**SOUTHEAST UNIVERSITY**

****

**Department of Computer Science Engineering**

**PROJECT NAME: Mess management system**

**Course Code: CSE3012 Course Title: Database Design Lab**

**Submitted TO: Kimia Aksir**

**Designation: Lecturer**

**Submitted By: Name: Ahamadul Kabir tashfiq  
 ID: 2017000000072**

**Batch: 45th**

**Section: 06**

**Name: Md.Arif Hossain  
 ID: 2017000000004**

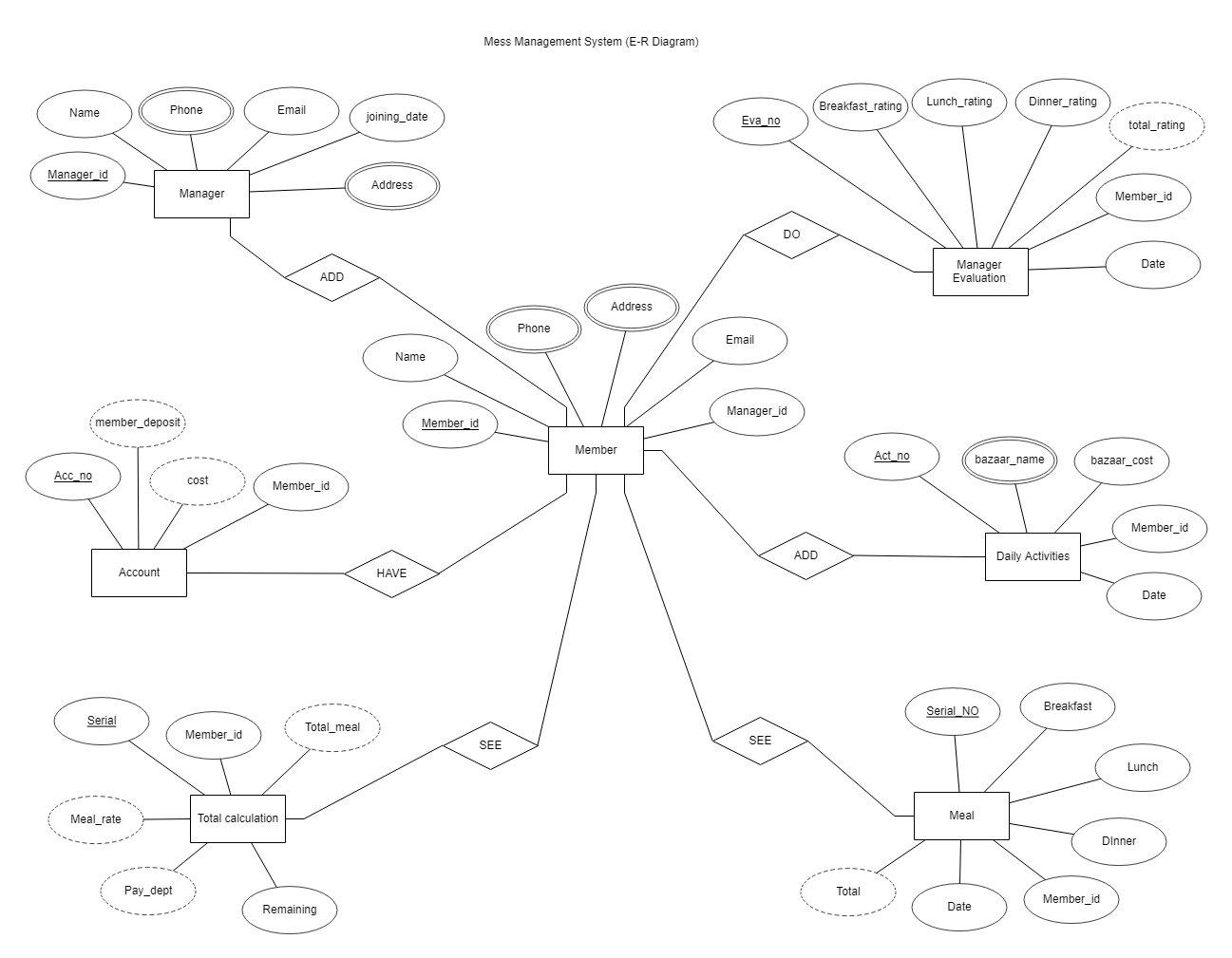
**Batch: 45th**

**Section: 06**

**Submission Date: 06-08-2019**

**Mess management system**

ER-Diagram:



Generally, most of the student passing there are student life live in the mess. To live in the mess, they have to follow some rules and regulation. The most important and biggest rule is, each of every mess member has to take the entire responsibility of the mass in individual month. He has to manage the entire ‘Mess management system’. Who manage the entire system we called him ‘Manager’. He has to manage lot of things, such as:

1. Add new member.
2. Manage account.
3. Check daily activities
4. Paying attention to meals
5. Total calculation.

A manager has to notice this thing every day. Which is really boring and painful.

