A

Project Report

On

Food Corporation of India Management System

Developed By

Yash Patel-Department of IT, DD University
Mihir Prajapati-Department of IT, DD University

Guided By

Internal Guide:
Prof. Sunil K. Vithlani
Department of Information Technology
Faculty of Technology
DD University

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad-387001
October-2019

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "Food Corporation of India Management System" **is a bonafide report of the work carried out by**

1) Mr. Yash Patel, Student ID No: 17ITUOS080

2) Mr. Mihir Prajapati, Student ID No: 17ITUBS131

Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2019-2020.

Prof. Sunil K. Vithlani
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. Vipul Dabhi
Head, Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank Prof. Sunil K. Vithlani for providing us an opportunity to do the project work in DBMS and giving us all support and guidance which made us complete the project duly. We are extremely thank full to him for providing such a nice support, guidance by taking keen interest in our project, although he had busy schedule managing the lectures.

We would also like to express our special thanks of gratitude to our HOD Prof. Vipul Dabhi who gave us the golden opportunity to do this wonderful project on the topic: Fci Management System. We would also like to thank him for including such things in curriculum making it more interesting and useful practically.

Finally, we would like to thank each and every person who was there around us helping more or less in our project and keeping us motivated to work hard and complete the project. We would also thank them for inspiring us and sharing their ideas and views to make this project a success.

Yours sincerely,

Yash Patel (IT-093)
Mihir Prajapati (IT-098)

TABLE OF CONTENTS

1.1 SYSTEM OVERVIEW

FCI (FOOD CORPORATION OF INDIA) is an organization in which stores grain in different Type of Godown. It also moves (inward/outward) grain from different state. FCI allots grain to different scheme through different allotment type. FCI is handle by Government Of India With the help of this application we try to achieve that these all above work done easily and properly.

BRIEF DESCRIPTION OF SYSTEM:

FOOD CORPORATION OF INDIA MANAGEMENT SYSTEM This software firstly maintains the employee record of FOOD CORPORATION OF INDIA. We can view all information of employee in different format. This application is also manage the record of Godown and their capacity. When we want to store the grain in a Godown we check the capacity firstly. Same As in allocation and Movement we also check the stock and capacity of godown. In this System we also put information of procured grain.

1.2 Advantages of the proposed system

1).The purpose of the application is to help the director to get details about the FCI.

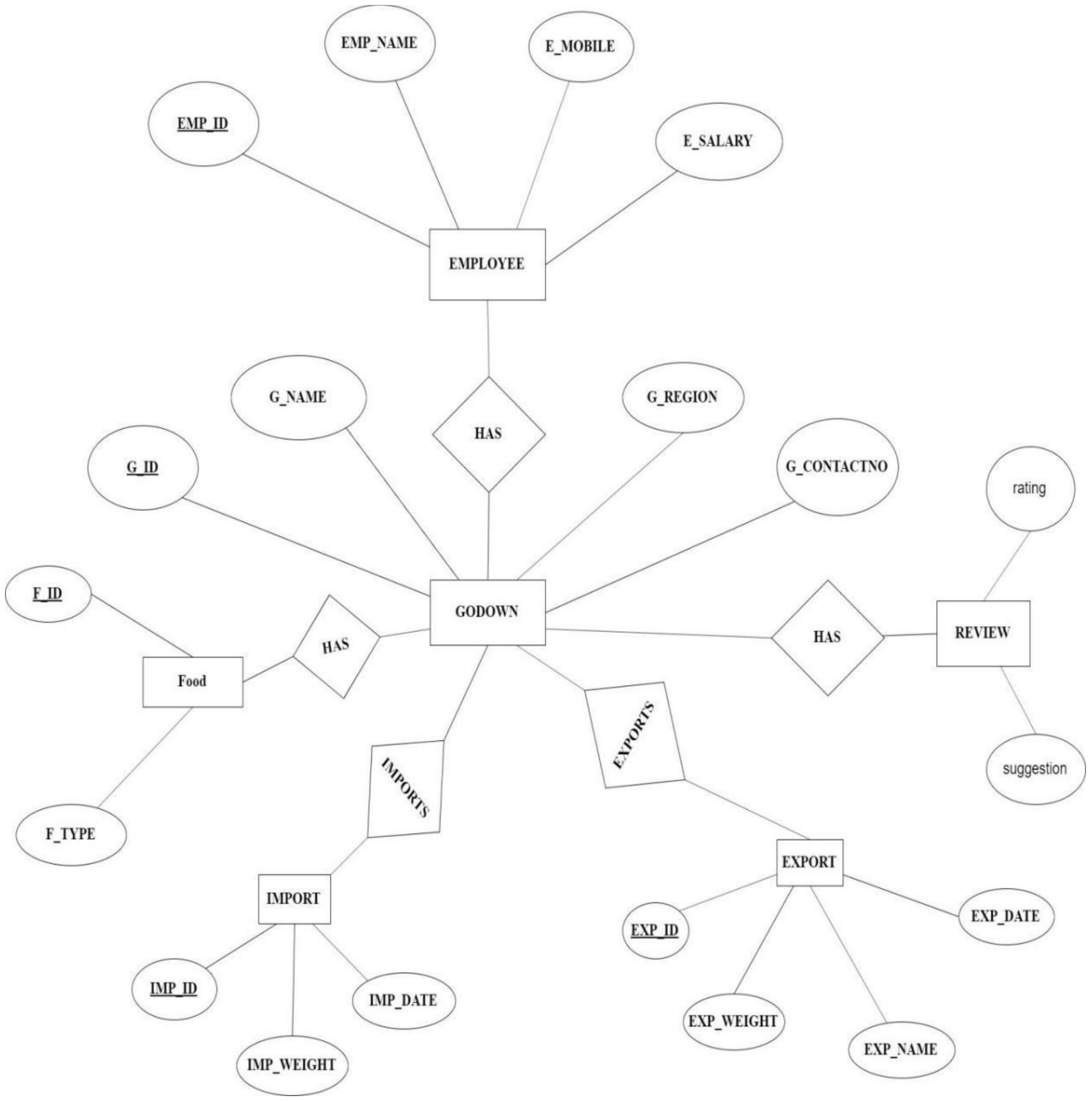
2). It help the Godown In-charge to manage the Godown details, and their capacity easily.

3).This application also help to movement and storage In-charge.

4).This application is also useful for manage detail of Procurement.

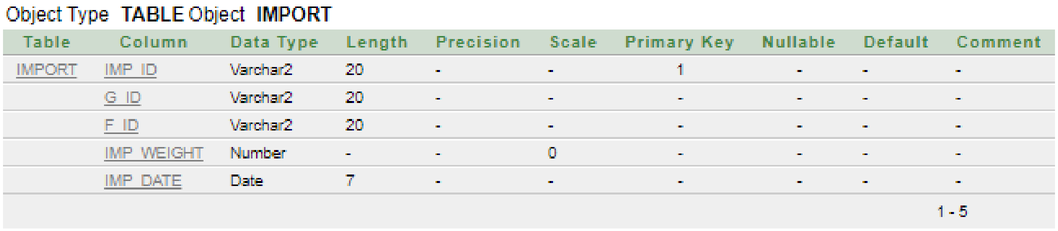
5). It also calculate the total food and vacant space in godown.

