Output**

SQL> desc reservoir\_storage\_index

Name Null? Type

----------------------------------------- -------- ----------------------------

STORAGE\_LEVEL\_ID NOT NULL NUMBER

CATEGORY VARCHAR2(30)

**DDL command for creation of groundwater table**

SQL> create table groundwater(

2 Groundwater\_level\_id number,

3 groundwater\_deficit\_class varchar2(30));

Table created.

**Output**

**SQL> desc groundwater**

Name Null? Type

----------------------------------------- -------- ----------------------------

GROUNDWATER\_LEVEL\_ID NOT NULL NUMBER

GROUNDWATER\_DEFICIT\_CLASS VARCHAR2(30)

**DDL command for creation of area table**

SQL> create table area(

2 area\_Id number,

3 area\_Name varchar2(30));

Table created.

**Output**

SQL> desc area

Name Null? Type

----------------------------------------- -------- ----------------------------

AREA\_ID NOT NULL NUMBER

AREA\_NAME VARCHAR2(30)

**DDL command for creation of crisis\_evaluation\_index table**

SQL> create table crisis\_evaluation\_index(

2 category\_of\_drought number,

3 phase\_of\_crisis varchar2(40));

Table created.

**Output**

SQL> desc crisis\_evaluation\_index

Name Null? Type

----------------------------------------- -------- ----------------------------

CATEGORY\_OF\_DROUGHT NOT NULL NUMBER

PHASE\_OF\_CRISIS VARCHAR2(40)

**DDL command for creation of matrix\_impact\_indiacatiors**

SQL> create table matrix\_impact\_indicators(

2 area\_Id number,

3 dev\_id number,

4 vci\_index number,

5 storage\_level\_id number,

6 groundwater\_level\_id number,

7 category\_of\_drought number);

Table created.

**Output**

SQL> desc matrix\_impact\_indicators

Name Null? Type

----------------------------------------- -------- ----------------------------

AREA\_ID NOT NULL NUMBER

DEV\_ID NUMBER

VCI\_INDEX NUMBER

STORAGE\_LEVEL\_ID NUMBER

GROUNDWATER\_LEVEL\_ID NUMBER

CATEGORY\_OF\_DROUGHT NUMBER

**DDL command for creation of area\_rainfall**

SQL> create table area\_rainfall(

2 area\_id number,

3 rainfall number,

4 month varchar2(20));

Table created.

**Output**

SQL> desc area\_rainfall

Name Null? Type

----------------------------------------- -------- ----------------------------

AREA\_ID NUMBER

RAINFALL NUMBER

MONTH VARCHAR2(20)

**DDL command for creation of area\_vegetation**

SQL> create table area\_vegetation(

2 area\_id number,

3 vci\_percentage number,

4 month varchar2(20),

Tablecreated.

**Output**

SQL> desc area\_vegetation

Name Null? Type

----------------------------------------- -------- ----------------------------

AREA\_ID NOT NULL NUMBER

VCI\_PERCENTAGE NUMBER

MONTH NOT NULL VARCHAR2(20)

**DDL command for creation of area\_moisture**

SQL> create table area\_moisture(

2 area\_id number,

3 percent\_availabel\_soil\_moisture number,

4 month varchar2(20),

Table created.

**Output**

SQL> desc area\_moisture

Name Null? Type

----------------------------------------- -------- ------------------------------------------

AREA\_ID NOT NULL NUMBER

PERCENT\_AVAILABEL\_SOIL\_MOISTURE NUMBER

MONTH NOT NULL VARCHAR2(20)

**DDL command to create area\_reservoir\_storage table**

SQL> create table area\_reservoir\_storage(

2 area\_id number,

3 percent\_deficit number,

4 month varchar2(20));

Table created.

**Output**

SQL> desc area\_reservoir\_storage;

Name Null? Type

----------------------------------------- -------- ----------------------------

AREA\_ID NOT NULL NUMBER

PERCENT\_DEFICIT NUMBER

MONTH NOT NULL VARCHAR2(20)

**DDL command to create area\_groundwater table**

SQL> create table area\_groundwater(

2 area\_id number,

3 groundwater\_drought\_index number,

4 month varchar2(20),

Table created.

**Output**

SQL> desc area\_groundwater

Name Null? Type

----------------------------------------- -------- ----------------------------

AREA\_ID NOT NULL NUMBER

GROUNDWATER\_DROUGHT\_INDEX NUMBER

MONTH NOT NULL VARCHAR2(20)

**DDL command for establishing constraints on rain table**

SQL> alter table rain add primary key(dev\_id);

Table altered.

**DDL command for establishing constraints on vegetation table**

SQL> alter table vegetation add primary key(vci\_index);

Table altered.

**DDL command for establishing constraints on moisture table**

SQL> alter table moisture add primary key(mai\_id);

Table altered.

**DDL command for establishing constraints on reservoir\_storage\_index table**

SQL> alter table Reservoir\_Storage\_Index add primary key(storage\_level\_id);

Table altered.

**DDL command for establishing constraints on groundwater table**

SQL> alter table groundwater add primary key(Groundwater\_level\_id);

Table altered.

**DDL command for establishing constraints on area table**

SQL> alter table area add primary key(area\_id);

Table altered.

**DDL command for establishing constraints on crisis\_evaluation\_index table**

SQL> alter table crisis\_evaluation\_index add primary key(category\_of\_drought);

Table altered.

**DDL command for establishing constraints on matrix\_impact\_indicator table**

SQL> alter table matrix\_impact\_indicators add foreign key(area\_Id) references area on delete cascade;

Table altered.

SQL> alter table matrix\_impact\_indicators add primary key(area\_id);

Table altered.

SQL> alter table matrix\_impact\_indicators add foreign key(dev\_id) references rain on delete cascade;

Table altered.

SQL> alter table matrix\_impact\_indicators add foreign key(vci\_index) references vegetation on delete cascade;

Table altered.

SQL> alter table matrix\_impact\_indicators add foreign key(storage\_level\_id) references reservoir\_storage\_index on delete cascade;

Table altered.

SQL> alter table matrix\_impact\_indicators add foreign key( groundwater\_level\_id) references groundwater on delete cascade;

Table altered.

SQL> alter table matrix\_impact\_indicators add foreign key(category\_of\_drought) references crisis\_evaluation\_index on delete cascade;

Table altered.

**DDL command for establishing constraints on area\_rainfall table**

SQL> alter table area\_rainfall add primary key(area\_id,month);

Table altered.

SQL> alter table area\_rainfall add foreign key(area\_id) references area;

Table altered.

**DDL command for establishing constraints on area\_vegetation table**

SQL> alter table area\_vegetation add primary key(area\_id,month);

Table altered.

SQL> alter table area\_vegetation add foreign key(area\_id) references area;

Table altered.

**DDL command for establishing constraints on area\_moisture table**

SQL> alter table area\_moisture add primary key(area\_id,month);

Table altered.

SQL> alter table area\_moisture add foreign key(area\_id) references area;

Table altered.

**DDL command for establishing constraints on area\_reservoir\_storage\_index table**

SQL> alter table area\_reservoir\_storage add primary key(area\_id,month);

Table altered.

SQL> alter table area\_reservoir\_storage add foreign key(area\_id) references area;

Table altered.

**DDL command for establishing constraints on area\_groundwater\_index table**

SQL> alter table area\_groundwater add primary key(area\_id,month);

Table altered.

SQL> alter table area\_groundwater add foreign key(area\_id) references area;

Table altered.

**DML commands: -**

**DML command to insert values into rain table**

SQL> insert into rain values(&dev\_id,'&category');

Enter value for dev\_id: 0

Enter value for category: normal

old 1: insert into rain values(&dev\_id,'&category')

new 1: insert into rain values(0,'normal')

1 row created.

SQL> /

Enter value for dev\_id: 1

Enter value for category: deficient

old 1: insert into rain values(&dev\_id,'&category')

new 1: insert into rain values(1,'deficient')

1 row created.

SQL> /

Enter value for dev\_id: 2

Enter value for category: large deficient

old 1: insert into rain values(&dev\_id,'&category')

new 1: insert into rain values(2,'large deficient')

1 row created.

SQL> /

Enter value for dev\_id: 3

Enter value for category: no rain

old 1: insert into rain values(&dev\_id,'&category')

new 1: insert into rain values(3,'no rain')

1 row created.

**Output: -**

SQL> select \* from rain;

DEV\_ID CATEGORY

---------- --------------------

0 normal

1 deficient

2 large deficient

3 no rain

**DML command to insert values into vegetation table**

SQL> insert into vegetation values(&vci\_index,'&vegetation\_condition');

Enter value for vci\_index: 0

Enter value for vegetation\_condition: good

old 1: insert into vegetation values(&vci\_index,'&vegetation\_condition')

new 1: insert into vegetation values(0,'good')

1 row created.

SQL> /

Enter value for vci\_index: 1

Enter value for vegetation\_condition: fair

old 1: insert into vegetation values(&vci\_index,'&vegetation\_condition')

new 1: insert into vegetation values(1,'fair')

1 row created.

SQL> /

Enter value for vci\_index: 2

Enter value for vegetation\_condition: poor

old 1: insert into vegetation values(&vci\_index,'&vegetation\_condition')

new 1: insert into vegetation values(2,'poor')

1 row created.

**Output**

SQL> select \* from vegetation;

VCI\_INDEX VEGETATION\_CONDIATIO

--------------- -------------------------

0 good

1 fair

2 poor

**DML command to insert values into moisture table**

SQL> insert into moisture

values(&mai\_id,'&agriculture\_dorught\_class');

Enter value for mai\_id: 0

Enter value for agriculture\_dorught\_class: no drought

old 1: insert into moisture

values(&mai\_id,'&agriculture\_dorught\_class')

new 1: insert into moisture values(0,'no drought')

1 row created.

SQL> /

Enter value for mai\_id: 1

Enter value for agriculture\_dorught\_class: mild drought

old 1:

insert into moisture values(&mai\_id,'&agriculture\_dorught\_class')

new 1: insert into moisture values(1,'mild drought')

1 row created.

SQL> /

Enter value for mai\_id: 2

Enter value for agriculture\_dorught\_class: moderate drought

old 1: insert into moisture

values(&mai\_id,'&agriculture\_dorught\_class')

new 1: insert into moisture values(2,'moderate drought')

1 row created.

SQL> /

Enter value for mai\_id: 3

Enter value for agriculture\_dorught\_class: severe drought

old 1: insert into moisture

values(&mai\_id,'&agriculture\_dorught\_class')

new 1: insert into moisture values(3,'severe drought')

1 row created.

**Output**

SQL> select \* from moisture;

MAI\_ID AGRICULTURE\_DROUGHT\_CLASS

-------------- -------------------------------------------

0 no drought

1 mild drought

2 moderate drought

3 severe drought

**DML command to insert values into reservoir\_storeage\_index table**

SQL> insert into reservoir\_storage\_index

values(&storage\_level\_id,'&catagory');

Enter value for storage\_level\_id: 0

Enter value for catagory: normal deficit

old 1: insert into reservoir\_storage\_index

values(&storage\_level\_id,'&catagory')

new 1: insert into reservoir\_storage\_index values(0,'normal deficit')

1 row created.

SQL> /

Enter value for storage\_level\_id: 1

Enter value for catagory: mild deficit

old 1: insert into reservoir\_storage\_index

values(&storage\_level\_id,'&catagory')

new 1: insert into reservoir\_storage\_index values(1,'mild deficit')

1 row created.

SQL> /

Enter value for storage\_level\_id: 2

Enter value for catagory: moderate deficit

old 1: insert into reservoir\_storage\_index

values(&storage\_level\_id,'&catagory')

new 1: insert into reservoir\_storage\_index values(2,'moderate deficit')

1 row created.

SQL> /

Enter value for storage\_level\_id: 3

Enter value for catagory: severe deficit

old 1: insert into reservoir\_storage\_index

values(&storage\_level\_id,'&catagory')

new 1: insert into reservoir\_storage\_index values(3,'severe deficit')

1 row created.

SQL> /

Enter value for storage\_level\_id: 4

Enter value for catagory: extreme deficit

old 1: insert into reservoir\_storage\_index

values(&storage\_level\_id,'&catagory')

new 1: insert into reservoir\_storage\_index values(4,'extreme deficit')

1 row created.

**Output**

SQL> select \* from reservoir\_storage\_index;

STORAGE\_LEVEL\_ID CATEGORY

-------------------------- --------------------

0 normal deficit

1 mild deficit

2 moderate deficit

3 severe deficit

4 extreme deficit

**DML command to insert values into groundwater table**

SQL> insert into groundwater

values(&groundwater\_level\_id,'&Groundwater\_deficit\_class');

Enter value for groundwater\_level\_id: 0

Enter value for groundwater\_deficit\_class: normal

old 1: insert into groundwater

values(&groundwater\_level\_id,'&Groundwater\_deficit\_class')

new 1: insert into groundwater values(0,'normal')

1 row created.

