Database Lab Project:

Jihad Al-Binaa Organization

Done by: Fatima Hammoud

Instructor: Dr. Weal Ayoub

Spring 23- 24

Contents

I. Introduction:

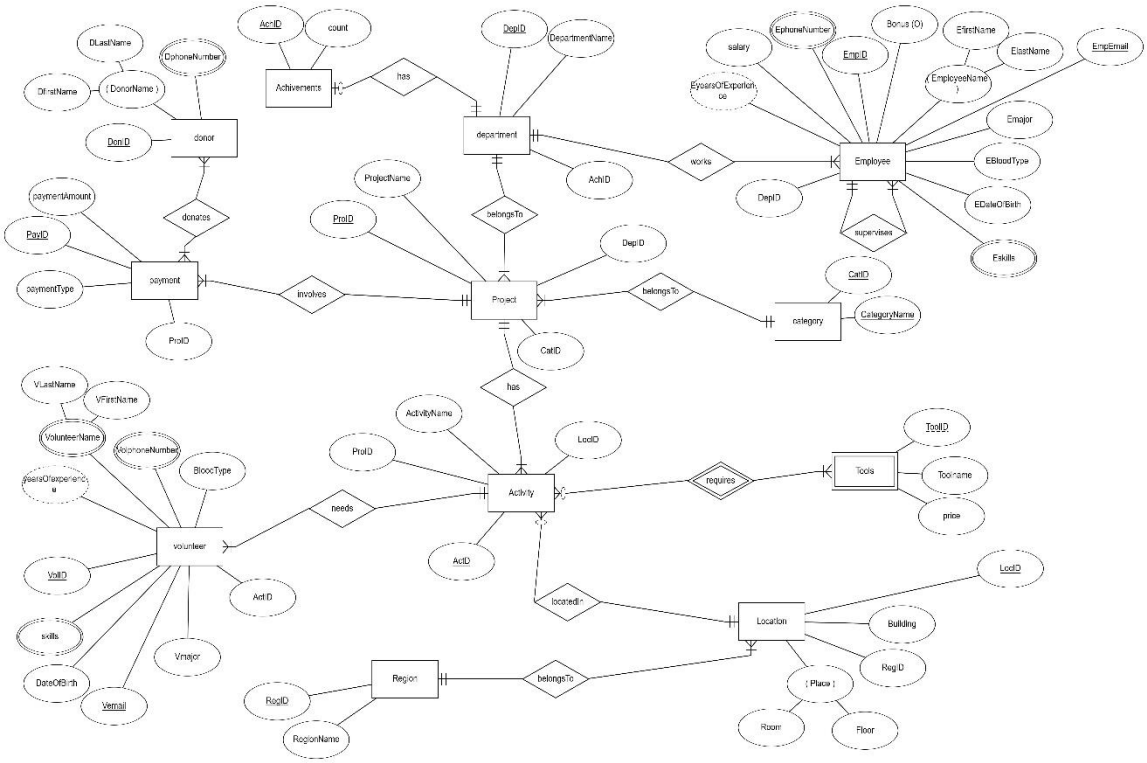
This project "Jihad Al-binaa Organization" is done by Fatima Hammoud, from the Computer Science Department at AL Maaref University. As part of our Database System lab, I have developed a comprehensive system that helps us to know the different types of projects done by this organization.

The system includes various entities such as project, employee, department, donor, payments, category, location, etc... These entities are interconnected, and their relationships are represented through an Entity-Relationship (ER) diagram. Additionally, I have implemented meaningful queries to extract valuable information from the database.

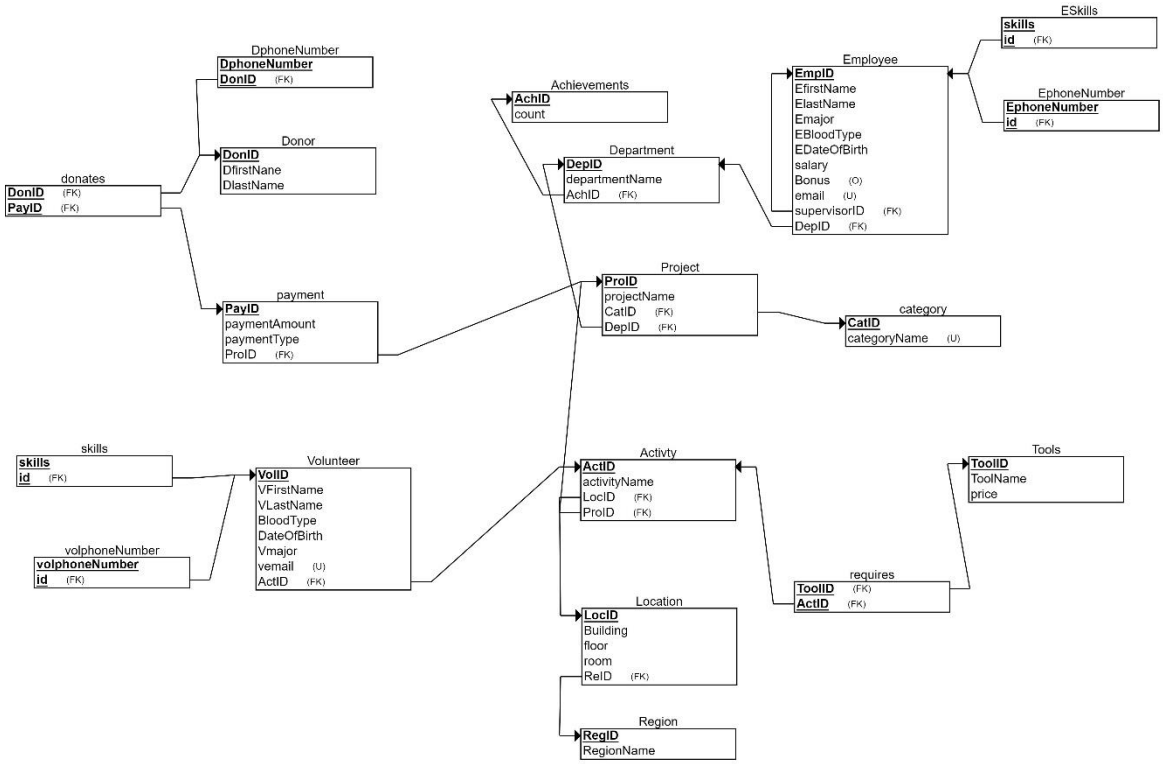
II. Main Idea:

My aim with this project is to design Jihad Al-binaa Organization System that seeks to develop society through its various programs and projects.
The organization has also built and repaired dozens of electrical plants, educational, social, health and recreational edifices, built parks and houses of worship, and worked to develop programs to support farmers, refine and enhance their skills through the establishment of agricultural development and extension centers.
After liberation in 2000, the Foundation focused on sustainable development and cooperative, environmental, voluntary and agricultural work. Building Jihad sincerely promises its dear citizens that it will spare no constructive and fruitful effort and that its workers and members will warn themselves to serve our patient people, our honorable people and our beloved homeland of Lebanon.