2. E-R DIAGRAM


EMP_NAME
E_MOBILE
EMP_ID
E_SALARY
EMPLOYEE
G_NAME
HAS
G_REGION
G_ID
G_CONTACTNO
rating
F_ID
HAS
GODOWN
HAS
REVIEW
Food
EXPORTS
IMPORTS
suggestion
F_TYPE
EXPORT
IMPORT
EXP_DATE
EXP_ID
IMP_ID
IMP_DATE
EXP_WEIGHT
EXP_NAME
IMP_WEIGHT

3. DATA DICTIONARY

IMPORT

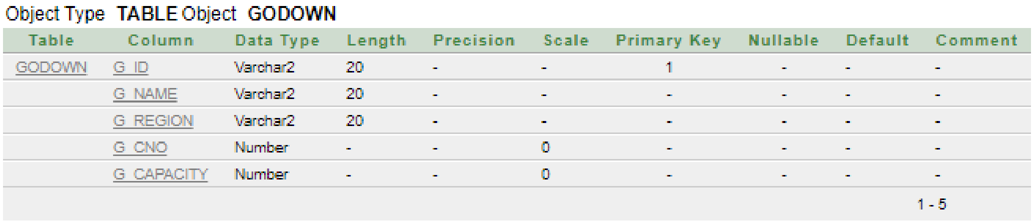


Object Type TABLE Object IMPORT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
IMPORT	IMP_ID	Varchar2	20	-	-	1	-	-	-
	G_ID	Varchar2	20	-	-	-	-	-	-
	F_ID	Varchar2	20	-	-	-	-	-	-
	IMP_WEIGHT	Number	-	-	0	-	-	-	-
	IMP_DATE	Date	7	-	-	-	-	-	-

1 - 5

GODOWN

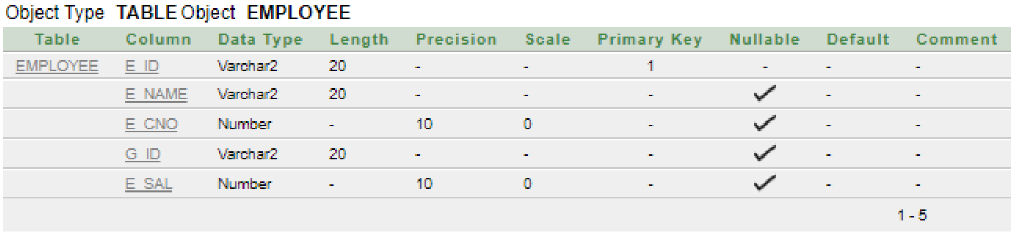


Object Type TABLE Object GODOWN

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
GODOWN	G_ID	Varchar2	20	-	-	1	-	-	-
	G_NAME	Varchar2	20	-	-	-	-	-	-
	G_REGION	Varchar2	20	-	-	-	-	-	-
	G_CNO	Number	-	-	0	-	-	-	-
	G_CAPACITY	Number	-	-	0	-	-	-	-

1 - 5

EMPLOYEE

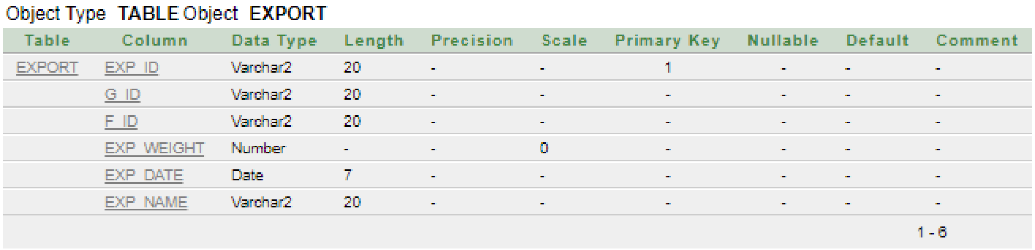


Object Type TABLE Object EMPLOYEE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	E_ID	Varchar2	20	-	-	1	-	-	-
	E_NAME	Varchar2	20	-	-	-	✓	-	-
	E_CNO	Number	-	10	0	-	✓	-	-
	G_ID	Varchar2	20	-	-	-	✓	-	-
	E_SAL	Number	-	10	0	-	✓	-	-

1 - 5

EXPORT

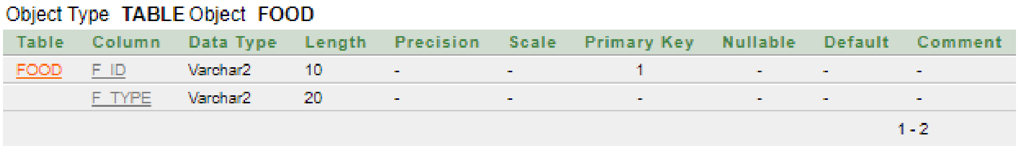


Object Type TABLE Object EXPORT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EXPORT	EXP_ID	Varchar2	20	-	-	1	-	-	-
	G_ID	Varchar2	20	-	-	-	-	-	-
	F_ID	Varchar2	20	-	-	-	-	-	-
	EXP_WEIGHT	Number	-	-	0	-	-	-	-
	EXP_DATE	Date	7	-	-	-	-	-	-
	EXP_NAME	Varchar2	20	-	-	-	-	-	-

1 - 6

FOOD

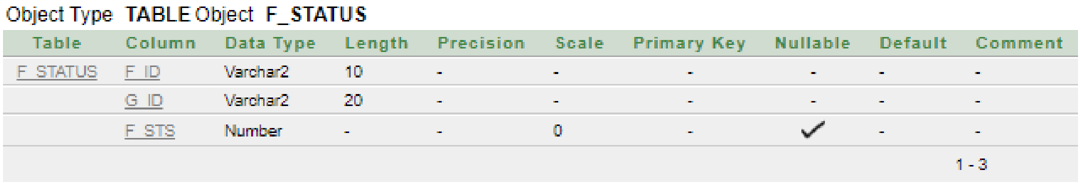


Object Type **TABLE** Object **FOOD**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FOOD	F_ID	Varchar2	10	-	-	1	-	-	-
	F_TYPE	Varchar2	20	-	-	-	-	-	-

1 - 2

STATUS

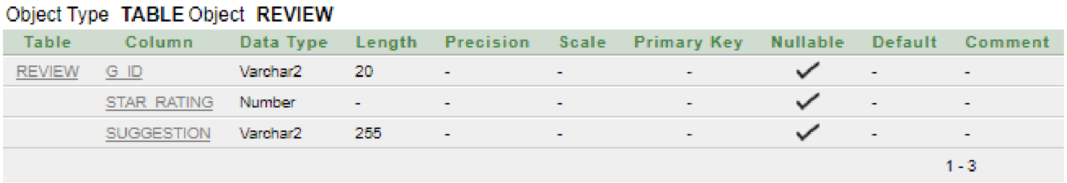


Object Type **TABLE** Object **F_STATUS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
F_STATUS	F_ID	Varchar2	10	-	-	-	-	-	-
	G_ID	Varchar2	20	-	-	-	-	-	-
	F_STS	Number	-	-	0	-	✓	-	-

1 - 3

REVIEW

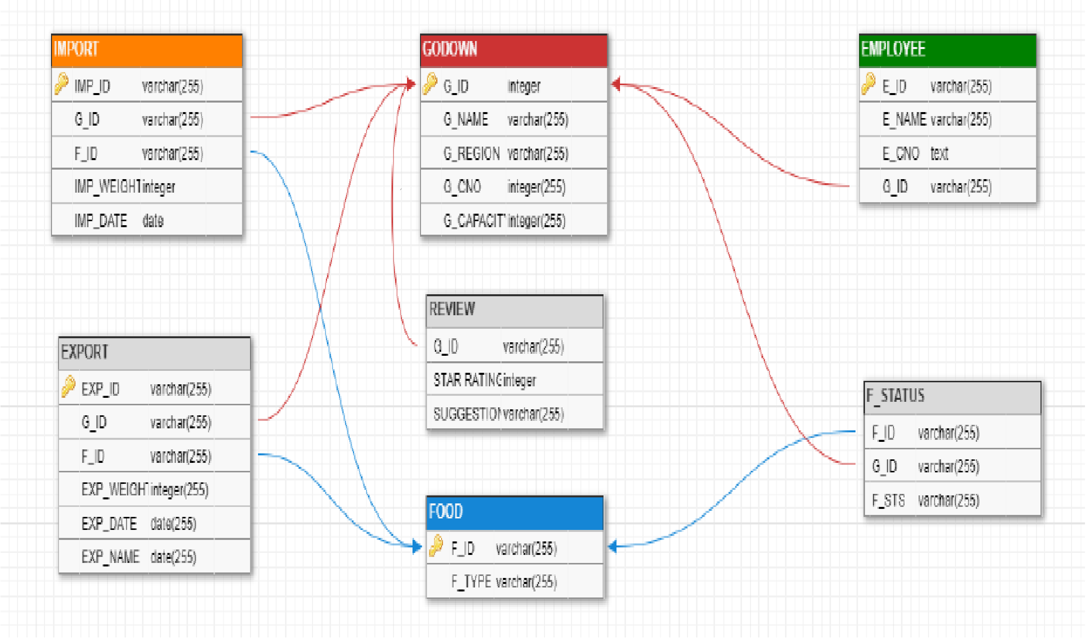


Object Type **TABLE** Object **REVIEW**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
REVIEW	G_ID	Varchar2	20	-	-	-	✓	-	-
	STAR_RATING	Number	-	-	-	-	✓	-	-
	SUGGESTION	Varchar2	255	-	-	-	✓	-	-

1 - 3

4. SCHEMA DIAGRAM



5. DATABASE IMPLEMENTATION

5.1 CREATE SCHEMA

- Create Godown Table:

CREATE TABLE GODOWN(G_ID VARCHAR2(20) NOT NULL PRIMARY KEY ,G_NAME VARCHAR2(20) NOT NULL,G_REGION VARCHAR2(20) NOT NULL,G_CNO INT NOT NULL,G_CAPACITY INTEGER NOT NULL);

- Create Employee Table:

CREATE TABLE EMPLOYEE (E_ID VARCHAR2(20), E_NAME VARCHAR2(20) , E_CNO NUMBER(10) , G_ID VARCHAR2 (20) , E_SAL NUMBER (10) , PRIMARY KEY (E_ID) , FOREIGN KEY(G_ID) REFERENCES GODOWN(G_ID));

- Create Import Table:

CREATE TABLE IMPORT (IMP_ID VARCHAR2(20) NOT NULL PRIMARY KEY , G_ID VARCHAR2(20) NOT NULL , F_ID VARCHAR2(20) NOT NULL , IMP_WEIGHT INTEGER NOT NULL , IMP_DATE DATE NOT NULL , FOREIGN KEY(G_ID) REFERENCES GODOWN(G_ID) , FOREIGN KEY(F_ID) REFERENCES FOOD(F_ID));

- Create Expert Table:

CREATE TABLE EXPORT (EXP_ID VARCHAR2(20) NOT NULL PRIMARY KEY , G_ID VARCHAR2(20) NOT NULL , F_ID VARCHAR2(20) NOT NULL , EXP_WEIGHT INTEGER NOT NULL , EXP_DATE DATE NOT NULL, EXP_NAME VARCHAR2(20) NOT NULL , FOREIGN KEY(G_ID) REFERENCES GODOWN(G_ID) , FOREIGN KEY(F_ID) REFERENCES FOOD(F_ID));

- Create Food Table:

CREATE TABLE FOOD (F_ID VARCHAR2(10) NOT NULL PRIMARY KEY ,F_TYPE VARCHAR2(20) NOT NULL);

- Create Food Status Table:

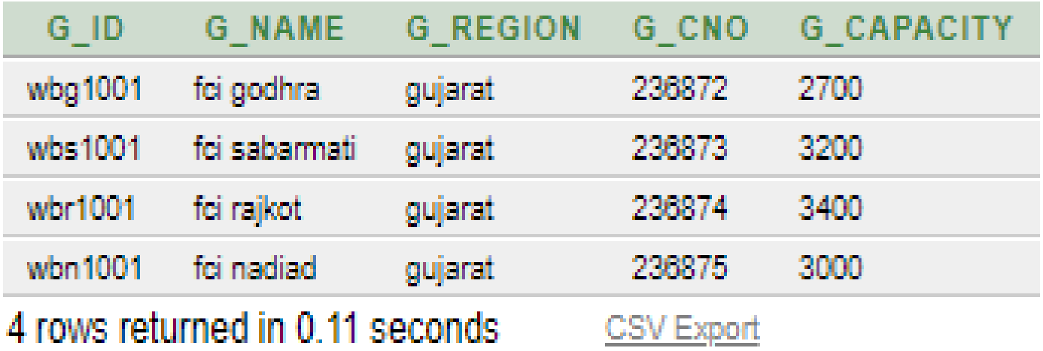
```
CREATE TABLE F_STATUS( F_ID VARCHAR2(10) NOT NULL,G_ID
VARCHAR2(20) NOT NULL , F_STS INTEGER  , FOREIGN KEY(G_ID)
REFERENCES GODOWN(G_ID)  , FOREIGN KEY(F_ID) REFERENCES
FOOD(F_ID));
```

5.2 INSERT DATA VALUES

- Insert values into Godown Table:

```
INSERT INTO GODOWN (G_ID, G_NAME, G_REGION, G_CNO,
G_CAPACITY) VALUES ('wbg1001','fci godhra','gujarat', 236272, 2500);

INSERT INTO GODOWN (G_ID, G_NAME, G_REGION, G_CNO,
G_CAPACITY) VALUES ('wbs1001','fci sabarmati','gujarat', 236273, 3000);

INSERT INTO GODOWN (G_ID, G_NAME, G_REGION, G_CNO,
G_CAPACITY) VALUES ('wbr1001','fci rajkot','gujarat', 236274, 3200);

INSERT INTO GODOWN (G_ID, G_NAME, G_REGION, G_CNO,
G_CAPACITY) VALUES ('wbn1001','fci nadiad','gujarat', 236275, 2800);
```

```
SELECT * FROM GODOWN;
```



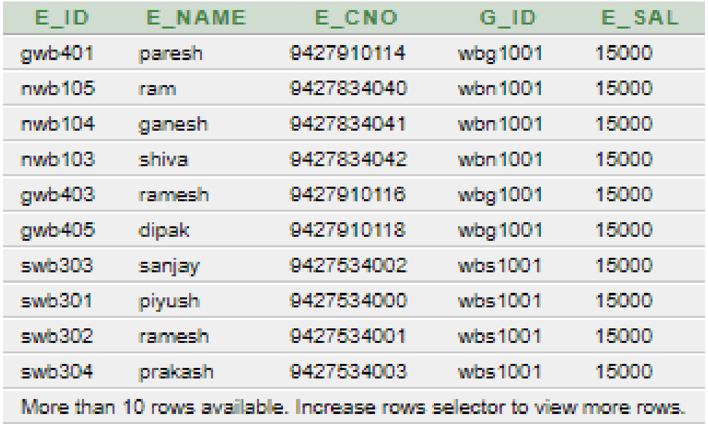
G_ID	G_NAME	G_REGION	G_CNO	G_CAPACITY
wbg1001	fci godhra	gujarat	236872	2700
wbs1001	fci sabarmati	gujarat	236873	3200
wbr1001	fci rajkot	gujarat	236874	3400
wbn1001	fci nadiad	gujarat	236875	3000

4 rows returned in 0.11 seconds

- Insert Values in Employee Table:

```
INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('gwb401','paresh', 9427910114,'wbg1001', 10000);

INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('gwb402','nayan', 9427910115,'wbg1001', 10000);

INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('gwb403','ramesh', 9427910116,'wbg1001', 10000);

INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('gwb405','dipak', 9427910118,'wbg1001', 10000);

INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('swb303','sanjay', 9427534002,'wbs1001', 10000);

INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('swb301','piyush', 9427534000,'wbs1001', 10000);

INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('swb302','ramesh', 9427534001,'wbs1001', 10000);

INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('swb304','prakash', 9427534003,'wbs1001', 10000);

INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('swb305','girish', 9427534004,'wbs1001', 10000);

INSERT INTO EMPLOYEE (E_ID, E_NAME, E_CNO, G_ID, E_SAL) VALUES
('rwb201','binkesh', 9427534060,'wbr1001', 10000);
```

```
SELECT * FROM EMPLOYEE;
```



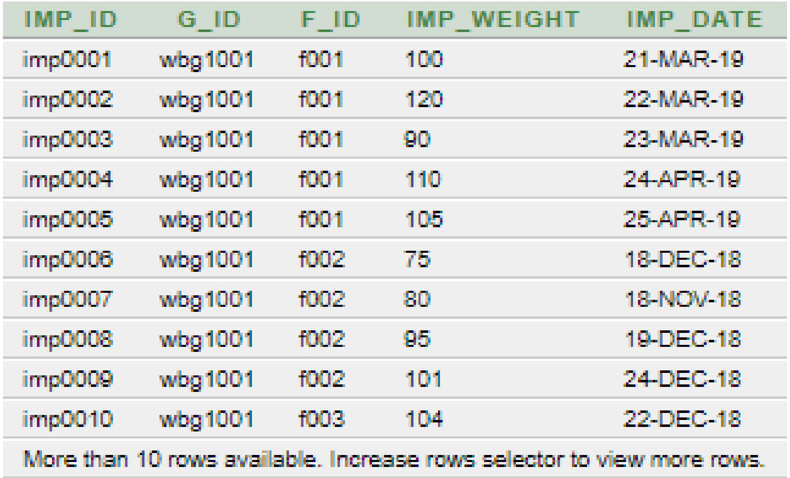
E_ID	E_NAME	E_CNO	G_ID	E_SAL
gwb401	paresh	9427910114	wbg1001	15000
nwb105	ram	9427834040	wbn1001	15000
nwb104	ganesh	9427834041	wbn1001	15000
nwb103	shiva	9427834042	wbn1001	15000
gwb403	ramesh	9427910116	wbg1001	15000
gwb405	dipak	9427910118	wbg1001	15000
swb303	sanjay	9427534002	wbs1001	15000
swb301	piyush	9427534000	wbs1001	15000
swb302	ramesh	9427534001	wbs1001	15000
swb304	prakash	9427534003	wbs1001	15000

More than 10 rows available. Increase rows selector to view more rows.