SQL> /

Enter value for groundwater\_level\_id: 1

Enter value for groundwater\_deficit\_class: mild

old 1: insert into groundwater

values(&groundwater\_level\_id,'&Groundwater\_deficit\_class')

new 1: insert into groundwater values(1,'mild')

1 row created.

SQL> /

Enter value for groundwater\_level\_id: 2

Enter value for groundwater\_deficit\_class: moderate

old 1: insert into groundwater

values(&groundwater\_level\_id,'&Groundwater\_deficit\_class')

new 1: insert into groundwater values(2,'moderate')

1 row created.

SQL> /

Enter value for groundwater\_level\_id: 3

Enter value for groundwater\_deficit\_class: severe

old 1: insert into groundwater

values(&groundwater\_level\_id,'&Groundwater\_deficit\_class')

new 1: insert into groundwater values(3,'severe')

1 row created.

SQL> /

Enter value for groundwater\_level\_id: 4

Enter value for groundwater\_deficit\_class: extreme

old 1: insert into groundwater

values(&groundwater\_level\_id,'&Groundwater\_deficit\_class')

new 1: insert into groundwater values(4,'extreme')

1 row created.

**Output**

SQL> select\* from groundwater;

GROUNDWATER\_LEVEL\_ID GROUNDWATER\_DEFICIT\_CLASS

----------------------------------- -------------------------------------------

0 normal

1 mild

2 moderate

3 severe

4 extreme

**DML command to insert values into crisis\_evaluation\_index table**

SQL> insert into crisis\_evaluation\_index

values(&category\_of\_drought,'&phase\_of\_crisis');

Enter value for category\_of\_drought: 0

Enter value for phase\_of\_crisis: normal

old 1: insert into crisis\_evaluation\_index

values(&category\_of\_drought,'&phase\_of\_crisis')

new 1: insert into crisis\_evaluation\_index values(0,'normal')

1 row created.

SQL> /

Enter value for category\_of\_drought: 1

Enter value for phase\_of\_crisis: alert/watch

old 1: insert into crisis\_evaluation\_index

values(&category\_of\_drought,'&phase\_of\_crisis')

new 1: insert into crisis\_evaluation\_index values(1,'alert/watch')

1 row created.

SQL> /

Enter value for category\_of\_drought: 2

Enter value for phase\_of\_crisis: warning

old 1: insert into crisis\_evaluation\_index

values(&category\_of\_drought,'&phase\_of\_crisis')

new 1: insert into crisis\_evaluation\_index values(2,'warning')

1 row created.

SQL> /

Enter value for category\_of\_drought: 3

Enter value for phase\_of\_crisis: emergency

old 1: insert into crisis\_evaluation\_index

values(&category\_of\_drought,'&phase\_of\_crisis')

new 1: insert into crisis\_evaluation\_index values(3,'emergency')

1 row created.

SQL> /

Enter value for category\_of\_drought: 4

Enter value for phase\_of\_crisis: extreme

old 1: insert into crisis\_evaluation\_index

values(&category\_of\_drought,'&phase\_of\_crisis')

new 1: insert into crisis\_evaluation\_index values(4,'extreme')

1 row created.

SQL> /

Enter value for category\_of\_drought: 5

Enter value for phase\_of\_crisis: mitigated

old 1: insert into crisis\_evaluation\_index

values(&category\_of\_drought,'&phase\_of\_crisis')

new 1: insert into crisis\_evaluation\_index values(5,'mitigated')

1 row created.

**Output**

SQL> select \* from crisis\_evaluation\_index;

CATEGORY\_OF\_DROUGHT PHASE\_OF\_CRISIS

---------------------------------- -------------------------

0 normal

1 alert/watch

2 warning

3 emergency

4 extreme

5 mitigated

6 rows selected.

**DML command to insert values into area table**

SQL> insert into area values(&area\_id,'&area\_name');

Enter value for area\_id: 1

Enter value for area\_name: Tamil Nadu

old 1: insert into area values(&area\_id,'&area\_name')

new 1: insert into area values(1,'Tamil Nadu')

1 row created.

SQL> insert into area values(&area\_id,'&area\_name');

Enter value for area\_id: 2

Enter value for area\_name: Andhra pradesh

old 1: insert into area values(&area\_id,'&area\_name')

new 1: insert into area values(2,'Andhra pradesh')

1 row created.

SQL> /

Enter value for area\_id: 3

Enter value for area\_name: Telangana

old 1: insert into area values(&area\_id,'&area\_name')

new 1: insert into area values(3,'Telangana')

1 row created.

SQL> /

Enter value for area\_id: 4

Enter value for area\_name: Karnataka

old 1: insert into area values(&area\_id,'&area\_name')

new 1: insert into area values(4,'Karnataka')

1 row created.

SQL> /

Enter value for area\_id: 5

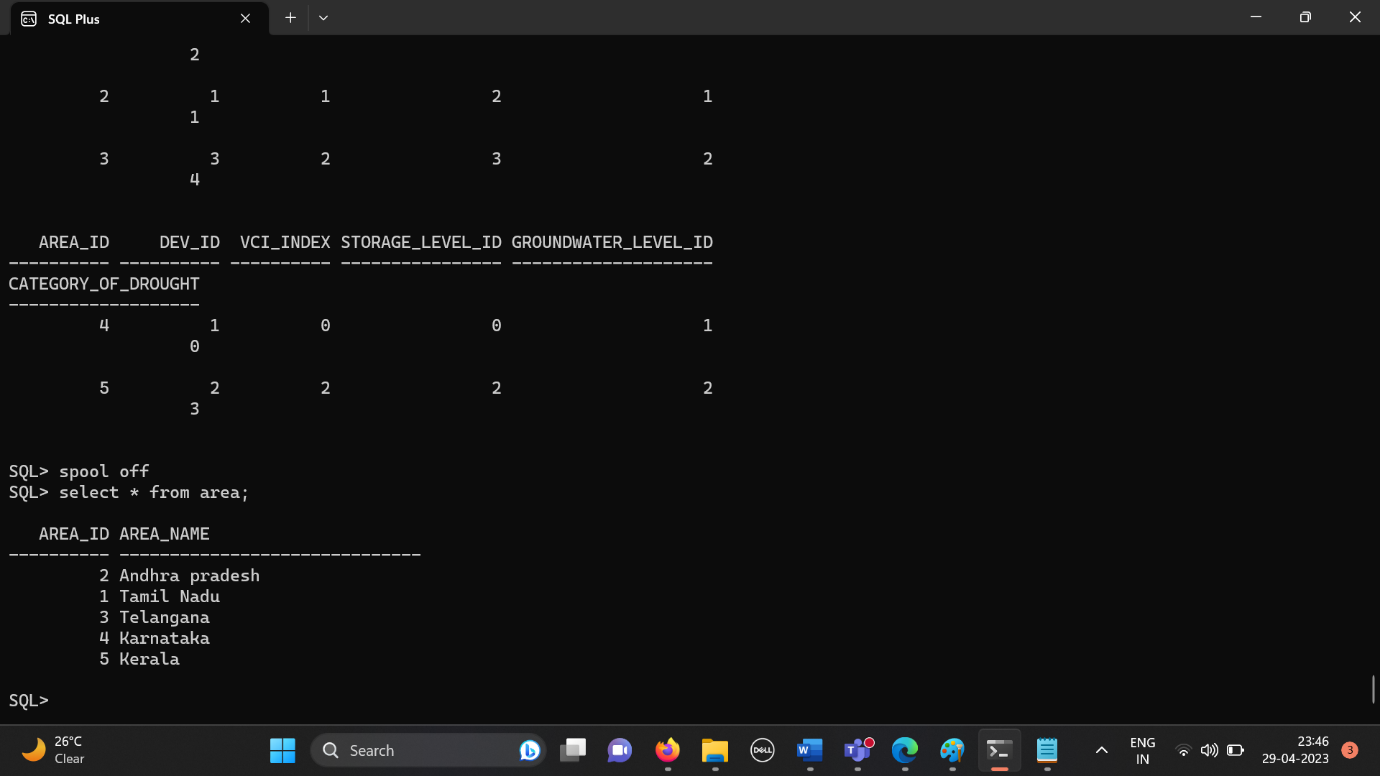
Enter value for area\_name: Kerala

old 1: insert into area values(&area\_id,'&area\_name')

new 1: insert into area values(5,'Kerala')

1 row created.

**Output**



**DML command to insert values into matrix\_impact\_indicators table**

SQL> insert into matrix\_impact\_indicators

values(&area\_id,&dev\_id,&vci\_index,&storage\_level\_id,&ground\_level\_id,&category\_of\_drought);

Enter value for area\_id: 1

Enter value for dev\_id: 2

Enter value for vci\_index: 2

Enter value for storage\_level\_id: 3

Enter value for ground\_level\_id: 2

Enter value for category\_of\_drought: 2

old 1: insert into matrix\_impact\_indicators

values(&area\_id,&dev\_id,&vci\_index,&storage\_level\_id,&ground\_level\_id,&category\_of\_drought)

new 1: insert into matrix\_impact\_indicators values(1,2,2,3,2,2)

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for dev\_id: 1

Enter value for vci\_index: 1

Enter value for storage\_level\_id: 2

Enter value for ground\_level\_id: 1

Enter value for category\_of\_drought: 1

old 1: insert into matrix\_impact\_indicators

values(&area\_id,&dev\_id,&vci\_index,&storage\_level\_id,&ground\_level\_id,&category\_of\_drought)

new 1: insert into matrix\_impact\_indicators values(2,1,1,2,1,1)

1 row created.

SQL> /

Enter value for area\_id: 3

Enter value for dev\_id: 3

Enter value for vci\_index: 2

Enter value for storage\_level\_id: 3

Enter value for ground\_level\_id: 2

Enter value for category\_of\_drought: 4

old 1: insert into matrix\_impact\_indicators

values(&area\_id,&dev\_id,&vci\_index,&storage\_level\_id,&ground\_level\_id,&category\_of\_drought)

new 1: insert into matrix\_impact\_indicators values(3,3,2,3,2,4)

1 row created.

SQL> insert into matrix\_impact\_indicators values(&area\_id,&dev\_id,&vci\_index,&storage\_level\_id,&ground\_level\_id,&category\_of\_drought);

Enter value for area\_id: 4

Enter value for dev\_id: 1

Enter value for vci\_index: 0

Enter value for storage\_level\_id: 0

Enter value for ground\_level\_id: 1

Enter value for category\_of\_drought: 0

old 1: insert into matrix\_impact\_indicators

values(&area\_id,&dev\_id,&vci\_index,&storage\_level\_id,&ground\_level\_id,&category\_of\_drought)

new 1: insert into matrix\_impact\_indicators values(4,1,0,0,1,0)

1 row created.

SQL> /

Enter value for area\_id: 5

Enter value for dev\_id: 2

Enter value for vci\_index: 2

Enter value for storage\_level\_id: 2

Enter value for ground\_level\_id: 2

Enter value for category\_of\_drought: 3

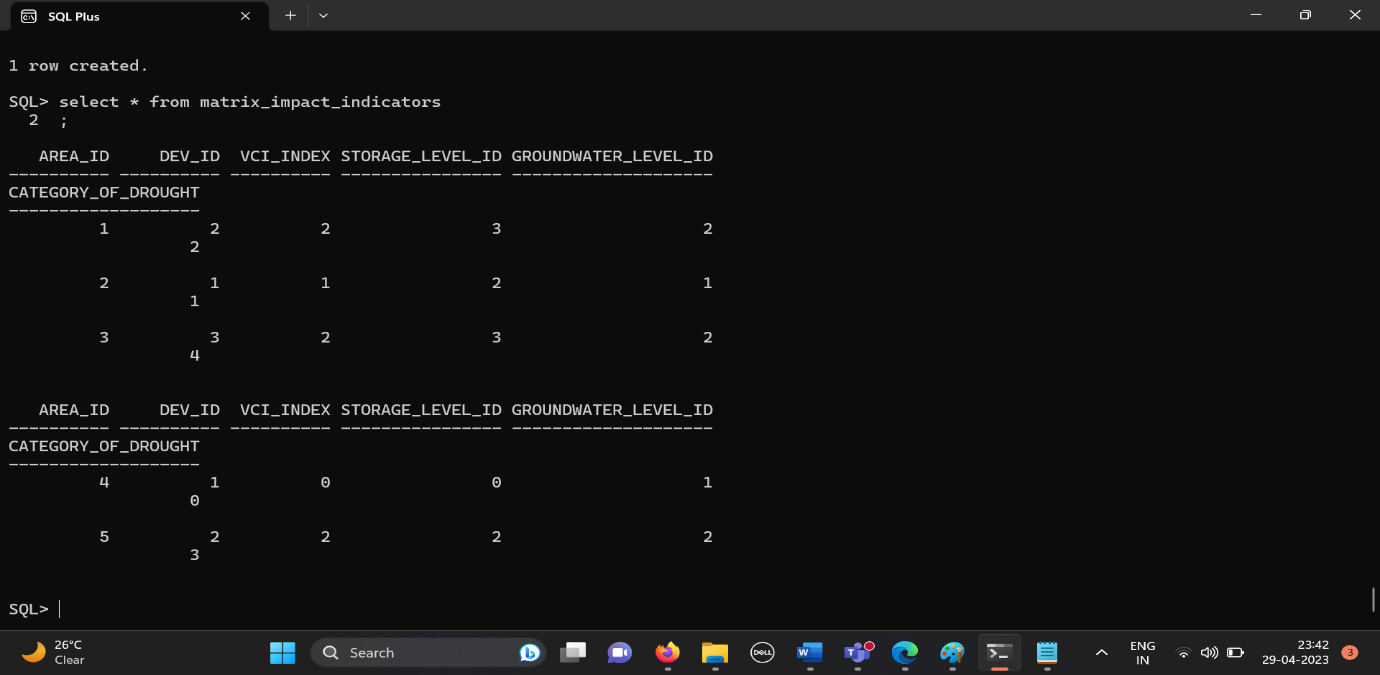
old 1: insert into matrix\_impact\_indicators

values(&area\_id,&dev\_id,&vci\_index,&storage\_level\_id,&ground\_level\_id,&category\_of\_drought)

new 1: insert into matrix\_impact\_indicators values(5,2,2,2,2,3)

1 row created.