III. ER diagram



IV. Relational Model:



V. Implementing the relational model

```
#1.Creating the database and use it
create database if not exists JihadAlBinaa;
use JihadAlBinaa;

#2.Creating Region Table of the location where activity is done
CREATE TABLE Region
(
 RegID INT NOT NULL,
 RegionName VARCHAR(50) NOT NULL,
 PRIMARY KEY (RegID)
);

#3.Creating Location Table where activity is done
CREATE TABLE Location
(
 LocID INT NOT NULL,
 Building VARCHAR(50) NOT NULL,
 floor CHAR(5) NOT NULL,
 room INT NOT NULL,
 ReID INT NOT NULL,
 PRIMARY KEY (LocID),
 FOREIGN KEY (ReID) REFERENCES Region(RegID)
);

#4.Creating Donor Table who donates for the project
CREATE TABLE Donor
(
 DonID INT NOT NULL,
 DfirstNane VARCHAR(50) NOT NULL,
 DlastName VARCHAR(50) NOT NULL,
 PRIMARY KEY (DonID)
);
```

#5.Creating Achievements Table containing number of achievements of department

```
CREATE TABLE Achievements
(
 AchID INT NOT NULL,
 count INT NOT NULL,
 PRIMARY KEY (AchID)
);
```

#6.Creating Category Table to show the category of project

```
CREATE TABLE category
(
 CatID INT NOT NULL,
 categoryName VARCHAR(50) NOT NULL,
 PRIMARY KEY (CatID),
 UNIQUE (categoryName)
);
```

#7.Create Tools table containing the tools needed for the activity

```
CREATE TABLE Tools
(
 ToolID INT NOT NULL,
 ToolName VARCHAR(50) NOT NULL,
 price NUMERIC(5,2) NOT NULL,
 PRIMARY KEY (ToolID)
);
```

#8.Create table for the donor phone number

```
CREATE TABLE DphoneNumber
(
 DphoneNumber INT NOT NULL,
 DonID INT NOT NULL,
 PRIMARY KEY (DphoneNumber, DonID),
 FOREIGN KEY (DonID) REFERENCES Donor(DonID)
);
```

#9.Create department table where organization has different departments responsible the project according its category

```
CREATE TABLE Department
(
  DepID INT NOT NULL,
  departmentName VARCHAR(50) NOT NULL,
  AchID INT NOT NULL,
  PRIMARY KEY (DepID),
  FOREIGN KEY (AchID) REFERENCES Achievements(AchID)

);
```

#10.Create employee table containing the details of employees working in the departments

```
CREATE TABLE Employee
(
  EmpID INT NOT NULL,
  EfirstName VARCHAR(50) NOT NULL,
  ElastName VARCHAR(50) NOT NULL,
  Emajor VARCHAR(50) NOT NULL,
  EBloodType VARCHAR(5) NOT NULL,
  EDateOfBirth DATE NOT NULL,
  salary NUMERIC(5,2) NOT NULL,
  Bonus INT,
  email VARCHAR(50) NOT NULL,
  supervisorID INT NOT NULL,
  depId int not null,
  PRIMARY KEY (EmpID),
  foreign key (depId) references department (depId),
  FOREIGN KEY (supervisorID) REFERENCES Employee(EmpID),
  UNIQUE (email)
);
```

#11.Create skills table for the employee

```
CREATE TABLE ESkills
```

```
(
 skills VARCHAR(50) NOT NULL,
 id INT NOT NULL,
 PRIMARY KEY (skills, id),
 FOREIGN KEY (id) REFERENCES Employee(EmpID)
);

#12.Create table for the employee phone number
CREATE TABLE EphoneNumber
(
 EphoneNumber CHAR(8) NOT NULL,
 id INT NOT NULL,
 PRIMARY KEY (EphoneNumber, id),
 FOREIGN KEY (id) REFERENCES Employee(EmpID)
);


#13.Create project table containing the category and department of
each project
CREATE TABLE Project
(
 ProID INT NOT NULL,
 projectName VARCHAR(50) NOT NULL,
 CatID INT NOT NULL,
 DepID INT NOT NULL,
 PRIMARY KEY (ProID),
 FOREIGN KEY (CatID) REFERENCES category(CatID),
 FOREIGN KEY (DepID) REFERENCES Department(DepID)
);

#14.Create activity table needed for the project
CREATE TABLE Activty
(
 ActID INT NOT NULL,
 activityName VARCHAR(50) NOT NULL,
 LocID INT NOT NULL,
 projId int not null,
```

```
 PRIMARY KEY (ActID),
 FOREIGN KEY (LocID) REFERENCES Location(LocID),
 foreign key (projId) references project(proId)
 );
```

#15.Create table requires that relates tools table with activity table

```
CREATE TABLE requires
(
 ToolID INT NOT NULL,
 ActID INT NOT NULL,
 PRIMARY KEY (ToolID, ActID),
 FOREIGN KEY (ToolID) REFERENCES Tools(ToolID),
 FOREIGN KEY (ActID) REFERENCES Activty(ActID)
);
```

#16.Create volunteer table who volunteers in the project through its activities

```
CREATE TABLE Volunteer
(
 VolID INT NOT NULL,
 VFirstName VARCHAR(50) NOT NULL,
 VLastName VARCHAR(50) NOT NULL,
 BloodType VARCHAR(5) NOT NULL,
 DateOfBirth DATE NOT NULL,
 Vmajor VARCHAR(50) NOT NULL,
 vemail VARCHAR(50) NOT NULL,
 ActID INT NOT NULL,
 PRIMARY KEY (VolID),
 FOREIGN KEY (ActID) REFERENCES Activty(ActID),
 UNIQUE (vemail)
);
```

#17.Create skills table for volunteer

```
CREATE TABLE skills
(
 skills VARCHAR(50) NOT NULL,
 id INT NOT NULL,
```

```
 PRIMARY KEY (skills, id),
 FOREIGN KEY (id) REFERENCES Volunteer(VolID)
);
```

#18.Create table for volunteer phone number

```
CREATE TABLE volphoneNumber
(
 volphoneNumber CHAR(8) NOT NULL,
 id INT NOT NULL,
 PRIMARY KEY (volphoneNumber, id),
 FOREIGN KEY (id) REFERENCES Volunteer(VolID)
);
```

#19.Create payment table containing the payments needed for projects

```
CREATE TABLE payment
(
 PayID INT NOT NULL,
 paymentAmount NUMERIC(5,2) NOT NULL,
 paymentType VARCHAR(50) NOT NULL,
 ProID INT NOT NULL,
 PRIMARY KEY (PayID),
 FOREIGN KEY (ProID) REFERENCES Project(ProID)
);
```

#20.Create donates table relating the donor by the payments

```
Create table donates
(
donId int not null,
payId int not null,
primary key(donId,payId),
foreign key(donId) references donor (donId),
foreign key(payId) references payment (payId)
);
```