- Insert Values in Import Table

```
INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0001','wbg1001','f001', 100,'21-Mar-2019');

INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0002','wbg1001','f001', 120,'22-Mar-2019');

INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0003','wbg1001','f001', 90,'23-Mar-2019');

INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0004','wbg1001','f001', 110,'24-Apr-2019');

INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0005','wbg1001','f001', 105,'25-Apr-2019');

INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0006','wbg1001','f002', 75,'18-Dec-2018');

INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0007','wbg1001','f002', 80,'18-Nov-2018');

INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0008','wbg1001','f002', 95,'19-Dec-2018');

INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0009','wbg1001','f002', 101,'24-Dec-2018');

INSERT INTO IMPORT (IMP_ID, G_ID, F_ID, IMP_WEIGHT, IMP_DATE)
VALUES ('imp0010','wbg1001','f003', 104,'22-Dec-2018');
```

Import Table:



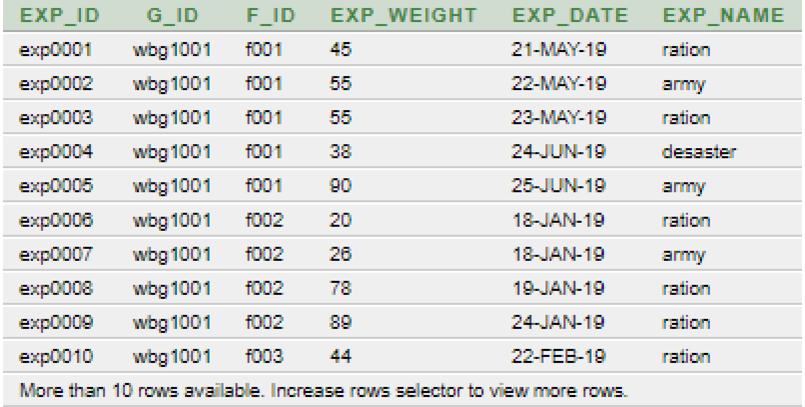
IMP_ID	G_ID	F_ID	IMP_WEIGHT	IMP_DATE
imp0001	wbg1001	f001	100	21-MAR-19
imp0002	wbg1001	f001	120	22-MAR-19
imp0003	wbg1001	f001	90	23-MAR-19
imp0004	wbg1001	f001	110	24-APR-19
imp0005	wbg1001	f001	105	25-APR-19
imp0006	wbg1001	f002	75	18-DEC-18
imp0007	wbg1001	f002	80	18-NOV-18
imp0008	wbg1001	f002	95	19-DEC-18
imp0009	wbg1001	f002	101	24-DEC-18
imp0010	wbg1001	f003	104	22-DEC-18

More than 10 rows available. Increase rows selector to view more rows.

- Insert Values in Export Table

```
INSERT INTO EXPORT (EXP_ID, G_ID,
F_ID,EXP_WEIGHT,EXP_DATE,EXP_NAME) VALUES
('exp0001','wbg1001','f001',45,'21-May-2019','ration');

INSERT INTO
EXPORT(EXP_ID,G_ID,F_ID,EXP_WEIGHT,EXP_DATE,EXP_NAME) VALUES
('exp0002','wbg1001','f001',55,'22-May-2019','army');

INSERT INTO EXPORT (EXP_ID, G_ID, F_ID, EXP_WEIGHT, EXP_DATE,
EXP_NAME) VALUES ('exp0003','wbg1001','f001', 55,'23-May-2019','ration');

INSERT INTO
EXPORT(EXP_ID,G_ID,F_ID,EXP_WEIGHT,EXP_DATE,EXP_NAME) VALUES
('exp0004','wbg1001','f001',38,'24-Jun-2019','desaster');

INSERT INTO
EXPORT(EXP_ID,G_ID,F_ID,EXP_WEIGHT,EXP_DATE,EXP_NAME) VALUES
('exp0005','wbg1001','f001',90,'25-Jun-2019','army');

INSERT INTO EXPORT
(EXP_ID,G_ID,F_ID,EXP_WEIGHT,EXP_DATE,EXP_NAME) VALUES
('exp0006','wbg1001','f002',20,'18-Jan-2019','ration');

INSERT INTO EXPORT
(EXP_ID,G_ID,F_ID,EXP_WEIGHT,EXP_DATE,EXP_NAME) VALUES
('exp0007','wbg1001','f002',26,'18-Jan-2019','army');

INSERT INTO EXPORT
(EXP_ID,G_ID,F_ID,EXP_WEIGHT,EXP_DATE,EXP_NAME) VALUES
('exp0008','wbg1001','f002',78,'19-Jan-2019','ration');

INSERT INTO EXPORT
(EXP_ID,G_ID,F_ID,EXP_WEIGHT,EXP_DATE,EXP_NAME) VALUES
('exp0009','wbg1001','f002',89,'24-Jan-2019','ration');

INSERT INTO EXPORT
(EXP_ID,G_ID,F_ID,EXP_WEIGHT,EXP_DATE,EXP_NAME) VALUES
('exp0010','wbg1001','f003',44,'22-Feb-2019','ration');
```

SELECT *FROM EXPORT;



EXP_ID	G_ID	F_ID	EXP_WEIGHT	EXP_DATE	EXP_NAME
exp0001	wbg1001	f001	45	21-MAY-19	ration
exp0002	wbg1001	f001	55	22-MAY-19	army
exp0003	wbg1001	f001	55	23-MAY-19	ration
exp0004	wbg1001	f001	38	24-JUN-19	desaster
exp0005	wbg1001	f001	90	25-JUN-19	army
exp0006	wbg1001	f002	20	18-JAN-19	ration
exp0007	wbg1001	f002	26	18-JAN-19	army
exp0008	wbg1001	f002	78	19-JAN-19	ration
exp0009	wbg1001	f002	89	24-JAN-19	ration
exp0010	wbg1001	f003	44	22-FEB-19	ration

More than 10 rows available. Increase rows selector to view more rows.

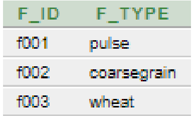
- Insert values into Food Table:

INSERT INTO FOOD (F_ID, F_TYPE) VALUES ('f001','pulse');

INSERT INTO FOOD (F_ID, F_TYPE) VALUES ('f002','coarsegrain');

INSERT INTO FOOD (F_ID, F_TYPE) VALUES ('f003','wheat');

SELECT * FROM FOOD;

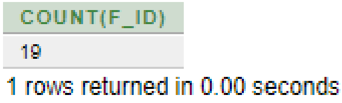


F_ID	F_TYPE
f001	pulse
f002	coarsegrain
f003	wheat

5.3 QUERIES

1) Total import count of particular godown

Select count (F_ID) from IMPORT where F_ID='f001'



```
COUNT(F_ID)
19
1 rows returned in 0.00 seconds
```

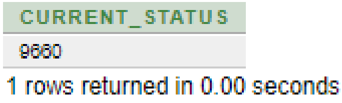
2) RETRIVE THE CURRENT STATUS OF GODOWN

SELECT SUM (IMP_WEIGHT) - SUM (EXP_WEIGHT) AS CURRENT_STATUS

FROM IMPORT

INNER JOIN EXPORT

ON IMPORT.G_ID = 'wbg1001' and EXPORT.G_ID='wbg1001';



```
CURRENT_STATUS
9660
1 rows returned in 0.00 seconds
```

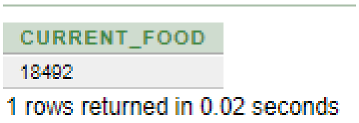
3) RETRIVE THE CURRENT STATUS OF FOOD

SELECT SUM (IMP_WEIGHT) – SUM (EXP_WEIGHT) AS CURRENT_FOOD

FROM IMPORT

INNER JOIN EXPORT

ON IMPORT.F_ID = 'f001' and EXPORT.F_ID='f001';



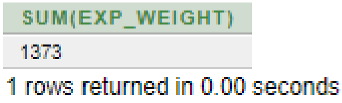
```
CURRENT_FOOD
18492
1 rows returned in 0.02 seconds
```