**Output**



**DML command to insert values into area\_rainfall table**

SQL> insert into area\_rainfall values(&area\_id,&rainfall,'&month');

Enter value for area\_id: 1

Enter value for rainfall: 20

Enter value for month: jan

old 1: insert into area\_rainfall values(&area\_id,&rainfall,'&month')

new 1: insert into area\_rainfall values(1,20,'jan')

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for rainfall: 30

Enter value for month: jan

old 1: insert into area\_rainfall values(&area\_id,&rainfall,'&month')

new 1: insert into area\_rainfall values(2,30,'jan')

1 row created.

SQL> /

Enter value for area\_id: 1

Enter value for rainfall: 70

Enter value for month: mar

old 1: insert into area\_rainfall values(&area\_id,&rainfall,'&month')

new 1: insert into area\_rainfall values(1,70,'mar')

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for rainfall: 20

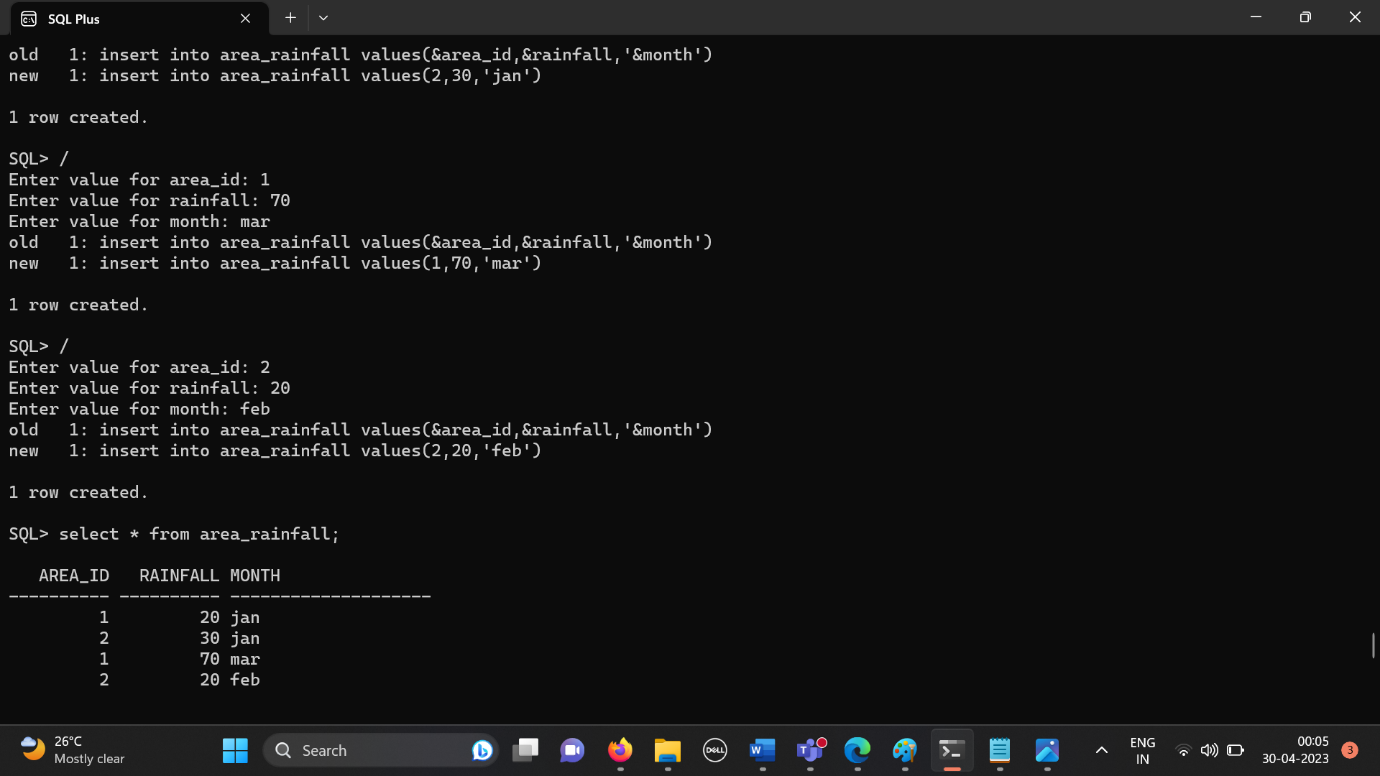
Enter value for month: feb

old 1: insert into area\_rainfall values(&area\_id,&rainfall,'&month')

new 1: insert into area\_rainfall values(2,20,'feb')

1 row created.

**Output: -**



**DML command to insert values into area\_vegetation table**

SQL> insert into area\_vegetation values(&area\_id,&vci\_percentage,'&month');

Enter value for area\_id: 1

Enter value for vci\_percentage: 20

Enter value for month: jan

old 1: insert into area\_vegetation

values(&area\_id,&vci\_percentage,'&month')

new 1: insert into area\_vegetation values(1,20,'jan')

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for vci\_percentage: 30

Enter value for month: jan

old 1: insert into area\_vegetation

values(&area\_id,&vci\_percentage,'&month')

new 1: insert into area\_vegetation values(2,30,'jan')

1 row created.

SQL> /

Enter value for area\_id: 1

Enter value for vci\_percentage: 40

Enter value for month: mar

old 1: insert into area\_vegetation

values(&area\_id,&vci\_percentage,'&month')

new 1: insert into area\_vegetation values(1,40,'mar')

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for vci\_percentage: 30

Enter value for month: feb

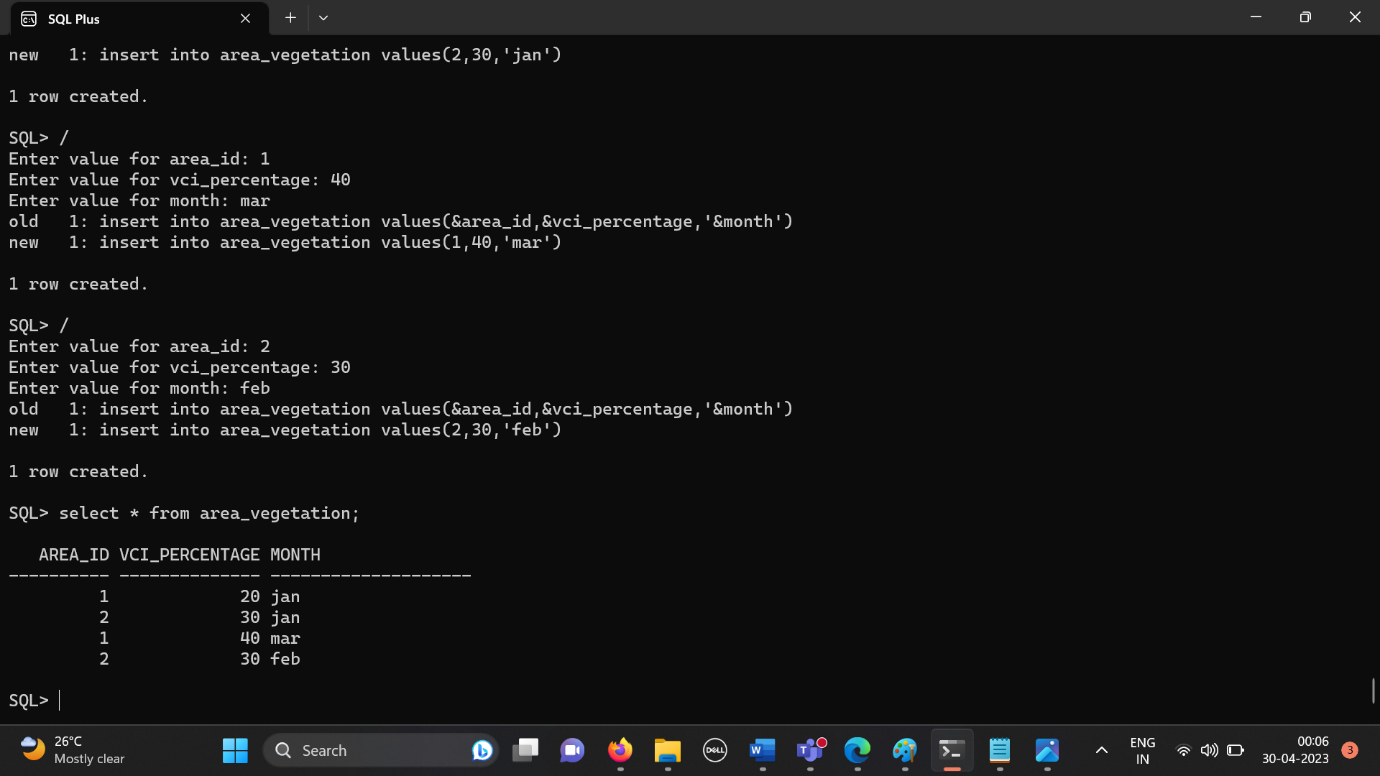
old 1: insert into area\_vegetation

values(&area\_id,&vci\_percentage,'&month')

new 1: insert into area\_vegetation values(2,30,'feb')

1 row created.

**Output**



**DML command to insert values into area\_moisture table**

SQL> insert into area\_moisture

values(&area\_id,&percent\_availabel\_soil\_moisture,'&month');

Enter value for area\_id: 1

Enter value for percent\_availabel\_soil\_moisture: 30

Enter value for month: jan

old 1: insert into area\_moisture

values(&area\_id,&percent\_availabel\_soil\_moisture,'&month')

new 1: insert into area\_moisture values(1,30,'jan')

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for percent\_availabel\_soil\_moisture: 40

Enter value for month: jan

old 1: insert into area\_moisture

values(&area\_id,&percent\_availabel\_soil\_moisture,'&month')

new 1: insert into area\_moisture values(2,40,'jan')

1 row created.