So, this is the purpose that we think to create a ‘Mess Management System’. Which is make this thinks easier and interesting. In our MMS (Mess Management System) we have added all features to make it easy for a manager. We use ‘MySQL’ database to store all the data.

Now we will provide some idea of our MMS (Mess management System): -

* A manager can add multiple member in the mess.
* Member add daily activities in the system.
* Member have an account Manager handle the account and total calculation through the member.
* Member see the number of meal to corresponding the date.
* Members can evaluate the manager and manager just only can see the evaluation rate.
* Every member can just only can see the position of account, number of meal and total calculation.

This are the features we have add in our ‘Mess management System’.

**List of Entity & Attribute:**

**Creating ‘Manager’ table:**

CREATE TABLE Manager (

manager\_id varchar (20) PRIMARY key,

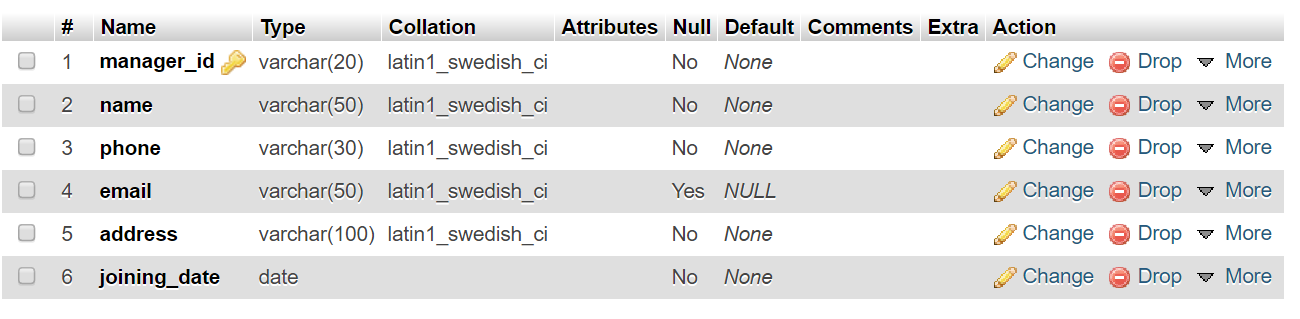
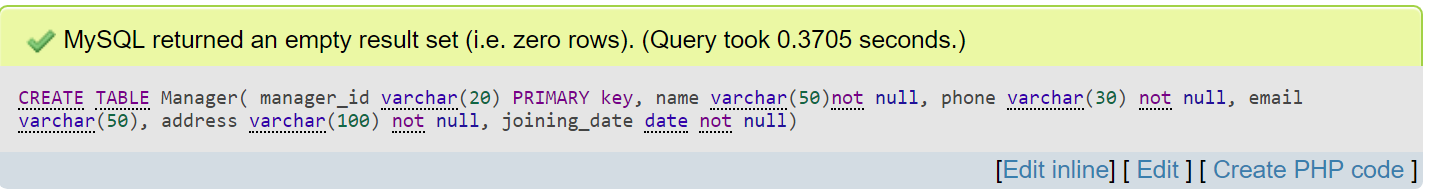
name varchar (50) not null,

phone varchar (30) not null,

email varchar (50),

address varchar (100) not null,

joining\_date date not null);