4) RETRIVE TOTAL WEIGHT OF EXPORT TO "ARMY"

SELECT SUM (EXP_WEIGHT)

FROM EXPORT

WHERE EXP_NAME='army';

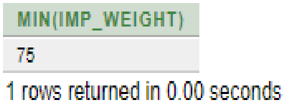


```
SUM(EXP_WEIGHT)
1373
1 rows returned in 0.00 seconds
```

5) FETCH THE MINIMUM IMPORT FROM ALL GODOWN

SELECT MIN (IMP_WEIGHT)

FROM IMPORT;



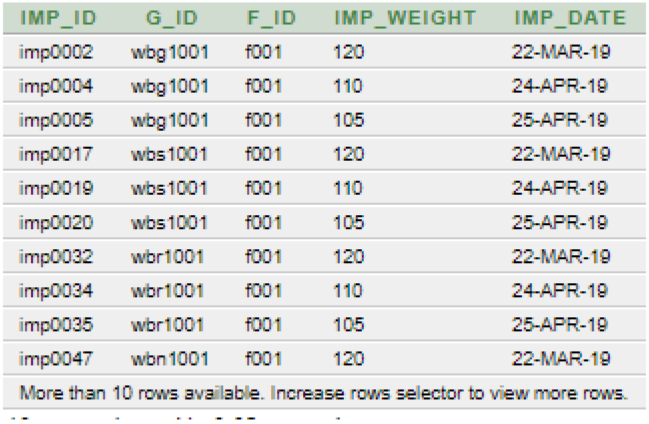
MIN(IMP_WEIGHT)
75

1 rows returned in 0.00 seconds

6) FETCH IMPORT OF PERTICULAR GODOWN IS GREATER THEN 1000:

SELECT * FROM IMPORT

WHERE F_ID='F001' AND IMP_WEIGHT>100



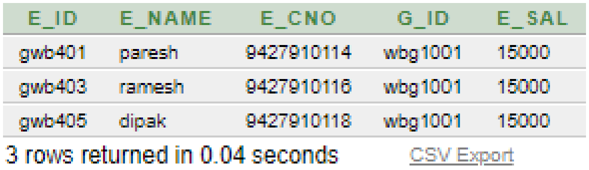
IMP_ID	G_ID	F_ID	IMP_WEIGHT	IMP_DATE
imp0002	wbg1001	f001	120	22-MAR-19
imp0004	wbg1001	f001	110	24-APR-19
imp0005	wbg1001	f001	105	25-APR-19
imp0017	wbs1001	f001	120	22-MAR-19
imp0019	wbs1001	f001	110	24-APR-19
imp0020	wbs1001	f001	105	25-APR-19
imp0032	wbr1001	f001	120	22-MAR-19
imp0034	wbr1001	f001	110	24-APR-19
imp0035	wbr1001	f001	105	25-APR-19
imp0047	wbn1001	f001	120	22-MAR-19

More than 10 rows available. Increase rows selector to view more rows.

7) SHOW THE EMPLOYEE OF PERTICULAR GODOWN

SELECT * FROM EMPLOYEE

WHERE G_ID='WBG1001'



E_ID	E_NAME	E_CNO	G_ID	E_SAL
gwb401	paresh	9427910114	wbg1001	15000
gwb403	ramesh	9427910116	wbg1001	15000
gwb405	dipak	9427910118	wbg1001	15000

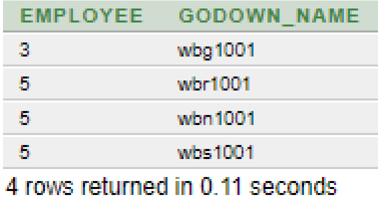
3 rows returned in 0.04 seconds CSV Export

8) SHOW THE TOTAL EMPLOYEE IN ALL GODOWN

SELECT COUNT (E_ID) AS EMPLOYEE, G_ID AS GODOWN_NAME

FROM EMPLOYEE

GROUP BY G_ID;



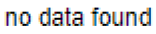
EMPLOYEE	GODOWN_NAME
3	wbg1001
5	wbr1001
5	wbn1001
5	wbs1001

4 rows returned in 0.11 seconds

9) SHOW THE EXPORT FOOD IN BETWEEN PERTICULAR DATE

SELECT F_ID FROM EXPORT

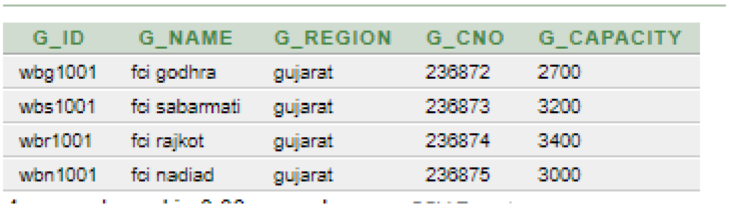
WHERE EXP_DATE BETWEEN '21-MAR-2019' AND '25-APR-2019';



no data found

10) Details of all godown

SELECT *FROM GODOWN



G_ID	G_NAME	G_REGION	G_CNO	G_CAPACITY
wbg1001	fci godhra	gujarat	236872	2700
wbs1001	fci sabarmati	gujarat	236873	3200
wbr1001	fci rajkot	gujarat	236874	3400
wbn1001	fci nadiad	gujarat	236875	3000

5.4 PL/SQL BLOCKS

5.4.1 PROCEDURES

1. CREATE A PROCEDURE TO REGISTER FOR IMPORT.

```
CREATE OR REPLACE PROCEDURE INSERT_INTO (ID IN

IMPORT.IMP_ID%TYPE,
GID IN IMPORT.G_ID%TYPE,
FID IMPORT.F_ID%TYPE,
IMPWEIGHT IMPORT.IMP_WEIGHT%TYPE,
IMPDATE IMPORT.IMP_DATE%TYPE
)
IS
BEGIN
INSERT INTO
IMPORT("IMP_ID","G_ID","F_ID","IMP_WEIGHT","IMP_DATE")VA
LUES(ID,GID,FID,IMPWEIGHT,IMPDATE);
END;
```

```
Procedure created.
```

```
DECLARE
A IMPORT.IMP_ID%TYPE;
B IMPORT.G_ID%TYPE;
C IMPORT.F_ID%TYPE;
D IMPORT.IMP_WEIGHT%TYPE;
E IMPORT.IMP_DATE%TYPE;
BEGIN
A:=:IMP_ID;
B: =:G_ID;
C: =:F_ID;
D: =:IMP_WEIGHT;
E: =:IMP_DATE;
INSERT_INTO (A, B, C, D, E);
END;
```

:IMP_ID

:G_ID

:F_ID

IMP_WEIGHT

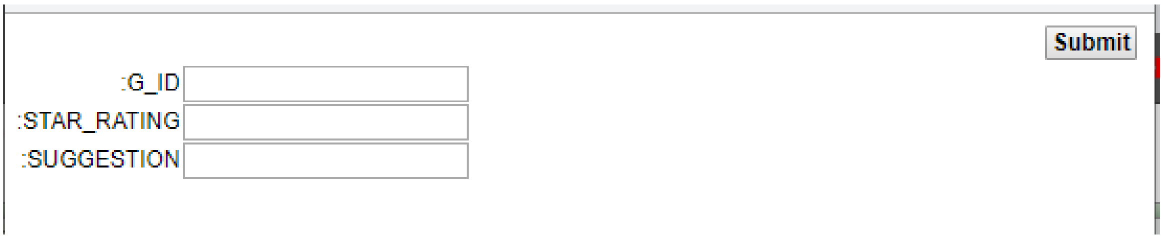
:IMP_DATE

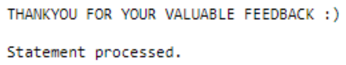
2. CREATE A PROCEDURE FOR FEEDBACK TO THE GODOWN BY GOVERNMENT OR PEOPLE

```
CREATE OR REPLACE PROCEDURE GIVE_FEEDBACK(C IN
REVIEW.G_ID%TYPE, SR IN REVIEW.STAR_RATING%TYPE, SG
IN REVIEW.SUGGESTION%TYPE)
IS
BEGIN

INSERT INTO
REVIEW ("G_ID","STAR_RATING","SUGGESTION")
VALUES(C, SR, SG);
END;
```

```
Procedure created.
```

```
DECLARE
A REVIEW.G_ID%TYPE;
B REVIEW.STAR_RATING%TYPE;
C REVIEW.SUGGESTION%TYPE;

BEGIN
A: =: G_ID;
 B: =:STAR_RATING;
C: =: SUGGESTION;
GIVE_FEEDBACK (A, B, C);
END;
```