#	Time	Action	Message	Duration / Fetch
2	16:47:16	create database if not exists JihadAlBinaa	1 row(s) affected	0.078 sec
3	16:47:16	use JihadAlBinaa	0 row(s) affected	0.000 sec
4	16:47:16	CREATE TABLE Region (RegID INT NOT NULL, RegionName VARCHAR(50) NO...	0 row(s) affected	0.594 sec
5	16:47:17	CREATE TABLE Location (LocID INT NOT NULL, Building VARCHAR(50) NOT N...	0 row(s) affected	0.109 sec
6	16:47:17	CREATE TABLE Donor (DonID INT NOT NULL, DfirstNane VARCHAR(50) NOT N...	0 row(s) affected	0.047 sec
7	16:47:17	CREATE TABLE Achievements (AchID INT NOT NULL, count INT NOT NULL, ...	0 row(s) affected	0.015 sec
8	16:47:17	CREATE TABLE category (CatID INT NOT NULL, categoryName VARCHAR(50) ...	0 row(s) affected	0.047 sec
9	16:47:17	CREATE TABLE Tools (ToolID INT NOT NULL, ToolName VARCHAR(50) NOT N...	0 row(s) affected	0.016 sec
10	16:47:17	CREATE TABLE DphoneNumber (DphoneNumber INT NOT NULL, DonID INT N...	0 row(s) affected	0.031 sec
11	16:47:17	CREATE TABLE Department (DepID INT NOT NULL, departmentName VARCHA...	0 row(s) affected	0.063 sec
12	16:47:17	CREATE TABLE Employee (EmpID INT NOT NULL, EfirstName VARCHAR(50) N...	0 row(s) affected	0.062 sec
13	16:47:17	CREATE TABLE ESkills (skills VARCHAR(50) NOT NULL, id INT NOT NULL, PR...	0 row(s) affected	0.062 sec
14	16:47:17	CREATE TABLE EphoneNumber (EphoneNumber CHAR(8) NOT NULL, id INT NO...	0 row(s) affected	0.047 sec
15	16:47:17	CREATE TABLE Project (ProID INT NOT NULL, projectName VARCHAR(50) NOT...	0 row(s) affected	0.063 sec
16	16:47:17	CREATE TABLE Activty (ActID INT NOT NULL, activityName VARCHAR(50) NOT...	0 row(s) affected	0.078 sec
17	16:47:17	CREATE TABLE requires (ToolID INT NOT NULL, ActID INT NOT NULL, PRIMA...	0 row(s) affected	0.078 sec
18	16:47:17	CREATE TABLE Volunteer (VolID INT NOT NULL, VFirstName VARCHAR(50) NO...	0 row(s) affected	0.062 sec
19	16:47:17	CREATE TABLE skills (skills VARCHAR(50) NOT NULL, id INT NOT NULL, PRI...	0 row(s) affected	0.063 sec
20	16:47:17	CREATE TABLE volphoneNumber (volphoneNumber CHAR(8) NOT NULL, id INT ...	0 row(s) affected	0.047 sec
21	16:47:17	CREATE TABLE payment (PayID INT NOT NULL, paymentAmount NUMERIC(5,2)...	0 row(s) affected	0.062 sec

#Part 2

VI. Select, Insert, Update delete

#Insert into achievement table

```
INSERT INTO achievements
VALUES (1,1),(2,2),(3,3),(4,4),(5,5);
```

23	16:37:06	INSERT INTO achievements VALUES (1,1),(2,2),(3,3),(4,4),(5,5)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec

#Insert into department table

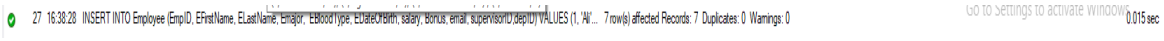
```
Insert into department
values
(1,'education' , 2 ), (2,'agriculture' , 5), (3,'construction' , 5) , (4,'health'
, 3);
```

25	16:37:27	Insert into department values (1,'education' , 2), (2,'agriculture' , 5), (3,'construction' , 5) , (4,'health' , 3)	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.000 sec

#Insert into employee table

```
INSERT INTO Employee (EmpID, EFirstName, ELastName, Emajor,
EBloodType, EDateOfBirth, salary, Bonus, email, supervisorID,depID)
VALUES
```

(1, 'Ali', 'Saad', 'engineer', 'O+', '2000-08-22', 500, 10.0,
'alisaad@gmail.com', 1,3),
(2, 'Fatima', 'Hammmoud', 'engineer', 'O-', '2000-05-14', 500, 10.0,
'fatimahammoud@gmail.com', 1,3),
(3, 'Mohammad', 'Kenaan', 'docotor', 'AB', '2000-10-26', 400, 10.0,
'ahmadkenaan@gmail.com', 3, 4),
(4, 'Ahmad', 'Kenaan', 'nurse', 'AB', '2000-10-26', 400, 10.0,
'mohammadkenaan@gmail.com', 3, 4),
(5, 'Nour', 'Zaraket', 'teacher', 'B', '2001-11-12', 250, 10.0,
'nourzaraket@gmail.com', 5 ,1),
(6, 'Nour', 'hmede', 'teacher', 'B', '2001-5-11', 250, 10.0,
'nourhmede@gmail.com', 5 ,1),
(7, 'Hassan', 'Rida', 'farmer', 'AB', '1999-12-28', 100, 10.0,
'hassanrida@gmail.com', 5 ,2),
(8, 'Jawad', 'Abbas', 'farmer', 'AB', '1999-12-10', 100, 10.0,
'jawadabbas@gmail.com', 5 ,2);



#Showng the info of each table

select * from achievements;

AchID	count
1	1
2	2
3	3
4	4
5	5
NULL	NULL

select * from department;

DepID	departmentName	AchID
1	education	2
2	agriculture	5
3	construction	5
4	health	3
NULL	NULL	NULL

```
select * from employee;
```

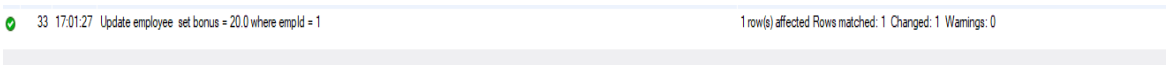
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

EmpID	EfirstName	ElastName	Emajor	EBloodType	EDateOfBirth	salary	Bonus	email	supervisorID	depId
1	Ali	Saad	engineer	O+	2000-08-22	500.00	10	alisaad@gmail.com	1	3
2	Fatima	Hammmoud	engineer	O-	2000-05-14	500.00	10	fatimahammoud@gmail.com	1	3
3	Mohammad	Kenaan	docotor	AB	2000-10-26	400.00	10	ahmadkenaan@gmail.com	3	4
4	Ahmad	Kenaan	nurse	AB	2000-10-26	400.00	10	mohammadkenaan@gmail.com	3	4
5	Nour	Zaraket	teacher	B	2001-11-12	250.00	10	nourzaraket@gmail.com	5	1
6	Nour	hmede	teacher	B	2001-05-11	250.00	10	nourhmede@gmail.com	5	1
7	Hassan	Rida	farmer	AB	1999-12-28	100.00	10	hassanrida@gmail.com	5	2
8	Jawad	Abbas	farmer	AB	1999-12-10	100.00	10	jawadabbas@gmail.com	5	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#Updating tables

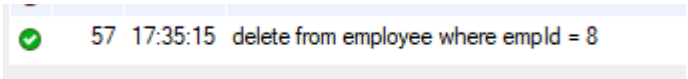
```
Update employee
set salary = 500
where empId = 3;
```

```
Update employee
set bonus = 20.0
where empId = 1;
```



#deleting record

```
delete from employee
where empId = 8;
```



VII. Subqueries

1#Select employee from education department

```
select * from employee
where depId = (select depId from department where departmentName
= 'education');
```

EmpID	EfirstName	ElastName	Emajor	EBloodType	EDateOfBirth	salary	Bonus	email	supervisorID	depId
5	Nour	Zaraket	teacher	B	2001-11-12	250.00	10	nourzaraket@gmail.com	5	1
6	Nour	hmede	teacher	B	2001-05-11	250.00	10	nourhmede@gmail.com	5	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2#Select department whose achievement is greater than 4

```
select * from department
where achId in (select achId from achievements where count > 4);
```

DepID	departmentName	AchID
2	agriculture	5
3	construction	5
NULL	NULL	NULL

3#Select employee whose supervisor's name contains letter m

```
select EfirstName,ElastName from employee
where supervisorId in (select supervisorId from employee group by
supervisorId having EfirstName like '%m%' or ElastName like '%m%');
```