**Creating ‘Member’ table:**

CREATE TABLE member(

member\_id varchar(20) PRIMARY key,

name varchar(50) not null,

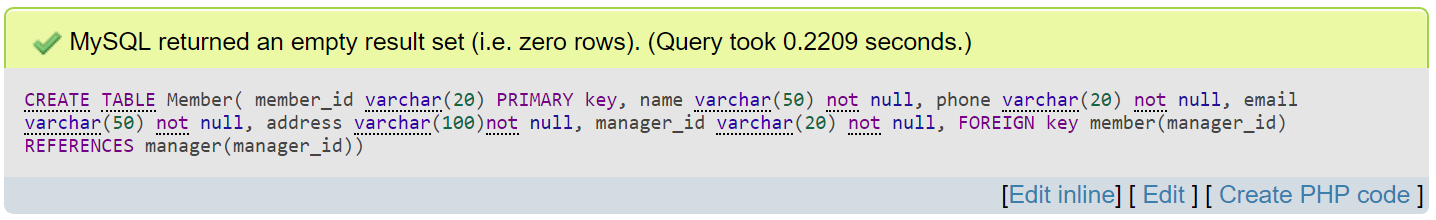
phone varchar(20) not null,

email varchar(50) not null,

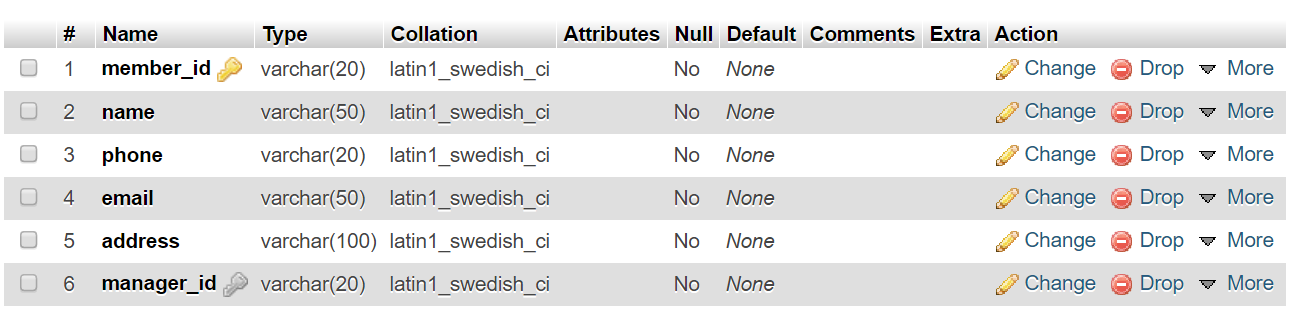
address varchar(100)not null,

manager\_id varchar(20) not null,

FOREIGN key member(manager\_id) REFERENCES manager(manager\_id)



**Creating ‘Manager Evaluation’ table:**



CREATE TABLE manager\_evaluation(

eva\_no int(20) PRIMARY key AUTO\_INCREMENT,

member\_id varchar(20) not null,

breakfast\_rating int(10)not null,

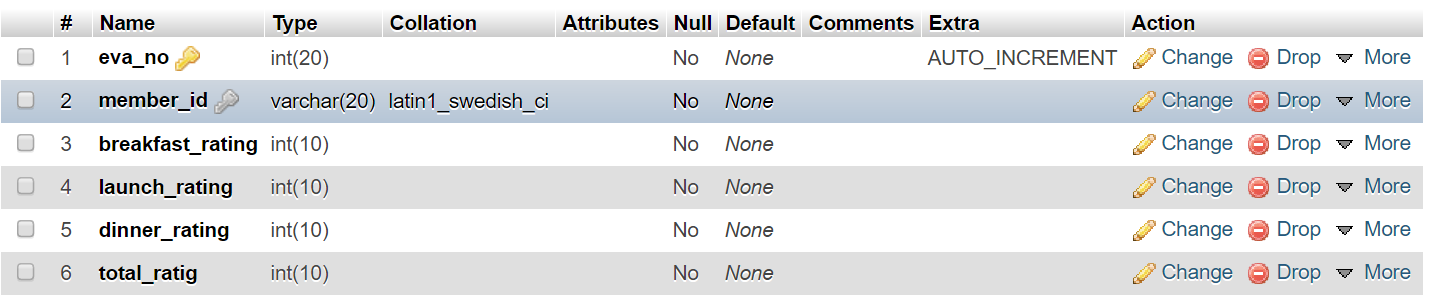
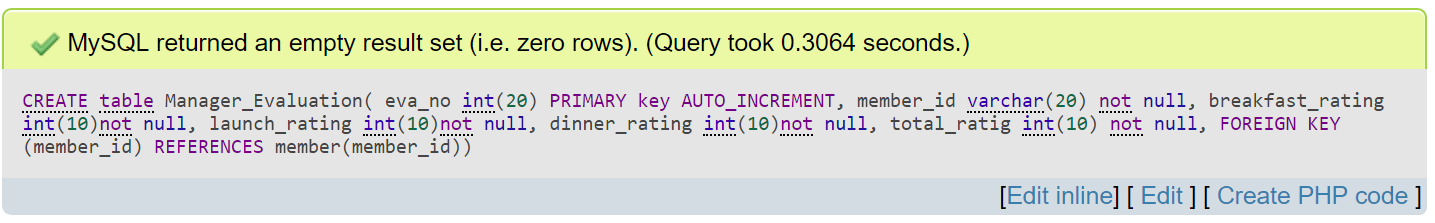
launch\_rating int(10)not null,

dinner\_rating int(10)not null,

total\_ratig int(10) not null,

FOREIGN KEY (member\_id) REFERENCES member(member\_id)

);