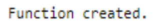


```
THANKYOU FOR YOUR VALUABLE FEEDBACK :)

Statement processed.
```

5.4.2 FUNCTIONS

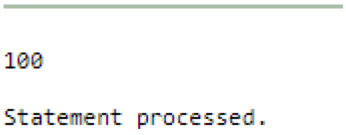
1) CREATE A FUNCTION WHICH DISPLAYS THE RETURN OF ALL AVERAGE IMPORT

```
CREATE OR REPLACE FUNCTION GETAVG RETURN INTEGER IS
AVGR INTEGER;
BEGIN
 SELECT AVG (IMP_WEIGHT) INTO AVGR FROM IMPORT;
 RETURN AVGR;
END;
```



```
Function created.
```

```
DECLARE
M IMPORT.IMP_WEIGHT%TYPE;
BEGIN
M: = GETAVG ();
DBMS_OUTPUT.PUT_LINE (M);
END;
```

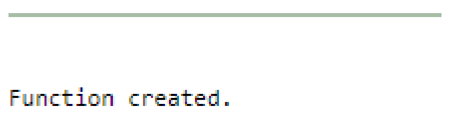


```
100

Statement processed.
```

2) FETCH THE CAPACITY OF THE GODOWN.

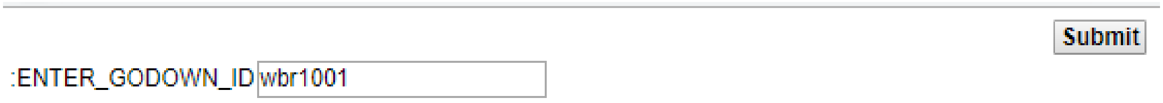
```
CREATE OR REPLACE FUNCTION GETTOTALCAPACITY (GID VARCHAR2) RETURN NUMBER IS
CAPACITY NUMBER;
KEYRET NUMBER;
```

```
BEGIN
 SELECT G_CAPACITY INTO KEYRET FROM GODOWN WHERE G_ID=GID;
 RETURN KEYRET;
```



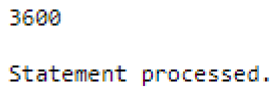
```
Function created.
```

```
DECLARE
M GODOWN.G_CAPACITY%TYPE;
A GODOWN.G_ID%TYPE;
BEGIN
A: =: G_ID;
M: =GETTOTALCAPACITY (A);
DBMS_OUTPUT.PUT_LINE (M);
END;
```



:ENTER_GODOWN_ID wbr1001

CAPACITY OF GODOWN:



```
3600

Statement processed.
```

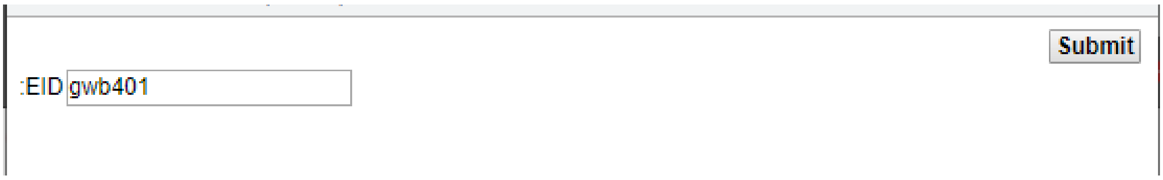
5.4.3 EXCEPTION

1. CHECK WHETHER A PARTICULAR EMPLOYEE IS WORKING OR NOT IF NOT THEN GIVE EXCEPTION

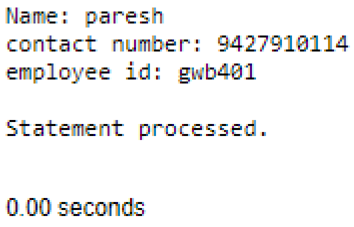
```
DECLARE
   EID EMPLOYEE.E_ID%TYPE;
   E_NAME EMPLOYEE.E_NAME%TYPE;
   E_CNO EMPLOYEE.E_CNO%TYPE;
BEGIN
EID: =: EID;
SELECT E_NAME INTO E_NAME   FROM EMPLOYEE    WHERE
EMPLOYEE.E_ID = EID;
```

```
SELECT E_CNO INTO E_CNO   FROM EMPLOYEE    WHERE
EMPLOYEE.E_ID = EID;
DBMS_OUTPUT.PUT_LINE ('NAME: '|| E_NAME);
DBMS_OUTPUT.PUT_LINE ('CONTACT NUMBER: ' || E_CNO);
DBMS_OUTPUT.PUT_LINE ('EMPLOYEE ID: ' || EID);
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE ('NO SUCH CUSTOMER!');
END;
```

OUTPUT**:**

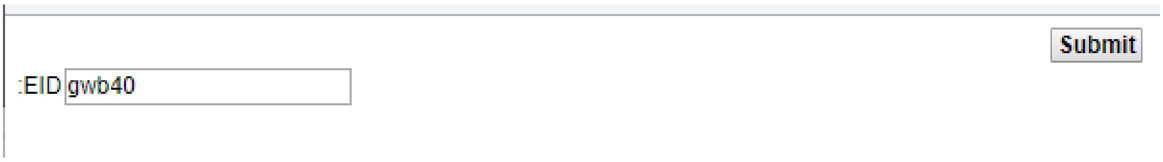


:EID gwb401

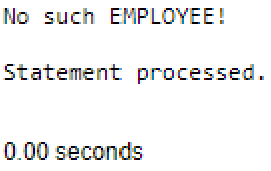


```
Name: paresh
contact number: 9427910114
employee id: gwb401

Statement processed.

0.00 seconds
```

OUPUT WITH EXCEPTION**:**



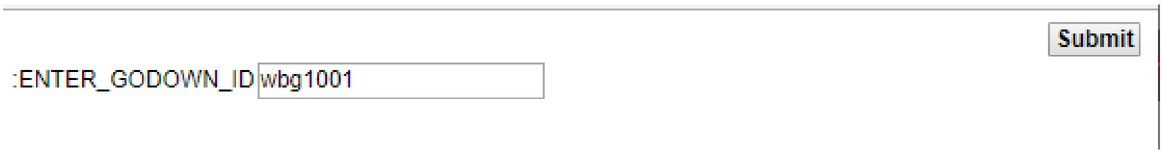
:EID gwb40



```
No such EMPLOYEE!

Statement processed.

0.00 seconds
```

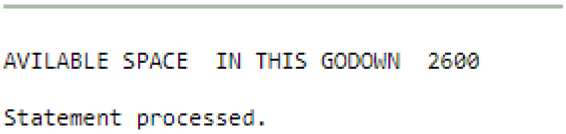
2) When food status is more than the godown capacity then it will give exception (user define)

```
DECLARE
  FID F_STATUS.F_ID%TYPE;
  GID F_STATUS.G_ID%TYPE;
  FSTS1 F_STATUS.F_STS%TYPE;
  FSTS2 F_STATUS.F_STS%TYPE;
  FSTS3 F_STATUS.F_STS%TYPE;
  A GODOWN.G_CAPACITY%TYPE;
  P VARCHAR2 (20);
  S NUMBER;
  V NUMBER;
FAIL EXCEPTION;
BEGIN
P: =:ENTER_GODOWN_ID;
SELECT G_CAPACITY INTO A FROM GODOWN WHERE G_ID=P;
SELECT   F_STS INTO   FSTS1 FROM F_STATUS WHERE F_ID ='F001' AND
G_ID=P;
SELECT   F_STS INTO   FSTS2 FROM F_STATUS WHERE F_ID ='F002' AND
G_ID=P;
SELECT F_STS INTO FSTS3 FROM F_STATUS WHERE F_ID ='F003' AND
G_ID=P;
 S: =FSTS1+FSTS2+FSTS3;

 IF A<S THEN RAISE FAIL;
 ELSE
 V: =A-S;
  DBMS_OUTPUT.PUT_LINE ('AVILABLE SPACE IN THIS GODOWN '||V);
END IF;
EXCEPTION
 WHEN FAIL THEN
  DBMS_OUTPUT.PUT_LINE ('NOT AB LE TO MORE STORE IN THIS
GODOWN');
END;
```



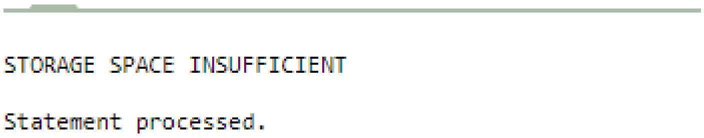
:ENTER_GODOWN_ID wbg1001

Here I enter godown ‘wbg1001’ so first it will compute total available food in godown and if it is greater than current capacity then it will give exception. If food status is less than current capacity then it will give this type of output.



```
AVILABLE SPACE  IN THIS GODOWN  2600

Statement processed.
```

If food status is greater than current capacity then it will give this type of output.



```
STORAGE SPACE INSUFFICIENT

Statement processed.
```

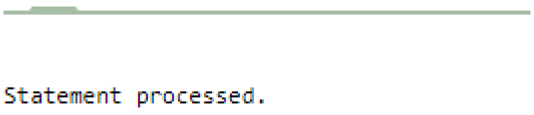
3) When there is no stock of food then it will also give exception.