EfirstName	ElastName
Fatima	Hammmoud
Mohammad	Kenaan
Ahmad	Kenaan
Nour	hmede

4#select department whose achievements is the highest

```
select departmentName from department
where achId in (select achId from achievements where count = (select
max(count) from achievements ));
```

departmentName
agriculture
construction

5#select the 2 highest salary for supervisors borning in 2000

```
select salary from employee
```

```
where supervisorId in (select supervisorId from employee where
EdateOfBirth like '%2000%' )
group by salary
order by salary desc limit 2;
```

	salary
▶	500.00
	400.00

VIII. Join queries

#inserting values

```
insert into department values ( 5,'enviromental' , 4);
select * from department;

insert into category
 values
 (1,'educational' ),
 (2, 'agricultural'),
 (3, 'constructional' ),
 (4,'health'),
 (5,'environmental');


insert into project values
(1,'Support_sessions_for_official_examination_students',1,1),
(2,'Restoration_of_poor_people''s_home',3,3),
(3,'cleanliness_campaign',5,5);

insert into donor values
(1, 'Ali','Jomaa'),
(2,'Hanaa','Mkahal'),
(3,'Kassem','abdalla');

insert into payment
values
(1,100.0 ,'cash',1),
```

```
 (2,500.0,'installment',2),
(3,400.0,'cash',3),
(4,200.0,'cash',3),
(5,600.0,'installment',2);

insert into donates values
(1,1),
(1,2),
(2,3),
(2,5),
(3,3),
(3,4);

select * from donates;


insert into tools
values
(1, 'pencil', 5),
(2, 'book', 10),
(3,'board_pen', 10),
(4,'garden_fork', 30),
(5, 'watering_can' , 40),
(6, 'trowel' , 10),
(7, 'building_stones',100),
(8,'wheelbarrow' , 100),
(9, 'beach_cleaner', 500);

insert into region values
(1, 'Nabateye'),
(2,'saida'),
(3,'beirut');

insert into location values
(1, 'municipality' , 1,1,1 ),
(2,'beach',0,0,2),
(3,'hay_solem_street',0,0,3);
```

```
insert into activty
values
(1, 'extra_exercises_solving' , 1,1),
(2,'cleaning_beach',2,3),
(3,'cleaning_street',3,3),
(4,'home_renovation',3,2);

insert into requires values
(1,1),(2,1),(3,1),(9,2),(4,3),(5,3),(8,4),(6,4);

select * from tools;
select * from activty;
select*  from requires;
```

✓	36	16:47:18	insert into department values (5,'enviromental' , 4)
✓	37	16:47:18	select * from department LIMIT 0, 1000
✓	38	16:47:18	insert into category values (1,'educational'), (2, 'agricultural'), (3, 'constructional'), (4,'healt...
✓	39	16:47:18	insert into project values (1,'Support_sessions_for_official_examination_students',1,1), (2,'Rest...
✓	40	16:47:18	insert into donor values (1, 'Ali','Jomaa'), (2,'Hanaa','Mkahal'), (3,'Kassem','abdalla')
✓	41	16:47:18	insert into payment values (1,100.0 ,'cash',1), (2,500.0,'installment',2), (3,400.0,'cash',3), (4,200...
✓	42	16:47:18	insert into donates values (1,1), (1,2), (2,3), (2,5), (3,3), (3,4)
✓	43	16:47:18	select * from donates LIMIT 0, 1000
✓	44	16:47:18	insert into tools values (1, 'pencil', 5), (2, 'book', 10), (3,'board_pen', 10), (4,'garden_fork', 30), ...
✓	45	16:47:18	insert into region values (1, 'Nabateye'), (2,'saida'), (3,'beirut')
✓	46	16:47:18	insert into location values (1, 'municipality' , 1,1,1), (2,'beach',0,0,2), (3,'hay_solem_street',0,0,3)
✓	47	16:47:18	insert into activty values (1, 'extra_exercises_solving' , 1,1), (2,'cleaning_beach',2,3), (3,'cleani...
✓	48	16:47:18	insert into requires values (1,1),(2,1),(3,1),(9,2),(4,3),(5,3),(8,4),(6,4)
✓	49	16:47:18	select * from tools LIMIT 0, 1000
✓	50	16:47:18	select * from activty LIMIT 0, 1000
✓	51	16:47:18	select* from requires LIMIT 0, 1000

1#select activities done in beirut

```
select * from activty as a
inner join location as l
on a.locId = l.locId
inner join region r
on l.reId = r.regId
```

```
where r.regionName = 'beirut';
```

ActID	activityName	LocID	projId	LocID	Building	floor	room	ReID	RegID	RegionName
3	cleaning_street	3	3	3	hay_solem_street	0	0	3	3	beirut
4	home_renovation	3	2	3	hay_solem_street	0	0	3	3	beirut

2#Select all tools needed for extra_exercises_solving as an activity

```
select toolName from tools as t
inner join requires as r
on r.toolId = t.toolId
inner join activty as a
on a.actId = r.actId
where activityName = 'extra_exercises_solving';
```

toolName
pencil
book
board_pen

3#Select all tools with the activity needing them(if any)

```
select t.toolName, a.activityName from tools as t
left join requires as r
on r.toolId = t.toolId
left join activty as a
on a.actId = r.actId;
```

toolName	activityName
pencil	extra_exercises_solving
book	extra_exercises_solving
board_pen	extra_exercises_solving
garden_fork	cleaning_street
watering_can	cleaning_street
trowel	home_renovation
building_stones	NULL
wheelbarrow	home_renovation
beach_cleaner	cleaning_beach

4#Select all departments in the organization with names of employee working in (if any)

```
select d.departmentName, e.EfirstName , e.ELastName from
department as d
left join employee as e
on d.depId = e.depId;
```

departmentName	EfirstName	ELastName
education	Nour	hmede
education	Nour	Zaraket
agriculture	Hassan	Rida
construction	Fatima	Hammmoud
construction	Ali	Saad
health	Ahmad	Kenaan
health	Mohammad	Kenaan
enviromental	NULL	NULL

5#Select the average of price of tools used in all activities done in the organization

```
select avg(price) from tools as t
inner join requires as r
on r.toolId = t.toolId
inner join activty as a
on a.actId = r.actId;
```

	avg(price)
▶	88.125000

IX. Views

1#Create a view called projectsActivityDetails showing all activities and tools needed done for cleanliness_campaign project.

```
Create view projectsActivityDetails as
select p.projectName, a.activityName, t.toolName ,t.price
from project p
inner join activty as a
on p.proId = a.projId
inner join requires r
on r.actId = a.actId
inner Join tools t
on r.toolId = t.toolId;

select * from projectsActivityDetails;
```

projectName	activityName	toolName	price
Support_sessions_for_official_examination_stu...	extra_exercises_solving	pencil	5.00
Support_sessions_for_official_examination_stu...	extra_exercises_solving	book	10.00
Support_sessions_for_official_examination_stu...	extra_exercises_solving	board_pen	10.00
Restoration_of_poor_people's_home	home_renovation	trowel	10.00
Restoration_of_poor_people's_home	home_renovation	wheelbarrow	100.00
cleanliness_campaign	cleaning_beach	beach_cleaner	500.00
cleanliness_campaign	cleaning_street	garden_fork	30.00
cleanliness_campaign	cleaning_street	watering_can	40.00

2#Create view called paymentsOfproject to show the payments of each project

```
create view paymentsOfprojectt as
select sum(paymentAmount) as s, p.projectName from payment pay
inner join project p
on pay.proId = p.proId
group by projectName;

select * from paymentsOfprojectt;
```

	sum(paymentAmount)	projectName
▶	100.00	Support_sessions_for_official_examination_stu...
	1100.00	Restoration_of_poor_people's_home
	600.00	cleanliness_campaign