**Creating ‘Daily Activity’ table:**

CREATE TABLE daily\_activity (

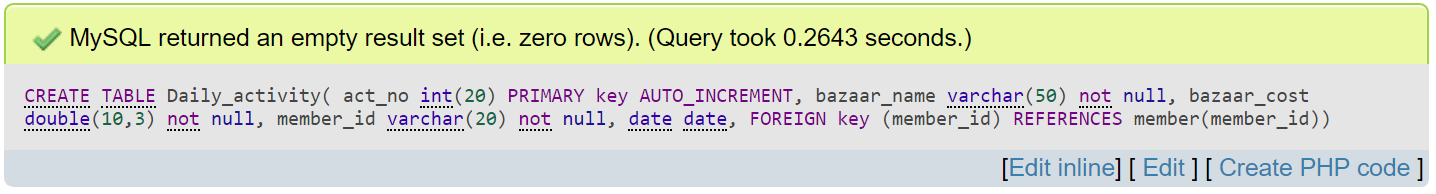
Act\_no int(20) PRIMARY key AUTO\_INCREMENT,

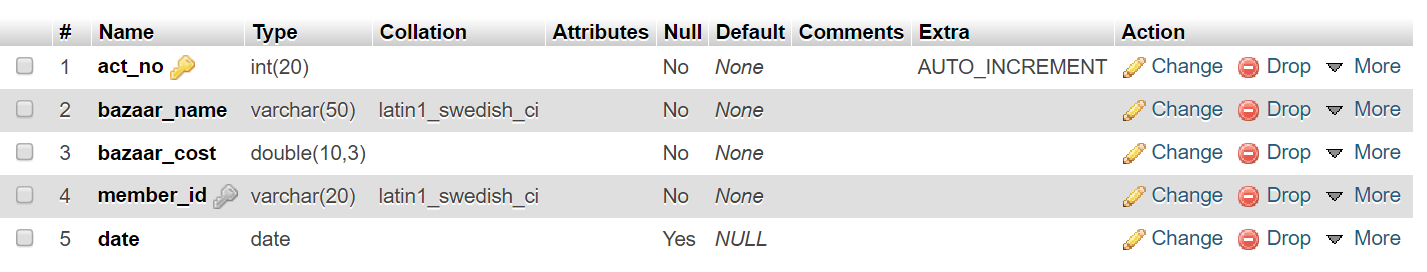
bazar\_name varchar(50) not null,

bazar\_cost double(10,3) not null,

member\_id varchar(20) not null,

FOREIGN key (member\_id) REFERENCES member(member\_id));





**Creating ‘Meal’ table:**

CREATE TABLE meal(

serial\_no int(20) PRIMARY key,

member\_id varchar(20),

breakfast int(10),

launch int(10),

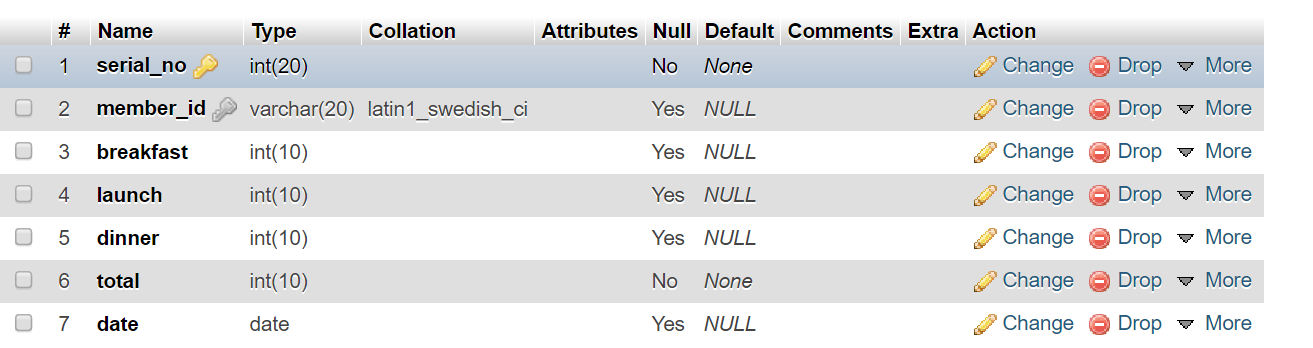
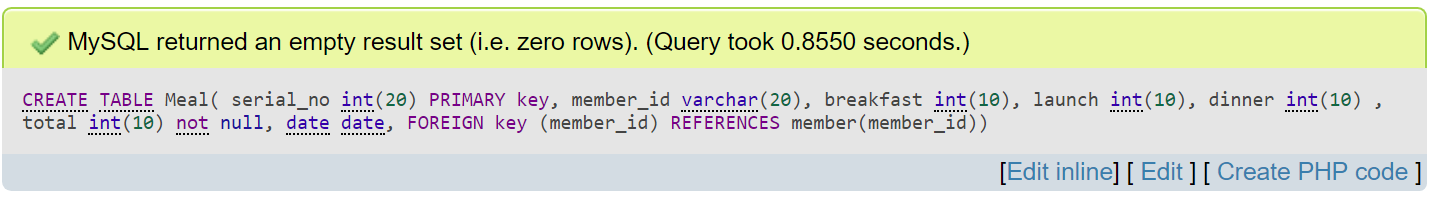
dinner int(10) ,

total int(10) not null,

date date,

FOREIGN key (member\_id) REFERENCES member(member\_id)