(User define)

```
DECLARE
   FID F_STATUS.F_ID%TYPE;
   GID F_STATUS.G_ID%TYPE;
   FSTS1 F_STATUS.F_STS%TYPE;
   FSTS2 F_STATUS.F_STS%TYPE;
   FSTS3 F_STATUS.F_STS%TYPE;
   A GODOWN.G_CAPACITY%TYPE;
   P VARCHAR2 (20);
   S NUMBER;
FAIL EXCEPTION;
BEGIN
P: =:ENTER_GODOWN_ID;
SELECT G_CAPACITY INTO A FROM GODOWN WHERE G_ID=P;
SELECT F_STS INTO   FSTS1 FROM F_STATUS WHERE F_ID ='F001' AND
G_ID=P;
```

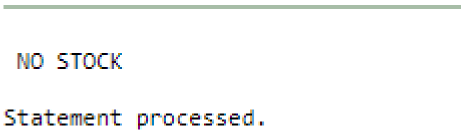
```
SELECT   F_STS INTO   FSTS2 FROM F_STATUS WHERE F_ID ='F002' AND
G_ID=P;
SELECT   F_STS INTO   FSTS3 FROM F_STATUS WHERE F_ID ='F003' AND
G_ID=P;
  S: =FSTS1+FSTS2+FSTS3;
 IF 0>S THEN RAISE FAIL;
  END IF;
EXCEPTION
  WHEN FAIL THEN
   DBMS_OUTPUT.PUT_LINE ('STORAGE SPACE INSUFFICIENT');
END;
```

When all status is ok then no exception generated.



```
Statement processed.
```

When there is no stock of food then it will give this type of error.

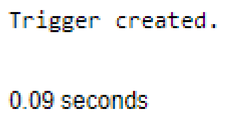


```
 NO STOCK

Statement processed.
```

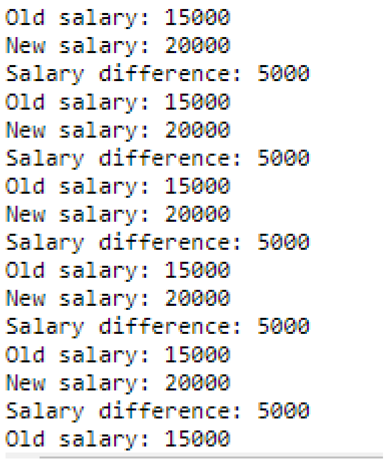
5.5 TRIGGERS

1. CREATE TRIGGER TO DISPLAY THE OLD AND NEW SALARY.

```
CREATE OR REPLACE TRIGGER DISPLAY_SALARY_CHANGES
BEFORE DELETE OR INSERT OR UPDATE ON EMPLOYEE
FOR EACH ROW
WHEN (NEW.E_SAL > 0)
DECLARE
   SAL_DIFF NUMBER;
BEGIN
   SAL_DIFF: =:NEW.E_SAL - :OLD.E_SAL;
   DBMS_OUTPUT.PUT_LINE ('OLD SALARY: ' ||:OLD.E_SAL);
   DBMS_OUTPUT.PUT_LINE ('NEW SALARY: ' ||:NEW.E_SAL);
   DBMS_OUTPUT.PUT_LINE ('SALARY DIFFERENCE: ' || SAL_DIFF);
END;
DECLARE
   TOTAL ROWS NUMBER (10);
BEGIN
   UPDATE EMPLOYEE
   SET E_SAL = E_SAL + 5000;
   IF SQL%NOTFOUND THEN
      DBMS_OUTPUT.PUT_LINE ('NO EMPLOYEES UPDATED');
   ELSIF SQL%FOUND THEN
      TOTAL_ROWS:= SQL%ROWCOUNT;
      DBMS_OUTPUT.PUT_LINE (TOTAL_ROWS || ' EMPLOYEES UPDATED ');
   END IF;
END;
```



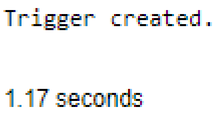
```
Trigger created.

0.09 seconds
```



```
Old salary: 15000
New salary: 20000
Salary difference: 5000
Old salary: 15000
New salary: 20000
Salary difference: 5000
Old salary: 15000
New salary: 20000
Salary difference: 5000
Old salary: 15000
New salary: 20000
Salary difference: 5000
Old salary: 15000
New salary: 20000
Salary difference: 5000
Old salary: 15000
```

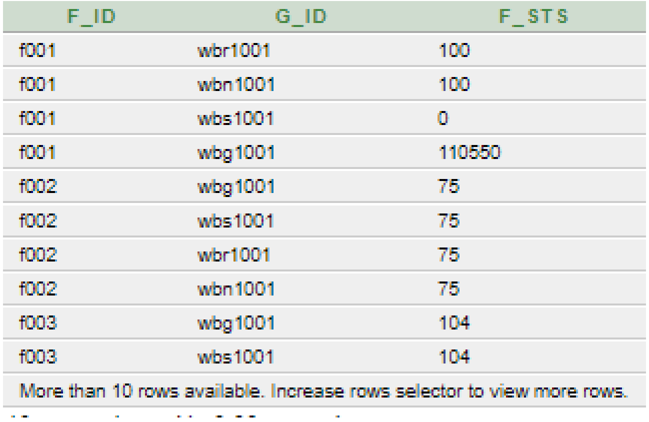
2) when export occur then update the status of godown

```
CREATE OR REPLACE
TRIGGER UPDT_AFT_DEL AFTER INSERT ON EXPORT FOR EACH ROW
BEGIN
UPDATE F_STATUS SET F_STS = F_STS-:NEW.EXP_WEIGHT WHERE: NEW.
F_ID = F_STATUS.F_ID AND: NEW. G_ID = F_STATUS.G_ID;
END;
```



```
Trigger created.

1.17 seconds
```

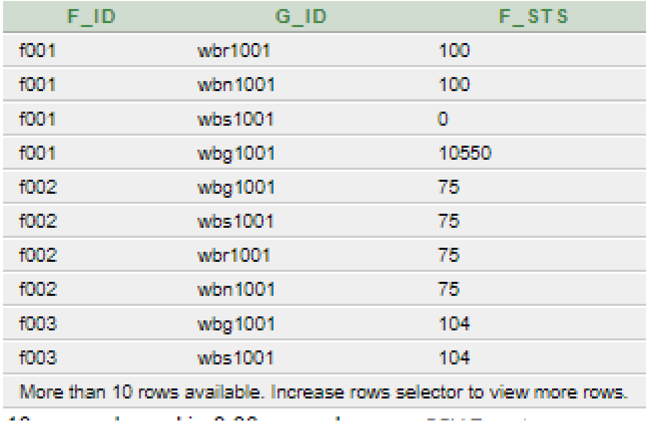
Before the trigger:



F_ID	G_ID	F_STS
f001	wbr1001	100
f001	wbn1001	100
f001	wbs1001	0
f001	wbg1001	110550
f002	wbg1001	75
f002	wbs1001	75
f002	wbr1001	75
f002	wbn1001	75
f003	wbg1001	104
f003	wbs1001	104

More than 10 rows available. Increase rows selector to view more rows.

After the trigger exporting 100000 tone from Godown=’wbg1001’ and food is ‘f001’

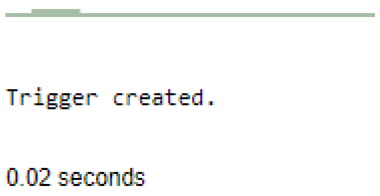


F_ID	G_ID	F_STS
f001	wbr1001	100
f001	wbn1001	100
f001	wbs1001	0
f001	wbg1001	10550
f002	wbg1001	75
f002	wbs1001	75
f002	wbr1001	75
f002	wbn1001	75
f003	wbg1001	104
f003	wbs1001	104

More than 10 rows available. Increase rows selector to view more rows.