3#Create a view to show the payments of the donor with the name of project

```
create view donorPayment as
select d.*, pay.paymentAmount, p.projectName from donor d
inner join donates as don
on don.donId = d.doniD
inner join payment as pay
on pay.payId = don.payId
inner join project as p
on p.proId = pay.proId;
```

```
select * from donorpayment;
```

	DonID	DfirstNane	DlastName	paymentAmount	projectName
▶	1	Ali	Jomaa	100.00	Support_sessions_for_official_examination_stu...
	1	Ali	Jomaa	500.00	Restoration_of_poor_people's_home
	2	Hanaa	Mkahal	400.00	cleanliness_campaign
	2	Hanaa	Mkahal	600.00	Restoration_of_poor_people's_home
	3	Kassem	abdalla	400.00	cleanliness_campaign
	3	Kassem	abdalla	200.00	cleanliness_campaign

4#Create a view to show the totalPayments of each donor

```
create view totalPaymentsOfEachDonor as
select DONiD ,dfirstNane , dlastName, sum(paymentAmount) from
donorpayment group by donId;
```

```
select * from totalpaymentOfEachDonor;
```

	DonID	DfirstNane	DlastName	sum(paymentAmount)
▶	1	Ali	Jomaa	600.00
	2	Hanaa	Mkahal	1000.00
	3	Kassem	abdalla	600.00

5#Create a view to show the region of each activity

```
create view regionOfActivity as
select a.activityName, r.regionName from activty a
inner join location l
on a.locId = l.locId
inner join region r
on r.regId = L.reId;

select * from regionOfActivity;
```

activityName	regionName
extra_exercises_solving	Nabateye
cleaning_beach	saida
cleaning_street	beirut
home_renovation	beirut

Here are the views:

X. Procedures:

1#Create a procedure that finds all activities done in beirut specified by input parameter regionName.

```
DELIMITER //
CREATE PROCEDURE GetActivityRegion( IN NameOfREGION
VARCHAR(255))
BEGIN
```

```
SELECT * FROM regionOfActivity WHERE regionName = NameoFregion;
END //
DELIMITER ;

call getactivityregion ('beirut');
```

	activityName	regionName
▶	cleaning_street	beirut
	home_renovation	beirut

2#Create a procedure that returns the average salary of employees according to certain major (case engineer)

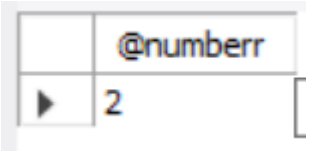
```
DELIMITER $$
CREATE PROCEDURE GetAverageSalary (
IN majorOfEmployee VARCHAR(25),
OUT average numeric(5,2)
)
BEGIN
SELECT avg(salary)
INTO average
FROM employee
WHERE Emajor = majorOfEmployee;
END$$
DELIMITER ;

call getAverageSalary('engineer',@average);
select @average;
```

	@average
▶	500.00

3#Create a procedure that displays the nb of achievemnts in a chosen department

```
delimiter //
```

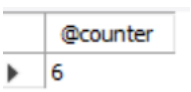
```
create procedure countOfAchievements
(in depName varchar(50), out numberr int )
begin
select count into numberr
from achievements a INNER JOIN  department d on a.achId = d.achId
where d.departmentName = depName;
end //
delimiter ;

call countOfAchievements('education' , @numberr);
select @numberr;
```



	@numberr
▶	2

4#Create a stored procedure SetCounter() that is used to increment the count of achivements of any department.

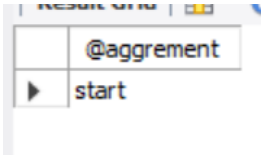
```
DELIMITER $$
CREATE PROCEDURE SetCounter( out counter int ,IN inc INT , in
depName varchar(50))
BEGIN
call countOfAchievements('construction' , @numberr);
select @numberr + inc into counter;
update achievements
set count = counter
where achId =(select achId from department where departmentName
= depName);
END$$
DELIMITER ;

call setCounter(@counter,1,'construction');
select @counter;
```



	@counter
▶	6

5#Create a procedure that checks whether paymentsOfproject is less than 500. If yes, do not start with project, else start.

```
DELIMITER $$

CREATE PROCEDURE startwithproject( nameOfproject varchar(50),

OUT aggrement VARCHAR(20))

BEGIN

DECLARE sum_ int DEFAULT 0;

SELECT s #edited above

INTO sum_

FROM paymentsOfProjectt

where projectName = nameOfproject;

IF sum_ > 500 THEN

SET aggrement = 'start';

END IF;

END$$

DELIMITER ;


call startwithproject('Restoration_of_poor_people''s_home',
@aggrement);

select @aggrement;
```



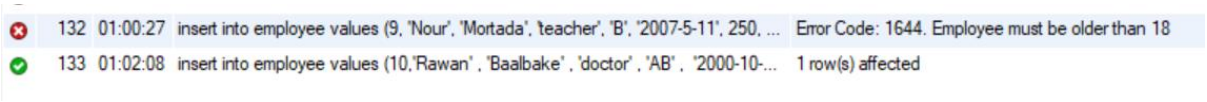
XI. Triggers

1#Create a BEFORE INSERT trigger to check the dateOfbirth value before inserting data into the person table. If less greater than year

2006,the trigger shall signal an error with the following message: "the age must be greater than 18"

```
DELIMITER //
CREATE TRIGGER before_insert_employee
BEFORE INSERT ON employee
FOR EACH ROW
BEGIN
 IF NEW.edateOfbirth > '2006-12-12' THEN
   SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Employee must be older than 18';
 END IF;
END //
DELIMITER ;
```

```
insert into employee values (9, 'Nour', 'Mortada', 'teacher', 'B', '2007-5-11', 250, 10.0, 'nourmortada@gmail.com', 5 ,1);
insert into employee values (10,'Rawan' , 'Baalbake' , 'doctor' , 'AB' , '2000-10-27', 400, 10.0, 'nourBaalbaki@gmail.com', 3, 4);
```



2#Creates an AFTER INSERT trigger that inserts a reminder into the reminders table if bloodtype is AB.

```
CREATE TABLE reminders (
id INT AUTO_INCREMENT,
memberId INT,
message VARCHAR(255) NOT NULL,
PRIMARY KEY (id , memberId)
);

DELIMITER $$
CREATE TRIGGER after_employee_insert
AFTER INSERT
ON employee FOR EACH ROW
BEGIN
IF new.EBloodType= 'AB' THEN
```

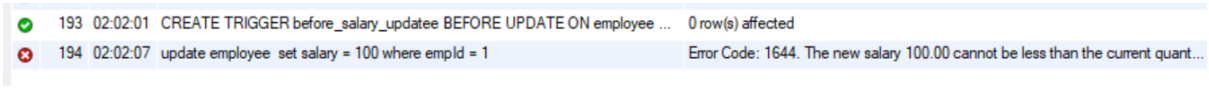
```
INSERT INTO reminders(memberId, message)
VALUES(new.empid,CONCAT('Hi ', new.efirstname, new.elastname,',
please contact us.'));
END IF;
END$$
DELIMITER ;

insert into employee values (8, 'Jawad', 'Abbas', 'farmer', 'AB', '1999-
12-10', 100, 10.0, 'jawadabbas@gmail.com', 5 ,2);
select * from reminders;
```

	id	memberId	message
▶	1	8	Hi JawadAbbas, please contact us.
*	NULL	NULL	NULL

3#Create a BEFORE UPDATE trigger on the employee table. The trigger is automatically fired before an update event occurs for each row in the employee table. If the value in the salary column is updated to a new value that is less than the current value, the trigger raises an error and stops the update.

```
DELIMITER $$
CREATE TRIGGER before_salary_updatee
BEFORE UPDATE
ON employee FOR EACH ROW
BEGIN
DECLARE errorMessage VARCHAR(255);
SET errorMessage = CONCAT('The new salary ', NEW.salary,
' cannot be less than the current quantity ',
OLD.salary);
IF new.salary < old.salary THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = errorMessage;
END IF;
END $$
DELIMITER ;

update employee
set salary = 100
```

```
where empId = 1;
```