);



**Creating ‘Total calculation’ table:**

CREATE TABLE total\_calculation(

serial int(20) PRIMARY key AUTO\_INCREMENT,

member\_id varchar(20),

total\_meal int(10) not null,

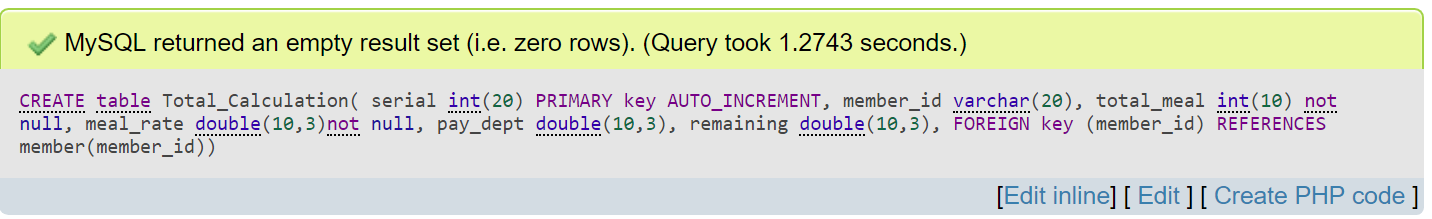
meal\_rate double(10,3)not null,

pay\_dept double(10,3),

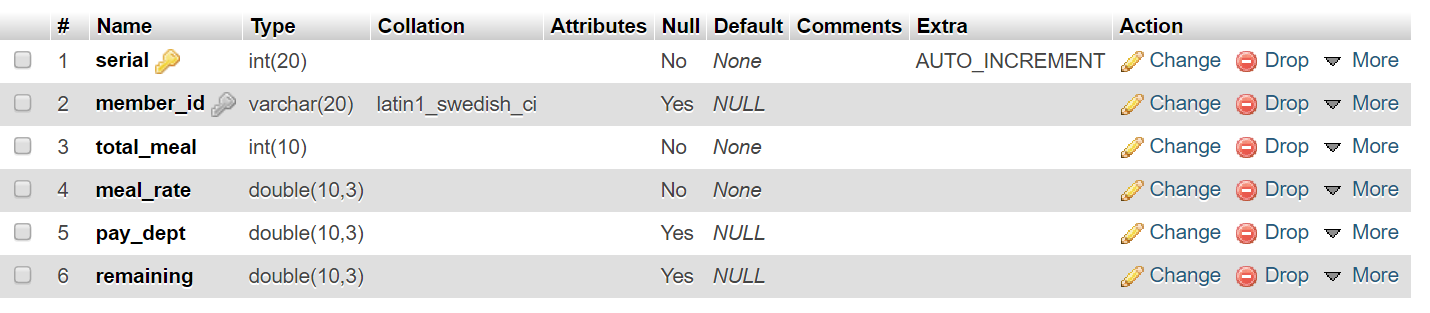
remaining double(10,3),

FOREIGN key (member\_id) REFERENCES member(member\_id)

);



**Creating ‘Account’ Table:**



CREATE TABLE account(

acc\_no varchar(20) PRIMARY key,

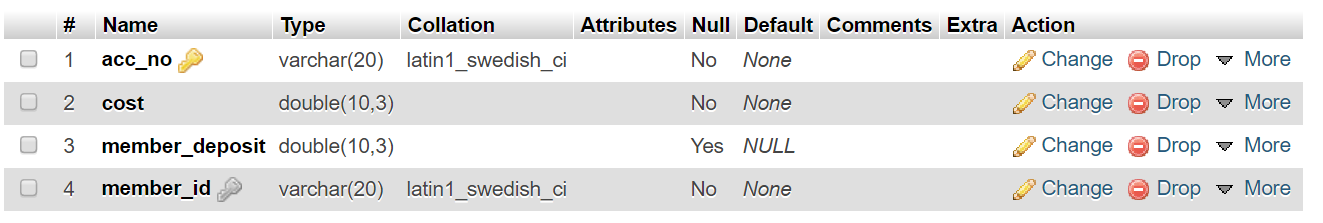
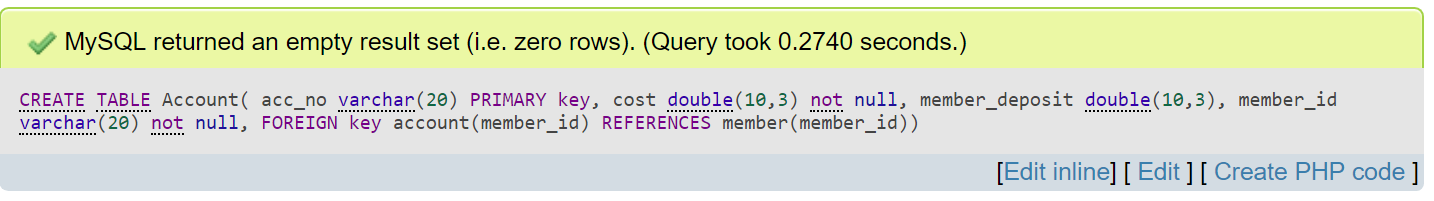
cost double(10,3) not null,