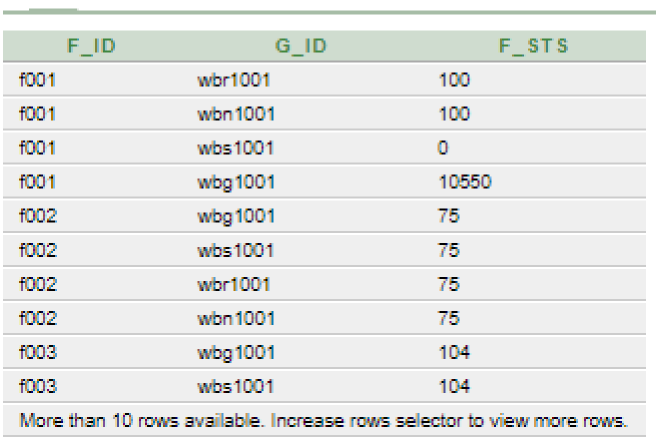
3) When import occur then update the status of godown

```
CREATE OR REPLACE
 TRIGGER UPDT_AFT_INS AFTER INSERT ON IMPORT FOR EACH ROW
BEGIN
UPDATE F_STATUS SET F_STS = F_STS+:NEW.IMP_WEIGHT WHERE:
NEW.F_ID = F_STATUS.F_ID AND: NEW.G_ID = F_STATUS.G_ID;
END;
```



```
Trigger created.

0.02 seconds
```

=>Before updating the godown (Import):

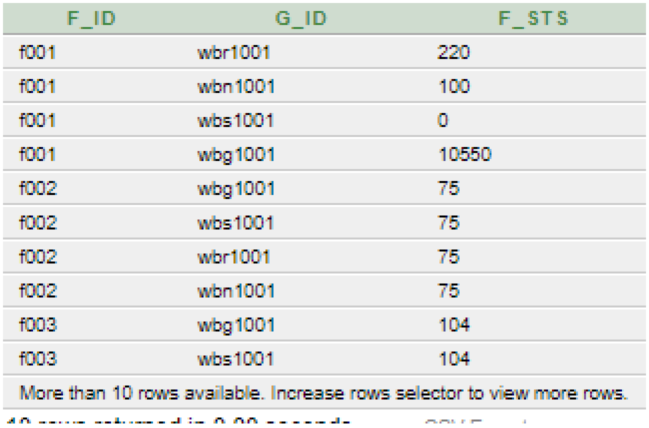


F_ID	G_ID	F_STS
f001	wbr1001	100
f001	wbn1001	100
f001	wbs1001	0
f001	wbg1001	10550
f002	wbg1001	75
f002	wbs1001	75
f002	wbr1001	75
f002	wbn1001	75
f003	wbg1001	104
f003	wbs1001	104

More than 10 rows available. Increase rows selector to view more rows.

=> After updating the godown (Import):

Import food ‘f001’ 120 tones to the godown ‘wbr1001’



F_ID	G_ID	F_STS
f001	wbr1001	220
f001	wbn1001	100
f001	wbs1001	0
f001	wbg1001	10550
f002	wbg1001	75
f002	wbs1001	75
f002	wbr1001	75
f002	wbn1001	75
f003	wbg1001	104
f003	wbs1001	104

More than 10 rows available. Increase rows selector to view more rows.

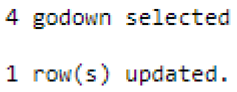
10 rows returned in 0.00 seconds CSV Export

5.6 CURSOR

5.6.1 IMPLICIT CURSOR

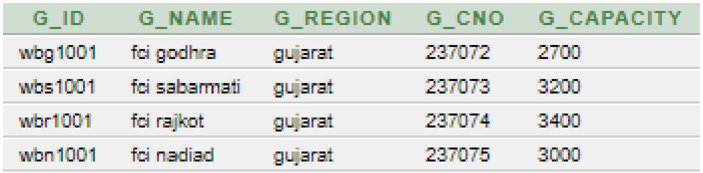
1. INCREASE THE CAPACITY OF GODOWN TO 200 MATRIC TONE.

```
DECLARE
  TOTAL_ROWS INTEGER;
BEGIN
  UPDATE GODOWN
  SET G_CAPACITY = G_CAPACITY + 200;
  IF SQL%NOTFOUND THEN
    DBMS_OUTPUT.PUT_LINE ('NO GODOWN SELECTED');
  ELSIF SQL%FOUND THEN
    TOTAL_ROWS:= SQL%ROWCOUNT;
    DBMS_OUTPUT.PUT_LINE ( TOTAL_ROWS || ' GODOWN SELECTED ');
  END IF;
END;
```



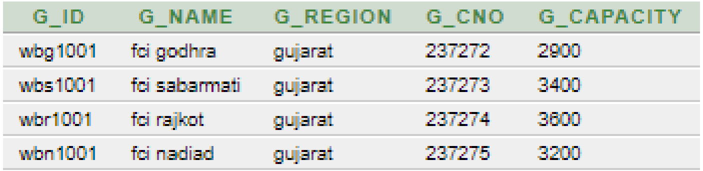
```
4 godown selected

1 row(s) updated.
```

BEFORE**:**



G_ID	G_NAME	G_REGION	G_CNO	G_CAPACITY
wbg1001	fci godhra	gujarat	237072	2700
wbs1001	fci sabarmati	gujarat	237073	3200
wbr1001	fci rajkot	gujarat	237074	3400
wbn1001	fci nadiad	gujarat	237075	3000

AFTER**:**

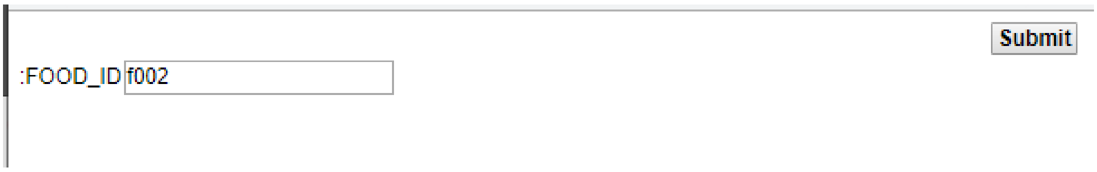


G_ID	G_NAME	G_REGION	G_CNO	G_CAPACITY
wbg1001	fci godhra	gujarat	237272	2900
wbs1001	fci sabarmati	gujarat	237273	3400
wbr1001	fci rajkot	gujarat	237274	3600
wbn1001	fci nadiad	gujarat	237275	3200

5.6.2 EXPLICIT CURSOR

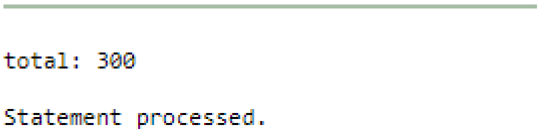
FETCH THE TOTAL FOOD OF ALL GODOWN SO WE CAN PREDECT THAT HOW MUCH FOOD IS AVILABLE (EXPLICIT CURSOR)

```
DECLARE
 CURSOR M IS SELECT F_STS,F_ID FROM F_STATUS;
 P NUMBER;
 Q VARCHAR2 (20);
BEGIN
 P: =0;
 Q: =:FOOD_ID;
 FOR I IN M
 LOOP
  IF I.F_ID=Q THEN
  P: =P + I.F_STS;
 END IF;
 END LOOP;
 DBMS_OUTPUT.PUT_LINE ('TOTAL: '|| P );
END;
```



:FOOD_ID f002

TOTAL FOOD IN ALL GODOWN:



```
total: 300

Statement processed.
```

6. FUTURE ENHANCEMENTS OF THE SYSTEM

- The System can be expanded by adding more number of godown in state/country.
- The tracking facility can be added to the system through which the any godown can track the import or export through GPS.
- The details of export and import can be more enhanced.
- The automatic generation of G_ID, E_ID and I_ID can be implemented.
- The policy of compensation of food damaged can be added.
- Frontend can be implemented

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project.

7. BIBLIOGRAPHY

Books:

- Database System Concepts
  By: - Henry F.Korth and A.Silberschatz

- PL/SQL Programming
  By: - Ivan Bayross

- The Complete Reference
  By: - George Koch

References: www.W3Schools.com/sql
References: www.Google.com