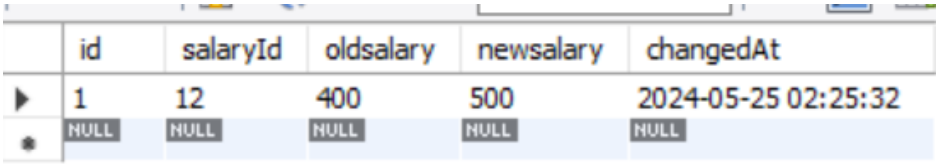


193	02:02:01	CREATE TRIGGER before_salary_updatee BEFORE UPDATE ON employee ...	0 row(s) affected
194	02:02:07	update employee set salary = 100 where empId = 1	Error Code: 1644. The new salary 100.00 cannot be less than the current quant...

4#Create an AFTER UPDATE trigger on the employee table. If you update the value in the quantity column to a new value, the trigger shall insert a new row to log the changes in the SalariesChanges table.

```
CREATE TABLE SalariesChanges (
id INT AUTO_INCREMENT PRIMARY KEY,
salaryId INT,
oldsalary INT,
newsalary INT,
changedAt TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
);


DELIMITER $$
CREATE TRIGGER after_salary_update
AFTER UPDATE
ON employee FOR EACH ROW
BEGIN
IF OLD.salary <> new.salary THEN
INSERT INTO SalariesChanges(salaryId,oldsalary, newsalary)
VALUES(old.empid, old.salary, new.salary);
END IF;
END$$
DELIMITER ;

insert into employee values
(11, 'jaafar', 'Kenaan', 'docotor', 'AB', '2000-10-26', 400, 10.0,
'jaafarkenaan@gmail.com', 3, 4),
(12, 'raneem', 'Kenaan', 'nurse', 'AB', '2000-10-26', 400, 10.0,
'raneemkenaan@gmail.com', 3, 4),
(13, 'sokayna', 'Zaraket', 'teacher', 'B', '2001-11-12', 250, 10.0,
'sokayanazaraket@gmail.com', 5 ,1);

update employee
set salary = 500
```

```
where empId = 12;

select * from salarieschanges;
select* from employee;
```



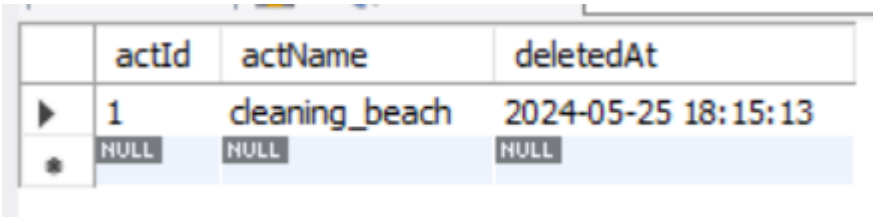
	id	salaryId	oldsalary	newsalary	changedAt
▶	1	12	400	500	2024-05-25 02:25:32
*	NULL	NULL	NULL	NULL	NULL

5#Create a trigger that inserts a new row into the activitiesArchives table before a row from the currentActivities table is deleted.

```
create table currentActivities(
actId int not null primary key,
actName varchar(50)
);

insert into currentActivities(actId,ActName) values
(1,'cleaning_beach'),(2,'cleaning_street');
select * from currentActivities;

create table activitiesArchive(
actId int not null primary key,
actName varchar(50),
deletedAt TIMESTAMP DEFAULT NOW()
);

DELIMITER $$
CREATE TRIGGER before_activity_delette
BEFORE DELETE
ON currentactivities FOR EACH ROW
BEGIN
INSERT INTO activitiesArchive(actId,ActName)
VALUES(old.actId,old.actName);
END$$
DELIMITER ;
```

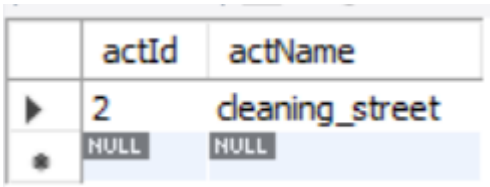
```
delete from currentActivities where actName ='cleaning_beach';
```

```
select * from activitiesArchive;
```



actId	actName	deletedAt
1	cleaning_beach	2024-05-25 18:15:13
NULL	NULL	NULL

```
select * from currentactivities;
```

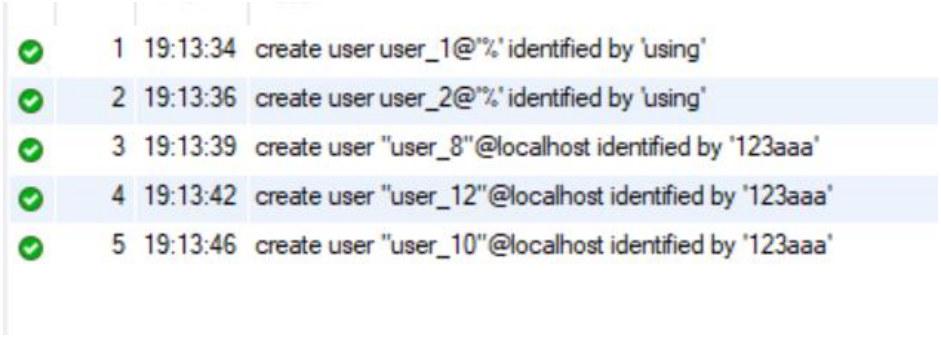


actId	actName
2	cleaning_street
NULL	NULL

XII. Creating users and granting permissions

#Creating 5 users with passwords ‘using’ for remote hosts and ‘123aaa’ for local hosts

```
create user user_1@'%' identified by 'using';
create user user_2@'%' identified by 'using';
create user "user_8"@localhost identified by '123aaa';
create user "user_12"@localhost identified by '123aaa';
create user "user_10"@localhost identified by '123aaa';
```