SQL> /

Enter value for area\_id: 1

Enter value for percent\_availabel\_soil\_moisture: 60

Enter value for month: mar

old 1: insert into area\_moisture

values(&area\_id,&percent\_availabel\_soil\_moisture,'&month')

new 1: insert into area\_moisture values(1,60,'mar')

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for percent\_availabel\_soil\_moisture: 50

Enter value for month: feb

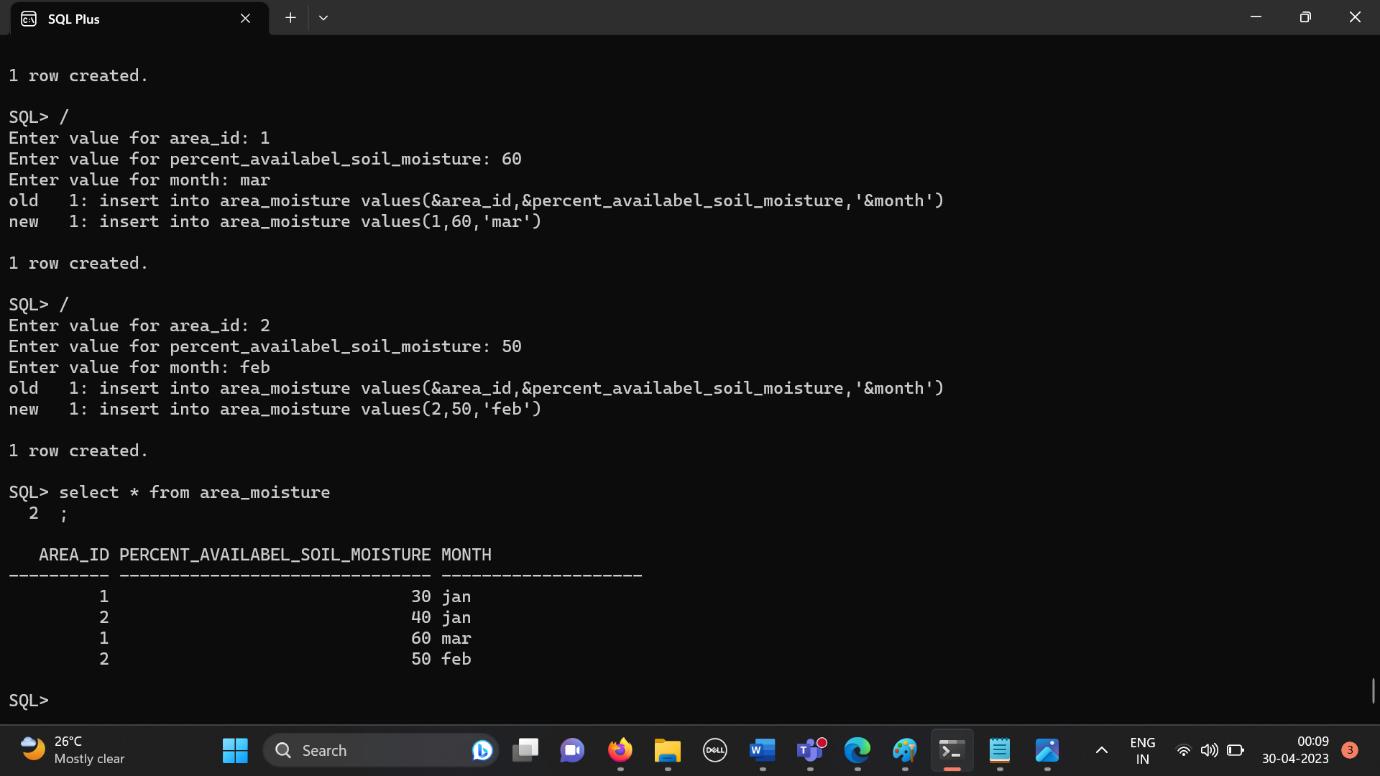
old 1: insert into area\_moisture

values(&area\_id,&percent\_availabel\_soil\_moisture,'&month')

new 1: insert into area\_moisture values(2,50,'feb')

1 row created.

**Output**



**DML command to insert values into area\_** **reservoir\_storage table**

SQL> insert into area\_reservoir\_storage

values(&area\_id,&percent\_deficit,'&month');

Enter value for area\_id: 1

Enter value for percent\_deficit: 40

Enter value for month: jan

old 1: insert into area\_reservoir\_storage

values(&area\_id,&percent\_deficit,'&month')

new 1: insert into area\_reservoir\_storage values(1,40,'jan')

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for percent\_deficit: 20

Enter value for month: jan

old 1: insert into area\_reservoir\_storage

values(&area\_id,&percent\_deficit,'&month')

new 1: insert into area\_reservoir\_storage values(2,20,'jan')

1 row created.

SQL> /

Enter value for area\_id: 1

Enter value for percent\_deficit: 10

Enter value for month: mar

old 1: insert into area\_reservoir\_storage

values(&area\_id,&percent\_deficit,'&month')

new 1: insert into area\_reservoir\_storage values(1,10,'mar')

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for percent\_deficit: 15

Enter value for month: feb

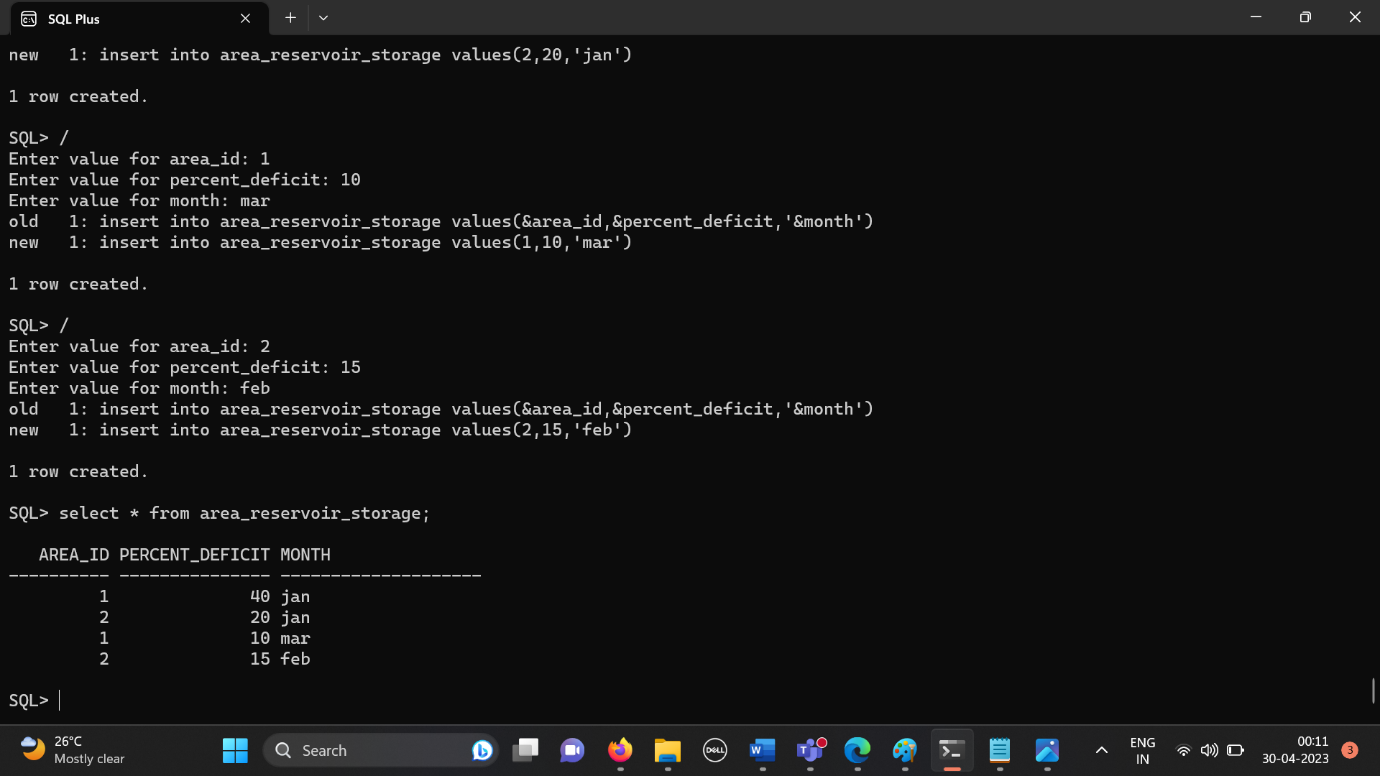
old 1: insert into area\_reservoir\_storage

values(&area\_id,&percent\_deficit,'&month')

new 1: insert into area\_reservoir\_storage values(2,15,'feb')

1 row created.

**Output**



**DML command to insert values into area\_** **groundwater table**

SQL> insert into area\_groundwater

values(&area\_id,&groundwater\_drought\_index,'&month');

Enter value for area\_id: 1

Enter value for groundwater\_drought\_index: 70

Enter value for month: jan

old 1: insert into area\_groundwater

values(&area\_id,&groundwater\_drought\_index,'&month')

new 1: insert into area\_groundwater values(1,70,'jan')

1 row created.

SQL> /

Enter value for area\_id: 2

Enter value for groundwater\_drought\_index: 39

Enter value for month: feb

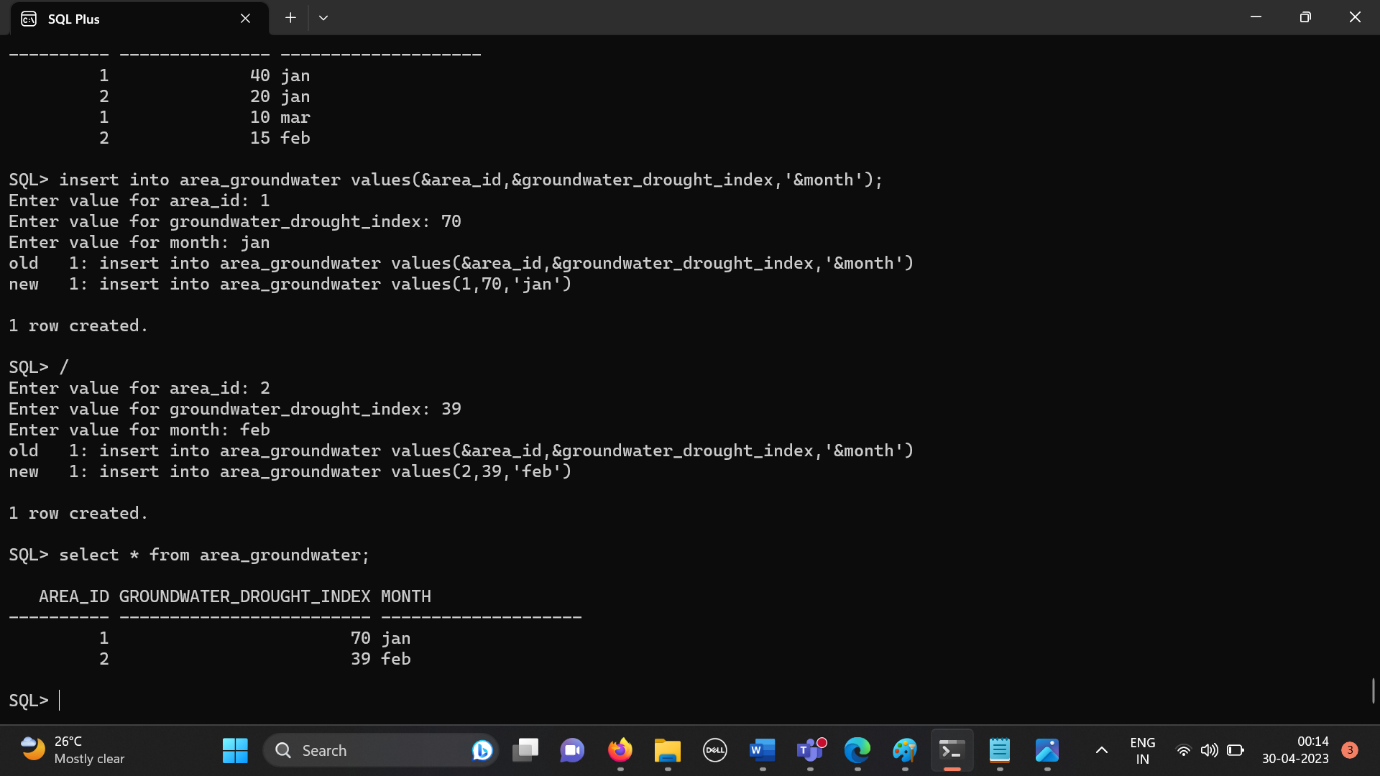
old 1: insert into area\_groundwater

values(&area\_id,&groundwater\_drought\_index,'&month')

new 1: insert into area\_groundwater values(2,39,'feb')

1 row created.

**Output**



**IMPLEMENTATION**

**JAVA-SQL connectivity :-**

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers

* JDBC-ODBC Bridge Driver,
* Native Driver,
* Network Protocol Driver, and
* Thin Driver

For this project I have used OJDBC.8 version java driver. The connection and the implementation of the application code is as follows.