member\_deposite double(10,3),

member\_id varchar(20) not null,

FOREIGN key account(member\_id) REFERENCES member(member\_id)

);



**Creating ‘view’:**

**Create Member View:**

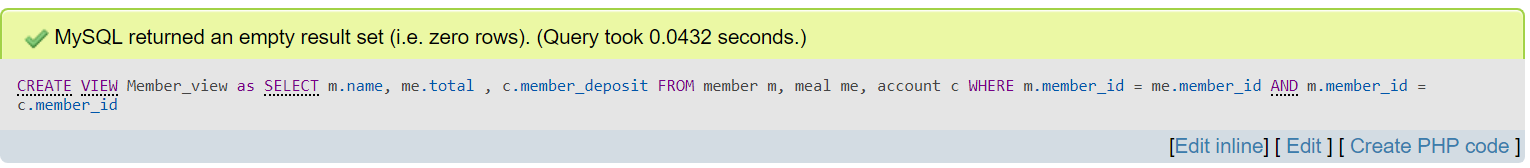
CREATE VIEW Member\_view

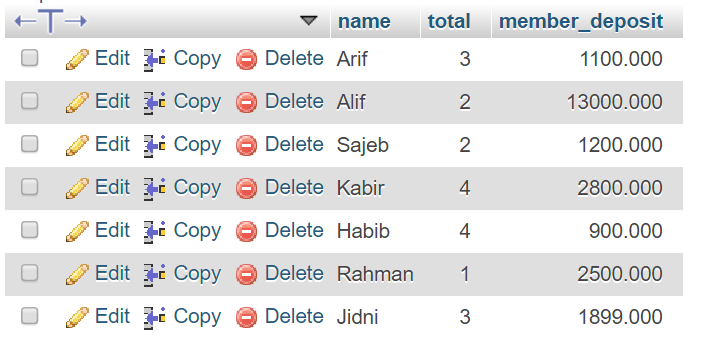
as SELECT m.name, me.total , c.member\_deposit

FROM member m, meal me, account c

WHERE m.member\_id = me.member\_id

AND m.member\_id = c.member\_id;





* **Create Member Info View:**

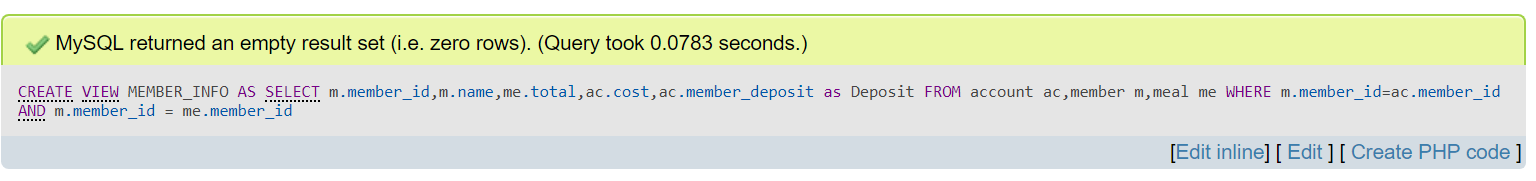
CREATE VIEW MEMBER\_INFO

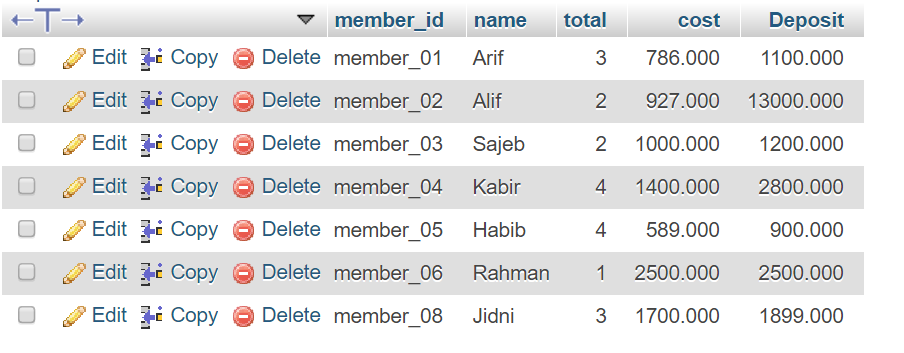
AS SELECT m.member\_id,m.name,me.total,ac.cost,

ac.member\_deposit as Deposit

FROM account ac,member m,meal me

WHERE m.member\_id=ac.member\_id AND m.member\_id = me.member\_id;