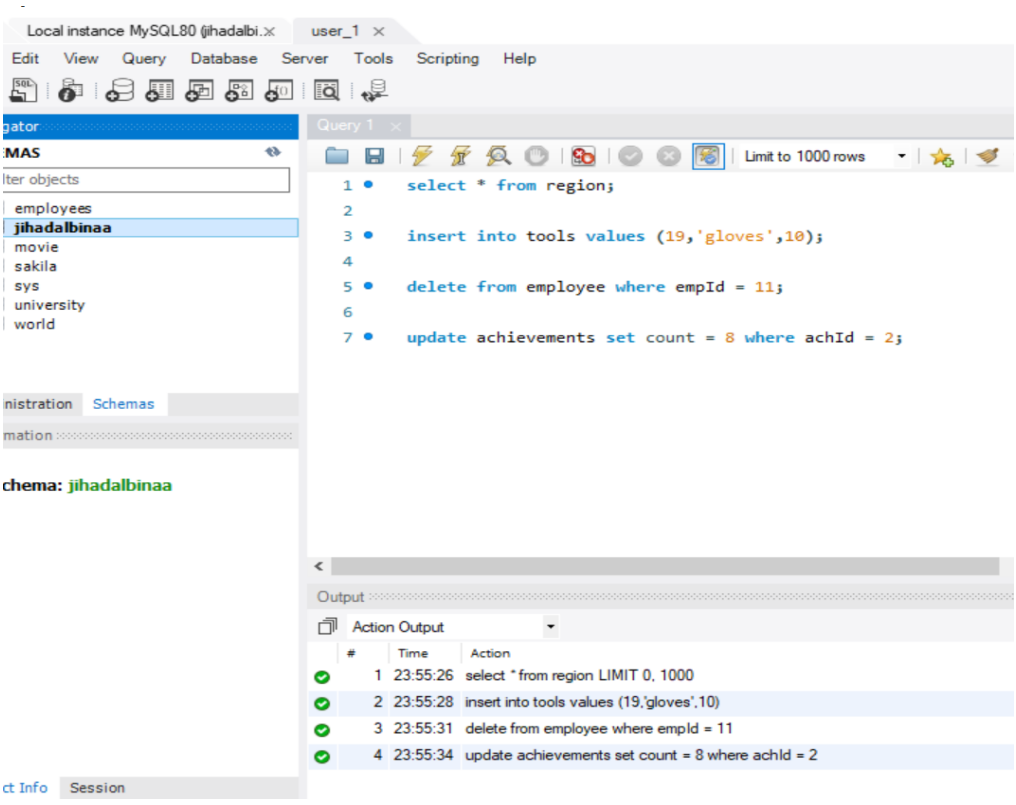
✓	1	19:13:34	create user user_1@'%' identified by 'using'
✓	2	19:13:36	create user user_2@'%' identified by 'using'
✓	3	19:13:39	create user "user_8"@localhost identified by '123aaa'
✓	4	19:13:42	create user "user_12"@localhost identified by '123aaa'
✓	5	19:13:46	create user "user_10"@localhost identified by '123aaa'

```
#permissions:

#for user_1 : can do anything in all databases

grant all on *.* to user_1@'%';

#for user_2 : can only select,  update, insert in activity and do anything in tools in jihadalbinaa database

grant select, update, insert on jihadalbinaa.activty to user_2@'%';

grant select, update, insert, delete on jihadalbinaa.tools to user_2@'%';

#for user_8 : can don anything in employee, department, achievements in jihadalbinaa database

grant select, update, insert, delete on jihadalbinaa.employee to "user_8"@localhost;

grant select, update, insert, delete on jihadalbinaa.department to "user_8"@localhost;

grant select, update, insert, delete on jihadalbinaa.achievements to "user_8"@localhost;

#for user_12 : can do anything in region table in jihadalbinaa database

grant select, update, insert, delete on jihadalbinaa.region to "user_12"@localhost;

#for user_10 : can only select from the jihadalbinaa database tables

grant select on jihadalbinaa.*  to "user_10"@localhost;
```

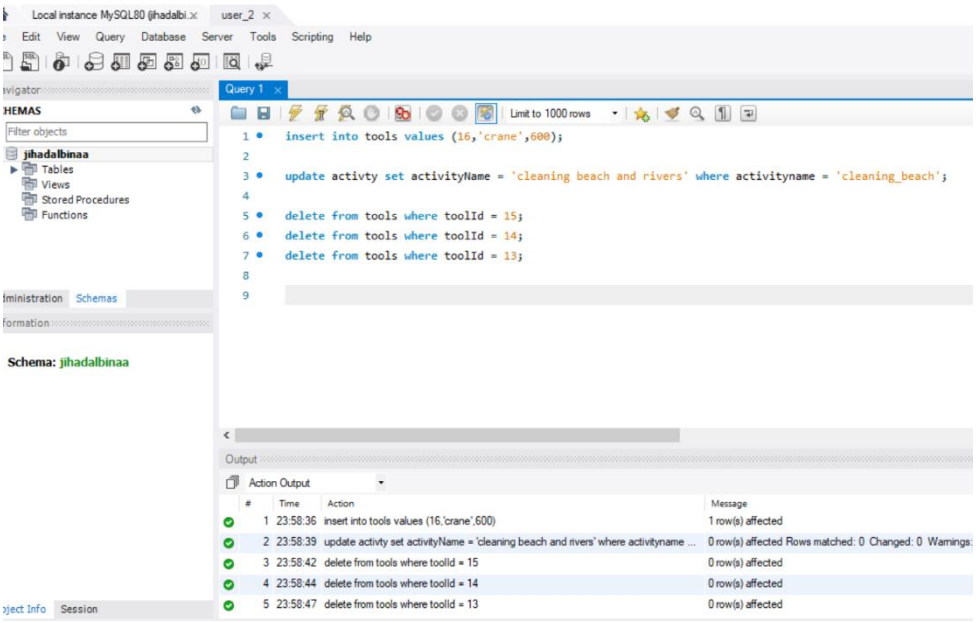
6	19:19:41	grant all on *.* to user_1@'%'
7	19:19:46	grant select, update, insert on jihadalbinaa.activty to user_2@'%'
8	19:19:49	grant select, update, insert, delete on jihadalbinaa.tools to user_2@'%'
9	19:19:53	grant select, update, insert, delete on jihadalbinaa.employee to "user_8"@localhost
10	19:19:56	grant select, update, insert, delete on jihadalbinaa.department to "user_8"@localhost
11	19:19:59	grant select, update, insert, delete on jihadalbinaa.achievements to "user_8"@local...
12	19:22:28	grant select, update, insert, delete on jihadalbinaa.region to "user_12"@localhost
19	20:53:22	grant select on jihadalbinaa.* to "user_10"@localhost