**Connection:-**

 Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","project","proj");

            Statement stmt = con.createStatement();

            String qry = "Select \* from "+*tablename*;

            ResultSet rs = stmt.executeQuery(qry);

            ResultSetMetaData rsm = rs.getMetaData();

**Application code with connection is as follows :-**

import javax.lang.model.util.ElementScanner14;

import javax.swing.\*;

import java.sql.\*;

import java.util.Properties;

import java.awt.event.\*;

import java.awt.\*;

public class frameclass extends JFrame implements ActionListener

{

    //private JFrame mainframe;

    private JScrollPane scrollPane;

    private JPanel mainPanel;

    private JRadioButton ins;

    private JRadioButton upd;

    private JRadioButton del;

    private JRadioButton view;

    private JTextField[] jtf;

    private JComboBox<String> tables;

    private JTextArea textArea;

    private JPanel updatPanel;

    private JPanel insertPanel;

    private JPanel deletPanel;

    private JPanel selectPanel;

    private JScrollPane scrol;

    private *int* i,size;

    private String tablename;

    public frameclass()

    {

        setTitle("Drought det sys");

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLayout(null);

        mainPanel = new JPanel();

        mainPanel.setBounds(0,0,600,500);

        mainPanel.setLayout(null);

        add(mainPanel);

        createMenuBar();

        selectPanel = new JPanel();

        selectPanel.setBounds(0,0,600,300);

        selectPanel.setLayout(null);

        System.out.println("hello1");

        JLabel msgop = new JLabel("Select the Table which you want ");

        msgop.setBounds(50, 50, 300, 25);

        tables = new JComboBox<String>();

        tables.setBounds(50, 80, 200, 25);

        tables.addItem("area");

        tables.addItem("area\_vegetation");

        tables.addItem("area\_moisture");

        tables.addItem("area\_reservoir\_storage");

        tables.addItem("area\_groundwater");

        tables.addItem("area\_rainfall");

        tables.addItem("matrix\_impact\_indicators");

        System.out.println("hello1");

        JLabel oprop = new JLabel("Select the operation you want to perform ");

        oprop.setBounds(50, 120, 300, 25);

        ins = new JRadioButton("INSERT", true);

        ins.setBounds(50, 150, 100, 25);

        upd = new JRadioButton("UPDATE", false);

        upd.setBounds(150, 150, 100, 25);

        del = new JRadioButton("DELETE", false);

        del.setBounds(250, 150, 100, 25);

        view = new JRadioButton("VIEW",false);

        view.setBounds(350,150, 100, 25);

        System.out.println("hello2");

        ButtonGroup bg = new ButtonGroup();

        bg.add(ins);

        bg.add(upd);

        bg.add(del);

        bg.add(view);

        JButton submit = new JButton("SUBMIT");

        submit.setBounds(50, 200, 100, 25);

        submit.addActionListener(this);

        selectPanel.add(msgop);

        selectPanel.add(tables);

        selectPanel.add(oprop);

        selectPanel.add(ins);

        selectPanel.add(upd);

        selectPanel.add(del);

        selectPanel.add(view);

        selectPanel.add(submit);

        insertPanel = new JPanel();

        insertPanel.setBounds(0,0,700,500);

        insertPanel.setLayout(null);

        insertPanel.setVisible(false);

        deletPanel = new JPanel();

        deletPanel.setBounds(0,0,700,500);

        deletPanel.setLayout(null);

        deletPanel.setVisible(false);

        updatPanel = new JPanel();

        updatPanel.setBounds(0,0,700,500);

        updatPanel.setLayout(null);

        updatPanel.setVisible(false);

        textArea = new JTextArea();

        scrollPane = new JScrollPane(textArea);

        scrollPane.setBounds(50, 400, 500, 125);

        mainPanel.add(scrollPane);

        mainPanel.add(selectPanel);

        mainPanel.add(insertPanel);

        mainPanel.add(deletPanel);

        mainPanel.add(updatPanel);

        //selectPanel.setVisible(true);

        //setBounds(100, 100, 1000, 1000);

        //scrol = new JScrollPane(mainPanel);

        //add(scrol);

        setPreferredSize(new Dimension(700,625));

        pack();

        setVisible(true);

    }

    @*Override*

    public *void* actionPerformed(ActionEvent *ae*)

    {

        tablename =(String)tables.getSelectedItem();

        if(ins.isSelected())

        {

            insertvalues(tablename);

        }else if(del.isSelected())

        {

            delete(tablename);

        }else if(upd.isSelected())

        {

            update(tablename);

        }else if(view.isSelected())

        {

            viewtable(tablename);

        }

    }

    public *void* viewtable(String *tablename*)

    {

        textArea.setText("");

        try{

            Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","project","proj");

            Statement stmt = con.createStatement();

            String qry = "Select \* from "+*tablename*;

            ResultSet rs = stmt.executeQuery(qry);

            ResultSetMetaData rsm = rs.getMetaData();

            String s = "";

            for(*int* j = 1;j<= rsm.getColumnCount();j++)

            {

                s+= rsm.getColumnName(j);

                s+= "   ";

            }

            textArea.append(s+"\n");

            s= "";

            while(rs.next())

            {

                for(*int* j = 1;j<=rsm.getColumnCount();j++)

                {

                    s+= rs.getString(rsm.getColumnName(j));

                    s+= "     ";

                }

                textArea.append(s+"\n");

                s = "";

            }

        }catch(Exception ex)

        {

            textArea.append("couldnt display the table");

        }

    }

    private *void* insertvalues(String *tablename*)

    {

        deletPanel.removeAll();

        deletPanel.revalidate();

        deletPanel.repaint();

        updatPanel.removeAll();

        updatPanel.revalidate();

        updatPanel.repaint();

        insertPanel.removeAll();

        insertPanel.revalidate();

        insertPanel.repaint();

        try{

            Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","project","proj");

            Statement stmt = con.createStatement();

            String qry = "Select \* from "+*tablename*;

            ResultSet rs = stmt.executeQuery(qry);

            ResultSetMetaData rsm = rs.getMetaData();

            jtf = new JTextField[rsm.getColumnCount()];

            JLabel[] jl = new JLabel[rsm.getColumnCount()];

            JLabel title = new JLabel(" INSERT ");

            //insertPanel.setBounds(50, 50, 700, 500);

            size = rsm.getColumnCount();

*int* x = 50; // Initial x-coordinate

*int* y = 25; // Initial y-coordinate

*int* labelWidth = 200; // Width of the label

*int* textFieldWidth = 150; // Width of the text field

*int* height = 25; // Height of each component

*int* spacing = 20; // Vertical spacing between components

            //title.setBounds(x,y,labelWidth,height);

            //insertPanel.add(title);

*int* panelHeight = (rsm.getColumnCount() \* (height + spacing)) + 100; // Adjust the value as needed

    insertPanel.setSize(700, panelHeight);

            for(*int* i = 0;i<rsm.getColumnCount();i++)

            {

                jl[i] = new JLabel(rsm.getColumnName(i+1));

                jtf[i] = new JTextField();

                jl[i].setBounds(x,y,labelWidth,height);

                jtf[i].setBounds(x + labelWidth + spacing, y, textFieldWidth, height);

                insertPanel.add(jl[i]);

                insertPanel.add(jtf[i]);

                y += height+spacing;

            }

            JButton sub = new JButton("SUBMIT", null);

            sub.addActionListener(new ActionListener()

            {

                @*Override*

                public *void* actionPerformed(ActionEvent *e*)

                {

                    try{

                        Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","project","proj");

                        Statement stmt = con.createStatement();

                        String qry = "insert into "+*tablename*+" values(";

                        String qry2 = "select calc\_catg\_of\_drought(";

                        //System.out.println(qry);

                        if(size == 2)

                        {

                            qry += jtf[0].getText()+",";

                            qry += "'"+jtf[1].getText()+"'";

                        }else if(size == 3)

                        {

                            qry += jtf[0].getText()+",";

                            qry += jtf[1].getText()+",";

                            qry += "'"+jtf[2].getText()+"'";

                        }

                        else if(size == 6)

                        {

                            System.out.println("hello3");

                                qry+=jtf[0].getText()+","+jtf[1].getText()+","+jtf[2].getText()+","+jtf[3].getText()+","+jtf[4].getText()+",";

                                /\*if(i != 4)

                                    qry2+= jtf[i].getText();

                                else

                                    qry2+=jtf[i];\*/

                            //System.out.println(qry2);

                            qry2 += jtf[1].getText()+","+jtf[2].getText()+","+jtf[3].getText()+","+jtf[4].getText()+") from dual";

                            System.out.println(qry2);

                            ResultSet rs = stmt.executeQuery(qry2);

                            while(rs.next())

                            {

                                qry += Integer.toString(rs.getInt(1));

                                break;

                            }

                        }

                        qry+=")";

                        System.out.println(qry);

                        stmt.executeQuery(qry);

                        textArea.setText("");

                        textArea.append("a row is inserted into "+*tablename*);

                        con.commit();

                    }

                    catch(Exception ex)

                    {

                        System.out.println(ex);

                        textArea.setText("");

                        textArea.append("couldnt perform insert");

                    }

                }

            }

            );

            sub.setBounds(x, y+spacing, textFieldWidth, height);

            insertPanel.add(sub);

            //f.setLayout(new GridLayout(size+1, 2, 3, 1));

            insertPanel.setSize(700,500);

            //System.out.println("hello");

            deletPanel.setVisible(false);

            selectPanel.setVisible(false);

            insertPanel.setVisible(true);

        }catch(Exception ex)

        {

            System.out.println(ex);

        }

    }

    private *void* update(String *tablename*)

    {

        deletPanel.removeAll();

        deletPanel.revalidate();

        deletPanel.repaint();

        insertPanel.removeAll();

        insertPanel.revalidate();

        insertPanel.repaint();

        updatPanel.removeAll();

        updatPanel.revalidate();

        updatPanel.repaint();

        try{

            Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","project","proj");

            Statement stmt = con.createStatement();

            String qry = "Select \* from "+*tablename*;

            ResultSet rs = stmt.executeQuery(qry);

            ResultSetMetaData rsm = rs.getMetaData();

            jtf = new JTextField[rsm.getColumnCount()];

            JLabel title = new JLabel(" DELETE ");

            updatPanel.add(title);

            JLabel[] jl = new JLabel[rsm.getColumnCount()];

            size = rsm.getColumnCount();

*int* x = 50; // Initial x-coordinate

*int* y = 50; // Initial y-coordinate

*int* labelWidth = 200; // Width of the label

*int* textFieldWidth = 150; // Width of the text field

*int* height = 25; // Height of each component

*int* spacing = 20; // Vertical spacing between components

            for(*int* i = 0;i<rsm.getColumnCount();i++)

            {

                jl[i] = new JLabel(rsm.getColumnName(i+1));

                jtf[i] = new JTextField();

                jl[i].setBounds(x,y,labelWidth,height);

                jtf[i].setBounds(x + labelWidth + spacing, y, textFieldWidth, height);

                updatPanel.add(jl[i]);

                updatPanel.add(jtf[i]);

                y += height+spacing;

            }

            JButton sub = new JButton("SUBMIT", null);

            sub.addActionListener(new ActionListener()

            {

                @*Override*

                public *void* actionPerformed(ActionEvent *e*)

                {

                    try{

                        Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","project","proj");

                        Statement stmt = con.createStatement();

                        String qry = "update "+*tablename*+" set ";

                        if(size == 2)

                        {

                            if(jtf[0].getText().length() == 0)

                            {

                                System.out.println("Cannot append");

                            }else

                            {

                                qry+= rsm.getColumnName(2)+" = '"+jtf[1].getText()+"' where "+rsm.getColumnName(1)+" = "+jtf[0].getText();

                            }

                        }else if(size == 3)

                        {

                            if(jtf[0].getText().length() == 0 || jtf[2].getText().length() == 0)

                            {

                                System.out.println("Cannot append");

                            }else{

                                qry += rsm.getColumnName(2)+" = "+jtf[1].getText()+" where "+rsm.getColumnName(1)+" = "+jtf[0].getText()+" and "+ rsm.getColumnName(3)+" = '"+jtf[2].getText()+" ' ";

                            }

                        }

                        else if(size == 6)

                        {

                            if(jtf[0].getText().length() == 0)

                            {

                                System.out.println("Cannot append");

                            }else{

*int* flag = 0;

*int*[] arr = new *int*[5];

                                    for(*int* j = 1;j<size-1;j++)

                                    {

                                        arr[j]= jtf[j].getText().length();

                                    }

                                    for(*int* j = 1;j<size-1;j++)

                                    {

                                        if(flag == 0)

                                        {

                                            qry += rsm.getColumnName(j+1) + " = "+ jtf[j].getText();

                                            flag = 1;

                                        }else{

                                            qry += " , "+rsm.getColumnName(j+1) + " = "+ jtf[j].getText();

                                        }

                                    }

                                    String qry2 = "select calc\_catg\_of\_drought("+jtf[1].getText()+","+jtf[2].getText()+","+jtf[3].getText()+","+jtf[4].getText()+") from dual";

                                    ResultSet rs = stmt.executeQuery(qry2);

                                    while(rs.next())

                                    {

                                        qry += ","+rsm.getColumnName(6)+" = "+rs.getInt(1);

                                        break;

                                    }

                                    qry += " where "+rsm.getColumnName(1)+" = "+jtf[0].getText();

                            }

                        }

                        System.out.println(qry);

                        stmt.executeQuery(qry);

                        textArea.setText("");

                        textArea.append(*tablename*+" is updated ");

                        con.commit();

                    }

                    catch(Exception ex)