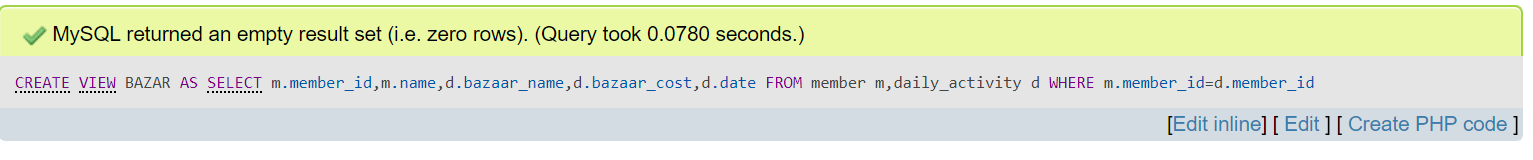
**Bazar with all information view:**

CREATE VIEW BAZAR

AS SELECT m.member\_id,m.name,d.bazaar\_name,d.bazaar\_cost,d.date

FROM member m,daily\_activity d

WHERE m.member\_id=d.member\_id;



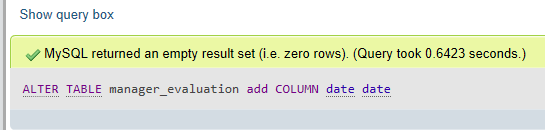


**List of Queries:**

**1.Alter column:**

ALTER TABLE manager\_evaluation

add COLUMN date date;

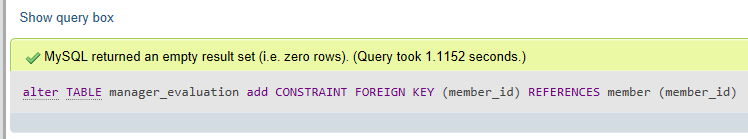


**2.Alter table:**

alter TABLE manager\_evaluation

add CONSTRAINT

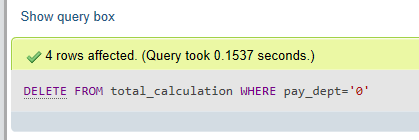
FOREIGN KEY member\_id REFERENCES member(member\_id);



**3.Delete:**

DELETE FROM total\_calculation

WHERE pay\_dept = ‘0’;



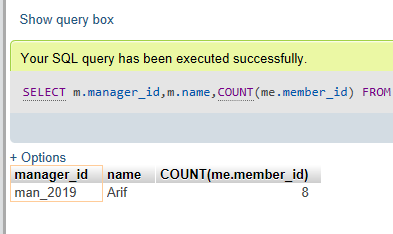
**4.Count of Joining multiTable:**

SELECT m.manager\_id,m.m\_name,COUNT(me.member\_id)

FROM member me,manager m

WHERE me.manager\_id=m.manager\_id

GROUP BY manager\_id;



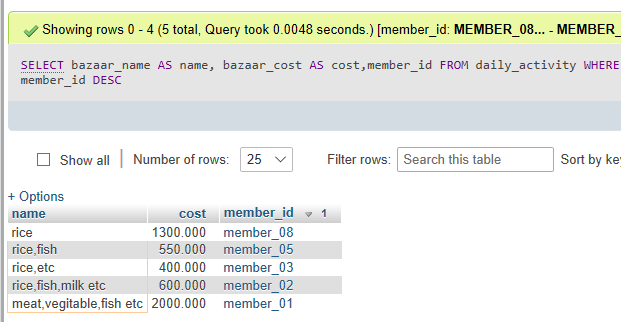
**5.** **Use of As,Like,Order by:**

SELECT bazaar\_name AS name, bazaar\_cost AS cost,member\_id

FROM daily\_activity

WHERE bazaar\_name LIKE ('rice%') or bazaar\_name LIKE('meat%')