XIII. Testing

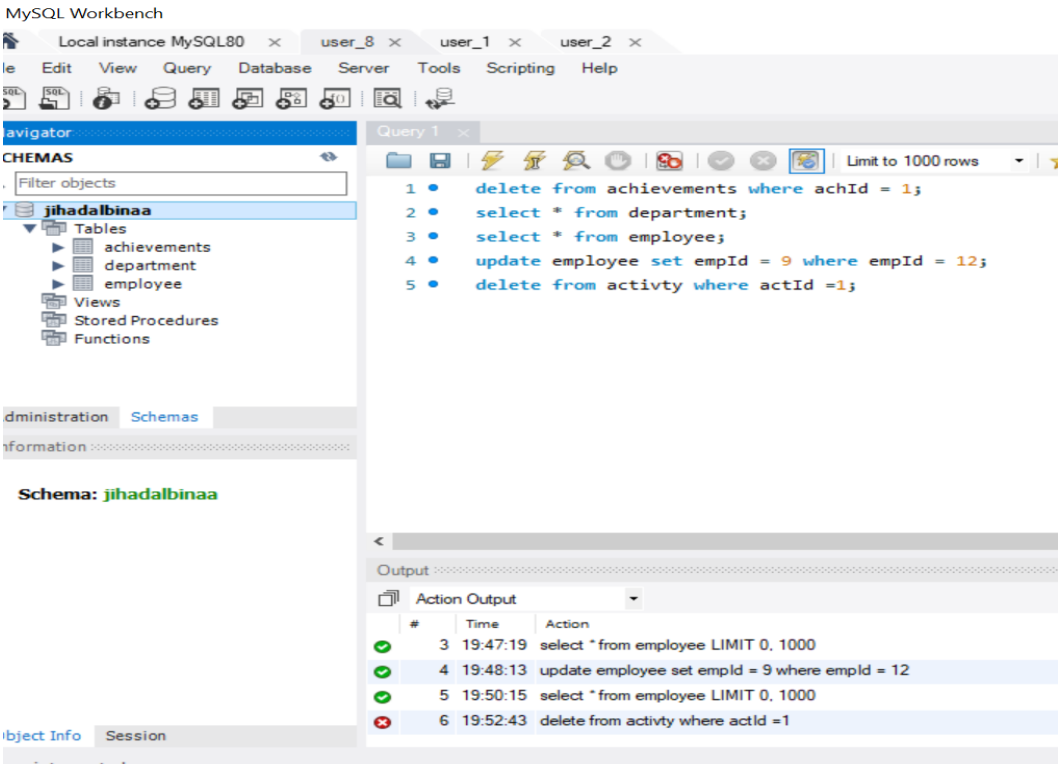
#for user_1



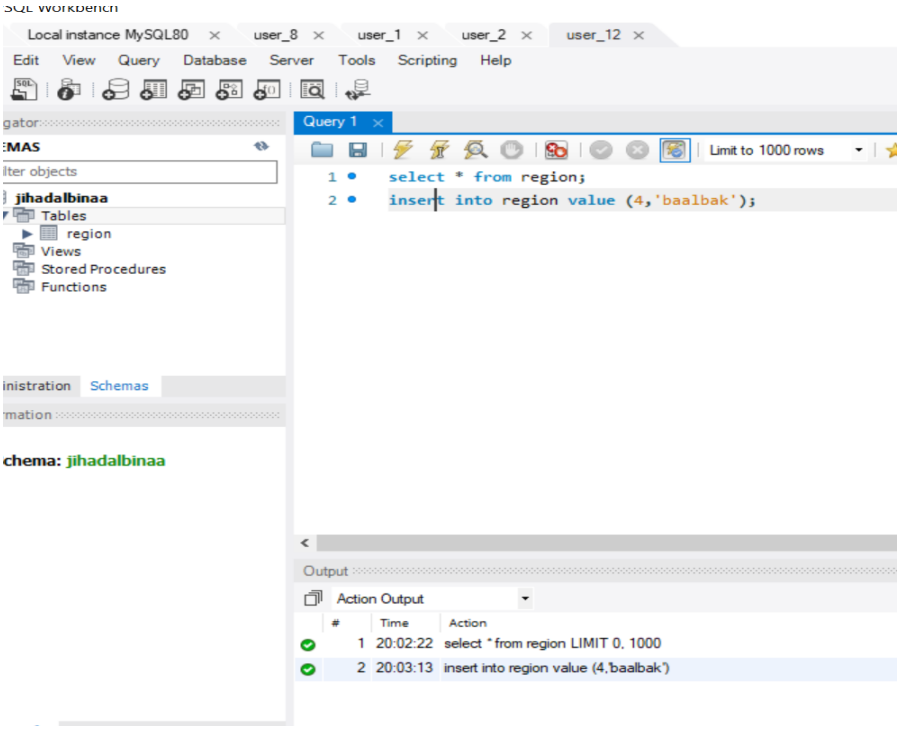
#for user_2



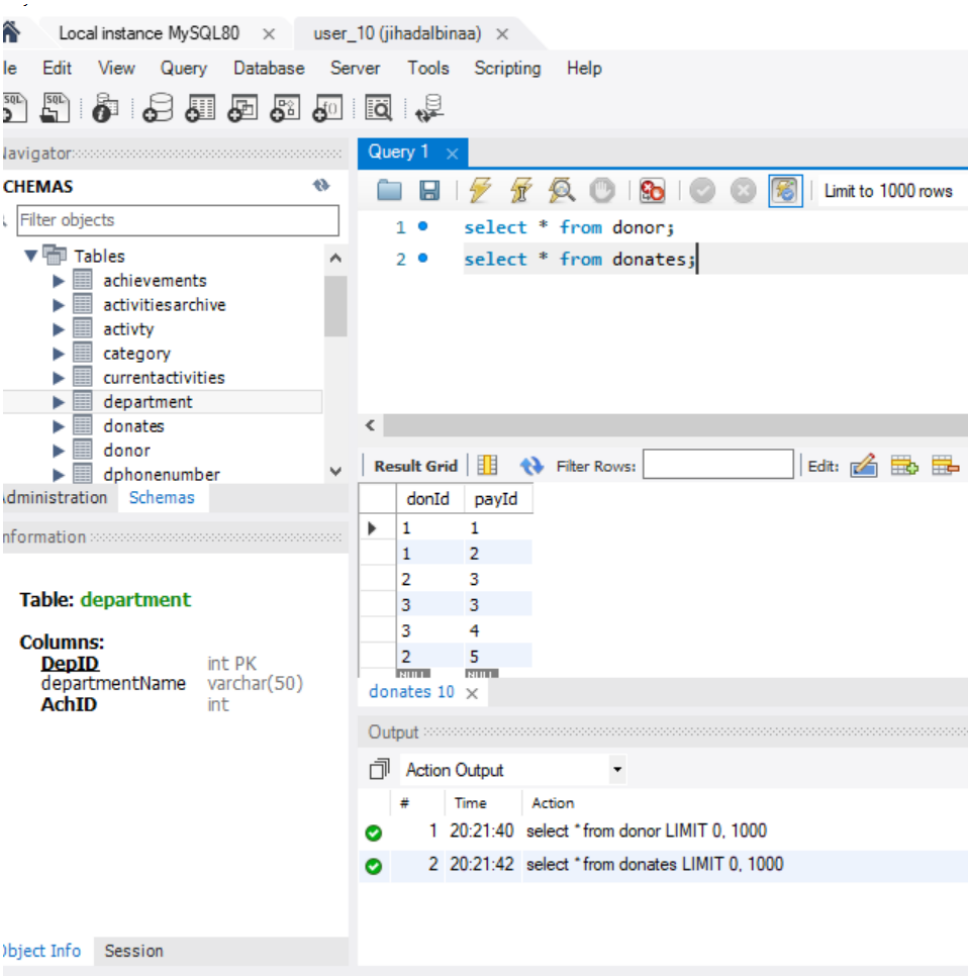
#for user 8



#user 12


Local instance MySQL80
user_8
user_1
user_2
user_12
Edit View Query Database Server Tools Scripting Help
Query 1
Limit to 1000 rows
1 select * from region;
2 insert into region value (4,'baalbak');
jihadalbinaa
Tables
region
Views
Stored Procedures
Functions
Schemas
chema: jihadalbinaa
Output
Action Output
Time Action
1 20:02:22 select * from region LIMIT 0, 1000
2 20:03:13 insert into region value (4,'baalbak')

#user 10


Local instance MySQL80
user_10 (jihadalbinaa)
Edit View Query Database Server Tools Scripting Help
Query 1
Limit to 1000 rows
1 select * from donor;
2 select * from donates;
CHEMAS
Filter objects
Tables
achievements
activitiesarchive
activty
category
currentactivities
department
donates
donor
dphonenumber
Schemas
Table: department
Columns:
DepID int PK
departmentName varchar(50)
AchID int
Result Grid
Filter Rows:
Edit:
donId payId
1 1
1 2
2 3
3 3
3 4
2 5
donates 10
Output
Action Output
Time Action
1 20:21:40 select * from donor LIMIT 0, 1000
2 20:21:42 select * from donates LIMIT 0, 1000
Session