                    {

                        ex.printStackTrace();

                        System.out.println(ex);

                    }

                }

            });

            sub.setBounds(x, y, textFieldWidth, height);

            updatPanel.add(sub);

            //f.setLayout(new GridLayout(size+1, 2, 3, 1));

            updatPanel.setSize(700,500);

            //System.out.println("hello");

            insertPanel.setVisible(false);

            selectPanel.setVisible(false);

            updatPanel.setVisible(true);

            deletPanel.setVisible(false);

        }catch(Exception e)

        {

            System.out.println(e);

        }

    }

    private *void* delete(String *tablename*)

    {

        insertPanel.removeAll();

        insertPanel.revalidate();

        insertPanel.repaint();

        updatPanel.removeAll();

        updatPanel.revalidate();

        updatPanel.repaint();

        deletPanel.removeAll();

        deletPanel.revalidate();

        deletPanel.repaint();

        try{

            Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","project","proj");

            Statement stmt = con.createStatement();

            String qry = "Select \* from "+*tablename*;

            ResultSet rs = stmt.executeQuery(qry);

            ResultSetMetaData rsm = rs.getMetaData();

            jtf = new JTextField[rsm.getColumnCount()];

            JLabel title = new JLabel(" DELETE ");

            //deletPanel.add(title);

            JLabel[] jl = new JLabel[rsm.getColumnCount()];

            size = rsm.getColumnCount();

*int* x = 50; // Initial x-coordinate

*int* y = 50; // Initial y-coordinate

*int* labelWidth = 200; // Width of the label

*int* textFieldWidth = 150; // Width of the text field

*int* height = 25; // Height of each component

*int* spacing = 20; // Vertical spacing between components

            for(*int* i = 0;i<rsm.getColumnCount();i++)

            {

                jl[i] = new JLabel(rsm.getColumnName(i+1));

                jtf[i] = new JTextField();

                jl[i].setBounds(x,y,labelWidth,height);

                jtf[i].setBounds(x + labelWidth + spacing, y, textFieldWidth, height);

                deletPanel.add(jl[i]);

                deletPanel.add(jtf[i]);

                y += height+spacing;

            }

            JButton sub = new JButton("SUBMIT", null);

            sub.addActionListener(new ActionListener()

            {

                @*Override*

                public *void* actionPerformed(ActionEvent *e*)

                {

                    try{

                        Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","project","proj");

                        Statement stmt = con.createStatement();

                        String qry = "Delete from "+*tablename*+" where ";

                        if(size == 2)

                        {

                            if(jtf[1].getText().length() == 0)

                            {

                                qry += rsm.getColumnName(1)+" = ";

                                qry += jtf[0].getText();

                            }else if(jtf[0].getText().length() == 0)

                            {

                                qry += rsm.getColumnName(2)+" = ";

                                qry += "'"+jtf[1].getText()+"'";

                            }else if(jtf[0].getText().length() != 0 && jtf[1].getText().length() != 0)

                            {

                                qry += rsm.getColumnName(1)+" = ";

                                qry += jtf[0].getText()+" and ";

                                qry += rsm.getColumnName(2)+" = ";

                                qry += "'"+jtf[1].getText()+"'";

                            }else

                            {

                                System.out.println("no");

                            }

                        }else if(size == 3)

                        {

*int*[] arr = new *int*[3];

                            for(*int* j = 0;j<3;j++)

                            {

                                arr[j] = jtf[j].getText().length();

                            }

*int* flag = 0;

                            for(*int* j = 0;j<size;j++)

                            {

                                if(arr[j] != 0)

                                {

                                    if(flag == 0)

                                    {

                                        if(j == size-1)

                                        {

                                            qry += rsm.getColumnName(j+1)+" = ";

                                            qry += "'"+jtf[j].getText()+"'";

                                        }else{

                                            qry += rsm.getColumnName(j+1)+" = ";

                                            qry += jtf[j].getText();

                                            flag = 1;

                                        }

                                    }else

                                    {

                                        if(j == size-1)

                                        {

                                            qry += " and "+rsm.getColumnName(j+1)+" = ";

                                            qry += "'"+jtf[j].getText()+"'";

                                        }else{

                                            qry += " and "+ rsm.getColumnName(j+1)+" = ";

                                            qry += jtf[j].getText();

                                        }

                                    }

                                }

                            }

                        }

                        else if(size == 6)

                        {

                            for(*int* i = 0;i<size;i++)

                            {

*int*[] arr = new *int*[3];

                                for(*int* j = 0;j<3;j++)

                                {

                                    arr[j] = jtf[j].getText().length();

                                }

*int* flag = 0;

                                for(*int* j = 0;j<size;j++)

                                {

                                    if(arr[j] != 0)

                                    {

                                        if(flag == 0)

                                        {

                                            qry += rsm.getColumnName(j+1)+" = ";

                                            qry += jtf[j].getText();

                                            flag = 1;

                                        }else

                                        {

                                            qry += " and "+ rsm.getColumnName(j+1)+" = ";

                                            qry += jtf[j].getText();

                                        }

                                    }

                                }

                            }

                        }

                        System.out.println(qry);

                        stmt.executeQuery(qry);

                        textArea.setText("");

                        textArea.append("1 row deleted from "+*tablename*);

                        con.commit();

                    }

                    catch(Exception ex)

                    {

                        System.out.println(ex);

                    }

                }

            });

            sub.setBounds(x, y, textFieldWidth, height);

            deletPanel.add(sub);

            //f.setLayout(new GridLayout(size+1, 2, 3, 1));

            deletPanel.setSize(500,400);

            //System.out.println("hello");

            insertPanel.setVisible(false);

            updatPanel.setVisible(false);

            selectPanel.setVisible(false);

            deletPanel.setVisible(true);

        }catch(Exception e)

        {

            System.out.println(e);

        }

    }

    private *void* createMenuBar()

    {

        JMenuBar menuBar = new JMenuBar();

        JMenu updateTables = new JMenu("UPDATE");

        JMenu insertTables = new JMenu("INSERT");

        JMenu deleteTables = new JMenu("DELETE");

        JMenu menuop = new JMenu("MENU");

        JMenu view = new JMenu("VIEW");

        JMenuItem[] updateItems = new JMenuItem[7];

        JMenuItem[] insertItems = new JMenuItem[7];

        JMenuItem[] deleteItems = new JMenuItem[7];

        JMenuItem[] viewItems = new JMenuItem[7];

        String[] tableNames = {"area", "area\_vegetation", "area\_moisture", "area\_reservoir\_storage", "area\_groundwater", "area\_rainfall","matrix\_impact\_indicators"};

        JMenuItem menu = new JMenuItem("MENU");

        JMenuItem exit = new JMenuItem("EXIT");

        menu.addActionListener(new ActionListener() {

            @*Override*

            public *void* actionPerformed(ActionEvent *e*){

                updatPanel.setVisible(false);

                deletPanel.setVisible(false);

                insertPanel.setVisible(false);

                selectPanel.setVisible(true);

            }

        }

        );

        exit.addActionListener(new ActionListener() {

            @*Override*

            public *void* actionPerformed(ActionEvent *e*){

                System.exit(0);

            }

        });

        menuop.add(menu);

        menuop.add(exit);

        for (*int* i = 0; i < 7; i++) {

            String tableName = tableNames[i];

            updateItems[i] = new JMenuItem(tableName);

            updateItems[i].addActionListener(new ActionListener() {

            @*Override*

            public *void* actionPerformed(ActionEvent *e*) {

                update(tableName);

            }

            });

        updateTables.add(updateItems[i]);

        insertItems[i] = new JMenuItem(tableName);

        insertItems[i].addActionListener(new ActionListener() {

            @*Override*

            public *void* actionPerformed(ActionEvent *e*) {

                System.out.println(tableName);

                insertvalues(tableName);

            }

        });

        insertTables.add(insertItems[i]);

        deleteItems[i] = new JMenuItem(tableName);

        deleteItems[i].addActionListener(new ActionListener() {

            @*Override*

            public *void* actionPerformed(ActionEvent *e*) {

                delete(tableName);

            }

        });

        deleteTables.add(deleteItems[i]);

        viewItems[i] = new JMenuItem(tableName);

        viewItems[i].addActionListener(new ActionListener() {

            @*Override*

            public *void* actionPerformed(ActionEvent *e*) {

                viewtable(tableName);

            }

        });

        view.add(viewItems[i]);

        }

        menuBar.add(menuop);

        menuBar.add(updateTables);

        menuBar.add(insertTables);

        menuBar.add(deleteTables);

        menuBar.add(view);

        setJMenuBar(menuBar);

    }

    public static *void* main(String[] *args*) {

        new frameclass();

    }

}

**The plsql function used to generate the category of drought of a place is :-**

create or replace function calc\_catg\_of\_drought(d\_id number,sto\_id number,g\_water\_id number,vci\_id number)

2 return number

3 is

4 cod number;

5 sd number;

6 md number;

7 nd number;

8 begin

9

10 nd:= 0;

11 md := 0;

12 sd:= 0;

13

14 if d\_id <1 then

15 nd := nd+1;

16 elsif d\_id<=2 then

17 md := md+1;

18 elsif d\_id = 3 then

19 sd := sd+1;

20 end if;

21

22 if sto\_id<=1 then

23 nd := nd+1;

24 elsif sto\_id=2 then

25 md:= md+1;

26 elsif sto\_id<=4 then

27 sd := sd+1;

28 end if;

29

30 if g\_water\_id<=1 then

31 nd := nd+1;

32 elsif g\_water\_id=2 then

33 md:= md+1;

34 elsif g\_water\_id<=4 then

35 sd := sd+1;

36 end if;

37

38 if vci\_id=0 then

39 nd:= nd+1;

40 elsif vci\_id = 1 then

41 md:= md+1;

42 elsif vci\_id = 2 then

43 sd:= sd+1;

44 end if;

45

46 if sd = 4 then

47 cod:=4;

48 elsif md >= 2 or sd >= 2 then

49 cod := 2;

50 else

51 cod := 0;

52 end if;

53

54 return cod;

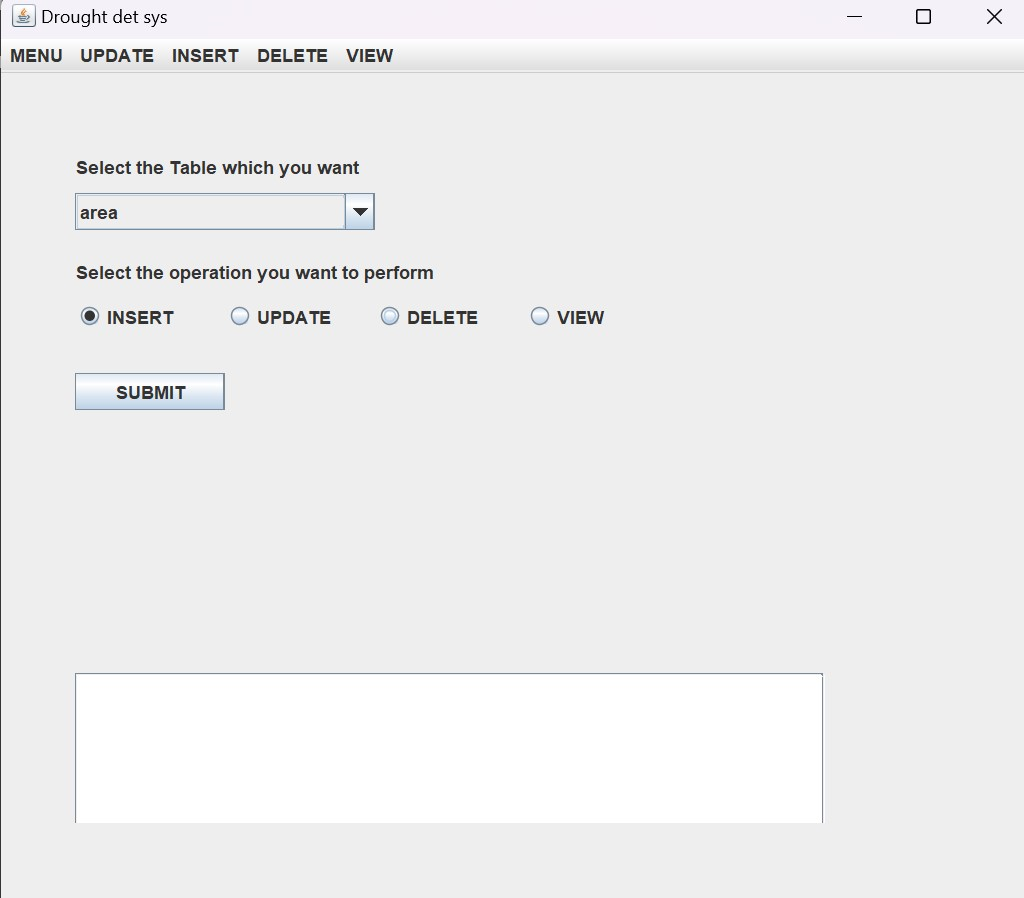
55 end;

56 /

This function generates the drought index id for a particular area.