ORDER BY member\_id DESC



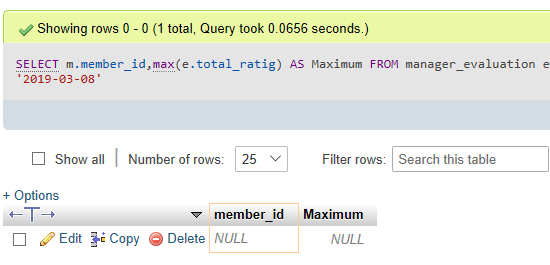
**6.Max finding of multiTable:**

SELECT m.member\_id,max(e.total\_ratig) AS Maximum

FROM manager\_evaluation e,member m

WHERE m.member\_id = e.member\_id AND

date BETWEEN '2019-03-03' and '2019-03-08';



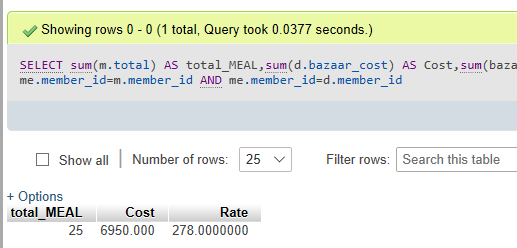
**7.Joining table:**

SELECT sum(m.total) AS total\_MEAL,sum(d.bazaar\_cost) AS Cost,sum(bazaar\_cost)/sum(total) AS Rate

FROM daily\_activity d,meal m ,member me

WHERE me.member\_id=m.member\_id AND

me.member\_id=d.member\_id;



**8.Joining table:**

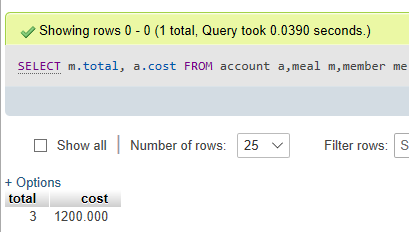
SELECT m.total, a.cost

FROM account a,meal m,member me

WHERE me.member\_id=m.member\_id AND

me.member\_id=a.member\_id AND

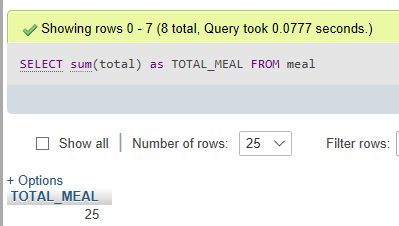
m.member\_id='member\_01';



**9.Sum:**

SELECT sum(total) as TOTAL\_MEAL

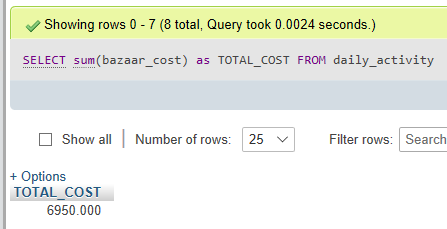
FROM meal;



**10.Sum:**

SELECT sum(bazaar\_cost) as TOTAL\_COST

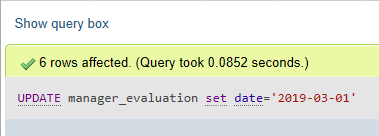
FROM daily\_activity;



**11.Update data:**

UPDATE manager\_evaluation

set date='2019-03-01';

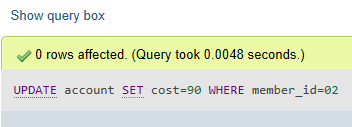


**12.Update data:**

UPDATE account

SET cost=524,member\_deposite=1000

WHERE member\_id='member\_02';

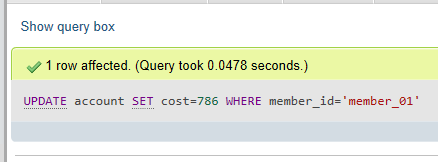


**13.Update data:**

UPDATE account

SET cost=786

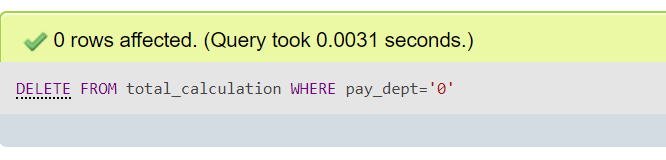
WHERE member\_id='member\_01';



**14.Delete whose dept null:**

DELETE FROM total\_calculation

WHERE pay\_dept='0';



**15.Third max rating:**

SELECT m.manager\_id AS Id,m.name AS Name,max(e.total\_ratig) as THIRD\_MAX

FROM manager\_evaluation e,manager m,member me

WHERE e.total\_ratig=(SELECT max(total\_ratig) FROM manager\_evaluation

WHERE total\_ratig<(SELECT max(total\_ratig) FROM manager\_evaluation

WHERE total\_ratig<(SELECT MAX(total\_ratig) FROM manager\_evaluation)))

AND me.manager\_id=m.manager\_id AND me.member\_id=e.member\_id;