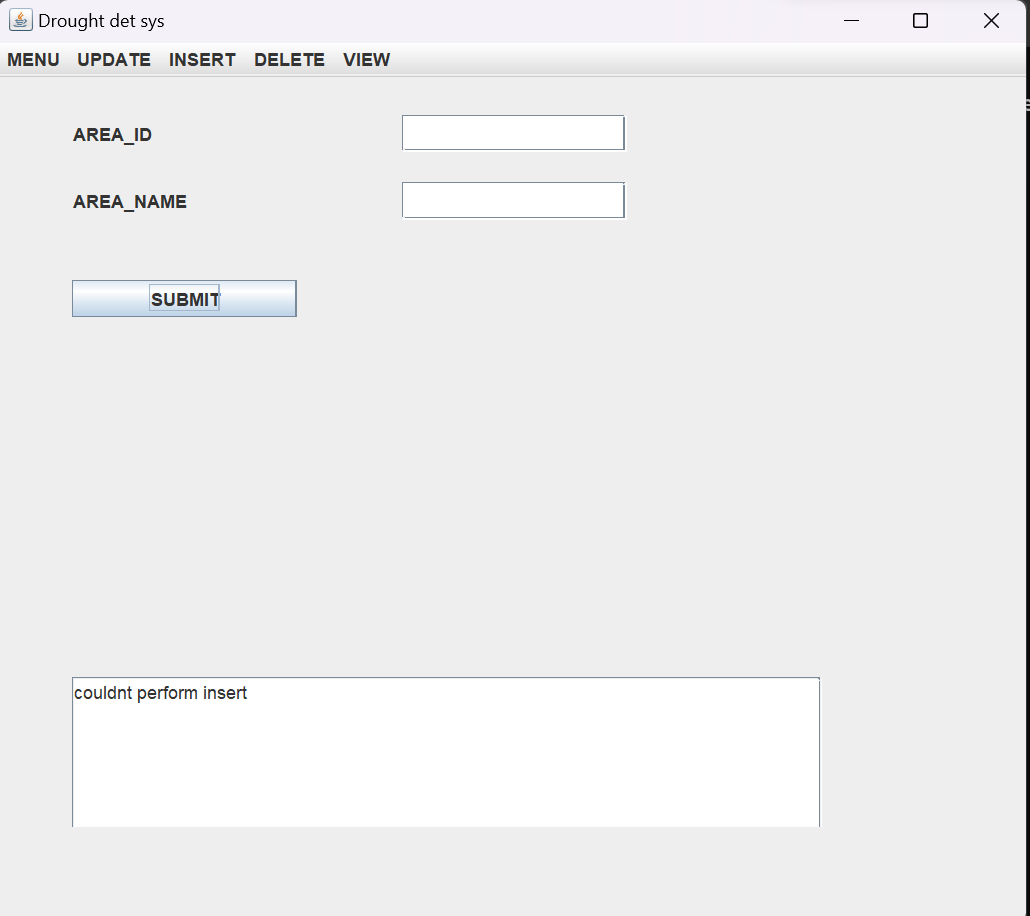
**TESTING**

**Homepage:-**

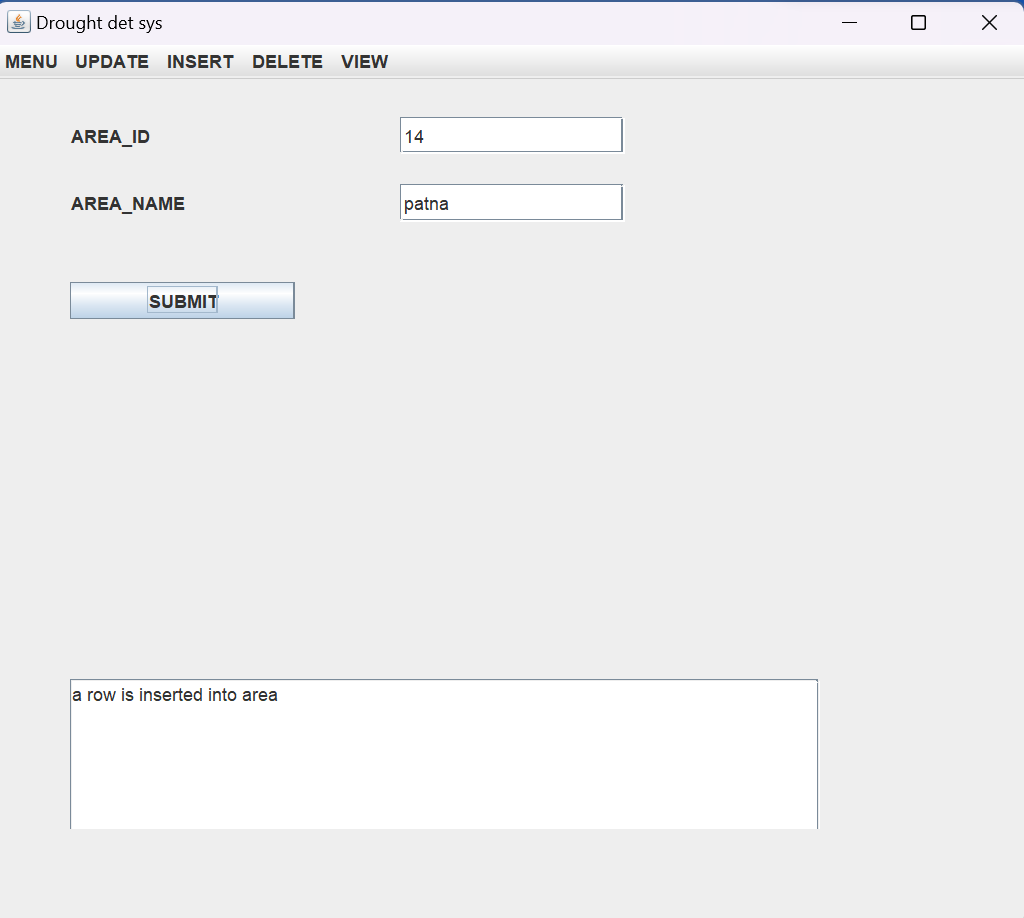


**Insert operations :-**

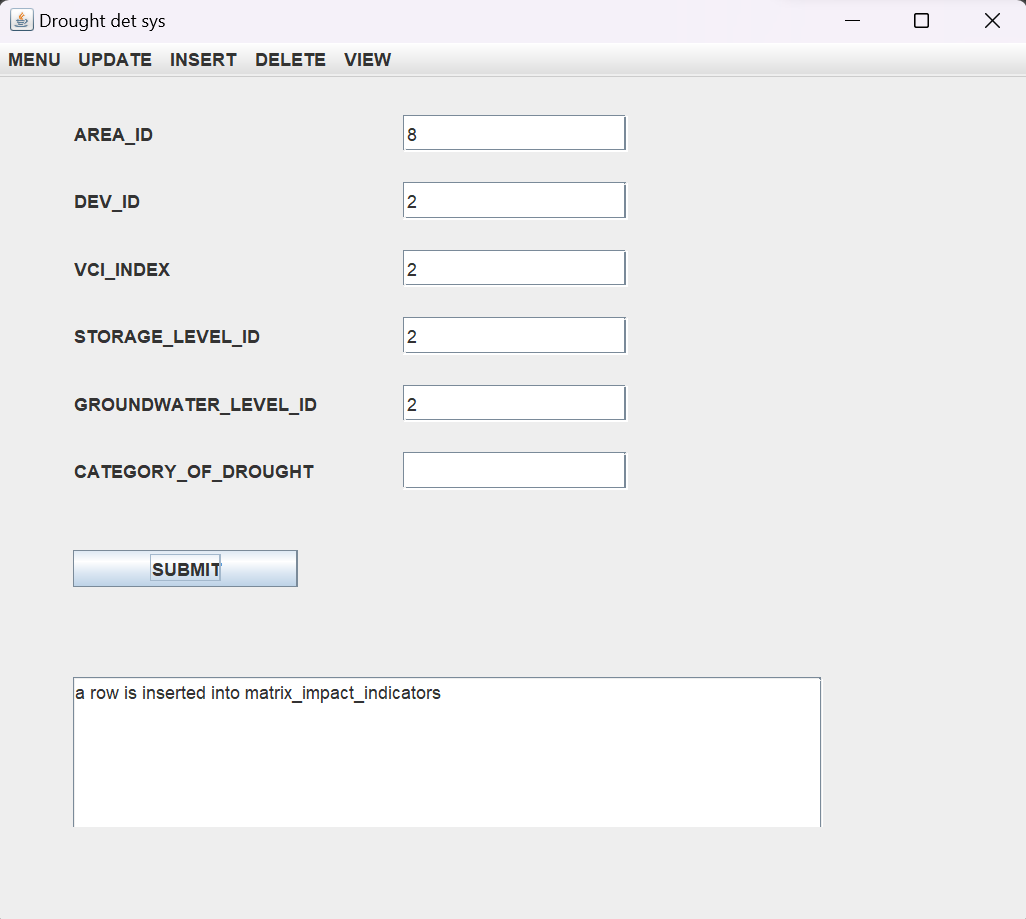
Insert without giving any information:-



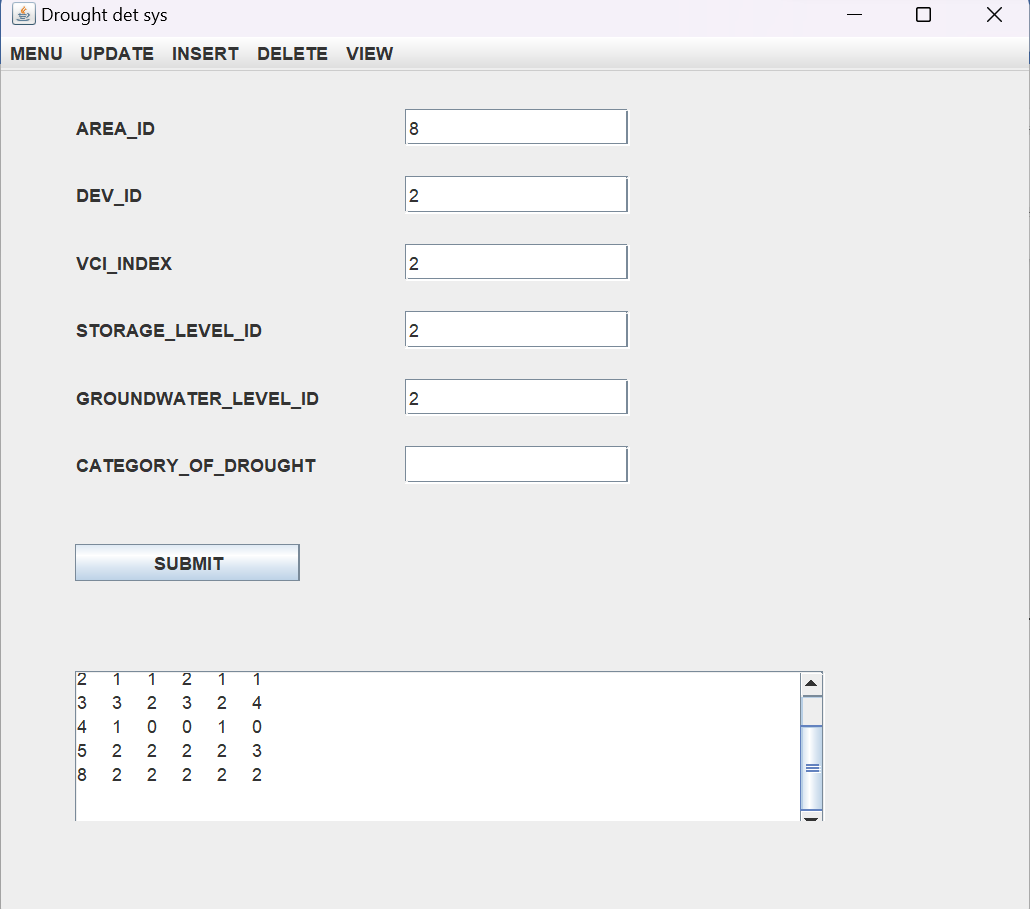
Insert with information:-



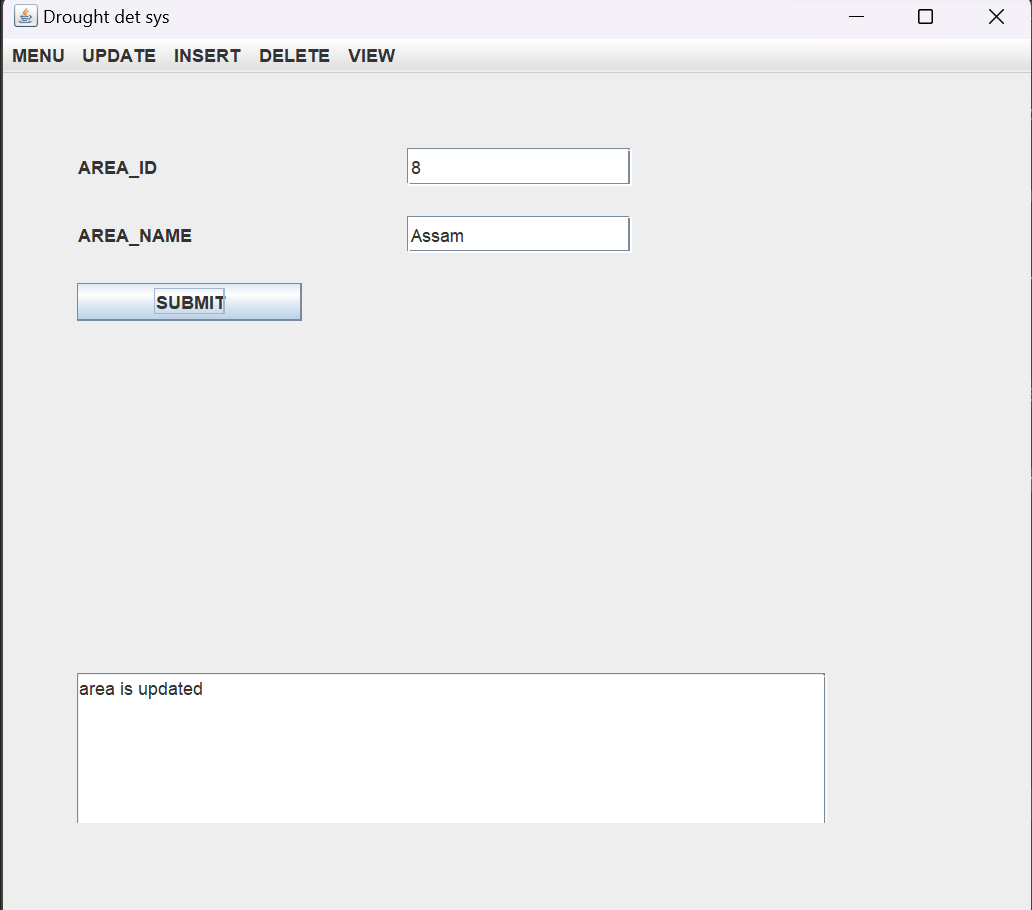
Insert by using plsql function:-



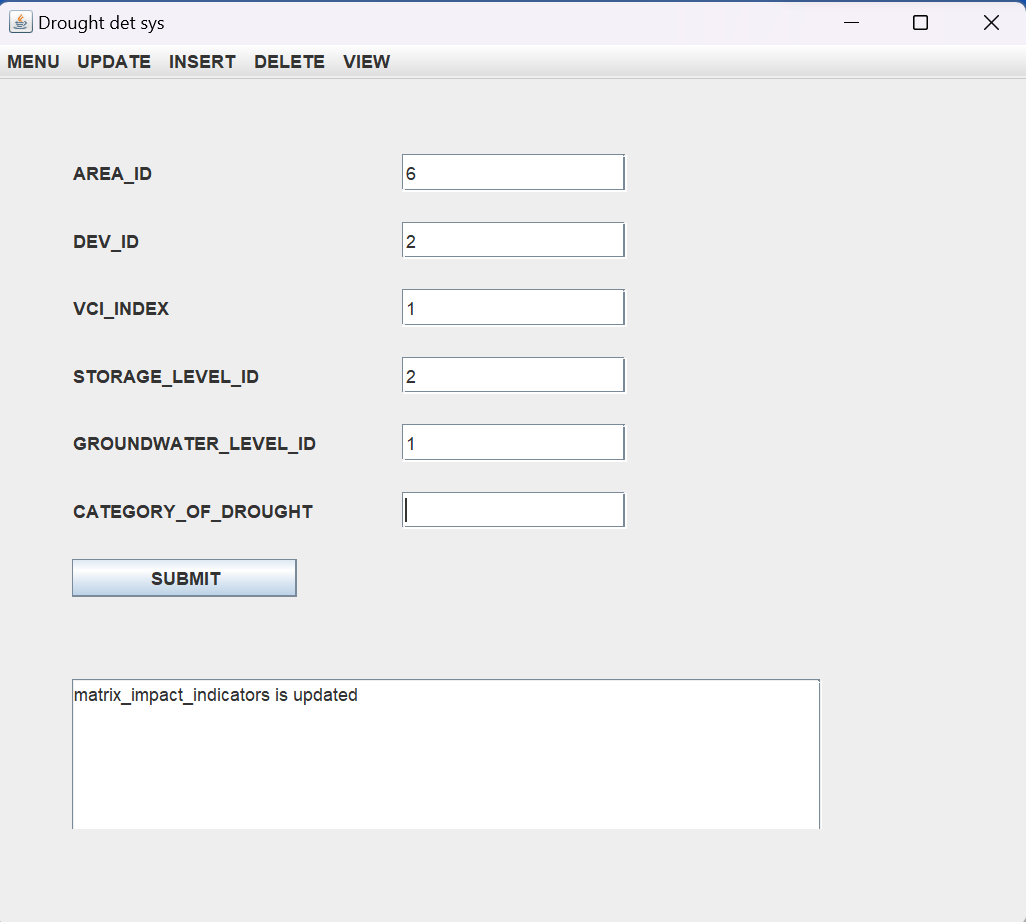
Matrix\_impact indicators table after insertion:-

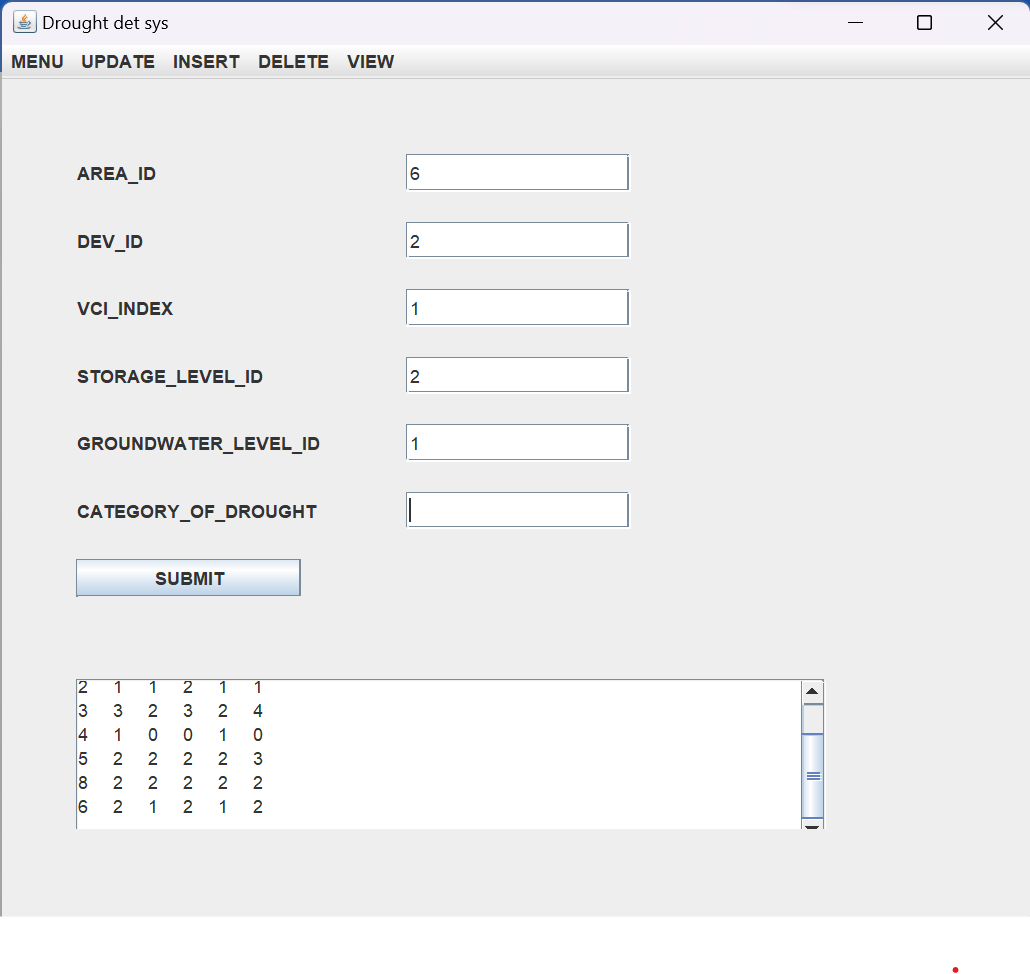


Updating of table area:-

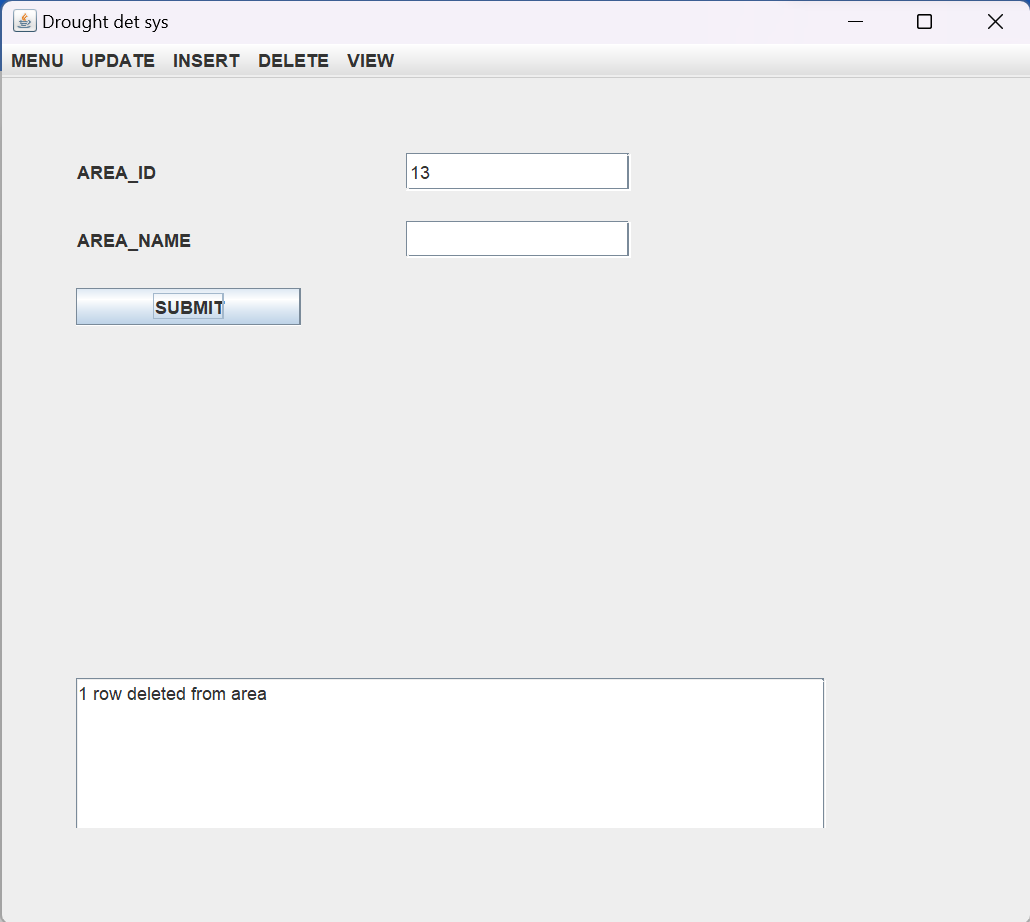


Updating table with plsql function:-





Deleting from tables: -



**Result :-**

I have successfully implemented **drought detection system** project

**Reference and future work:-**

This project drought detection system has several potential areas of future development and expansion. As technology advances and our understanding of droughts improves, there are numerous opportunities to enhance the system's functionality, incorporate additional data sources, and provide more advanced analysis and prediction capabilities.

The future scope of the Drought Detection System is vast and promising. By incorporating advanced data sources, machine learning techniques, climate models, GIS visualization, real-time monitoring, and citizen engagement, the system can evolve into a comprehensive and proactive drought management platform.

**References :-**

1. Java docs :- <https://docs.oracle.com/en/java/>
2. Oracle jdbc tutorial <https://docs.oracle.com/javase/tutorial/jdbc/basics/>