

Database Management & Database Design

Final Project Proposal

Name: Raj R Phadke

NUID: 001814155

1. Topic:

Database for an Automobile Manufacturing Unit (Factory)

2. Problem Statement:

An automobile manufacturing unit (Factory) manufactures an automobile vehicle(s) by collecting the required materials and assembling a fully functional vehicle.

From collecting raw materials to assembling a fully functional vehicle, there are many sub processes which facilitate the manufacturing process as whole.

Motor companies have fixed set of standards based on which vehicles are manufactured. This project will simulate the manufacturing process by collecting and uniting all the data.

By considering the functioning of one manufacturing unit (factory) of a motor company, the database will store information of all the components (mentioned in pt. 3) of the factory that altogether aid in the manufacturing process.

The database will include

- Triggers
- Stored Procedures
- Access Control
- Views etc.

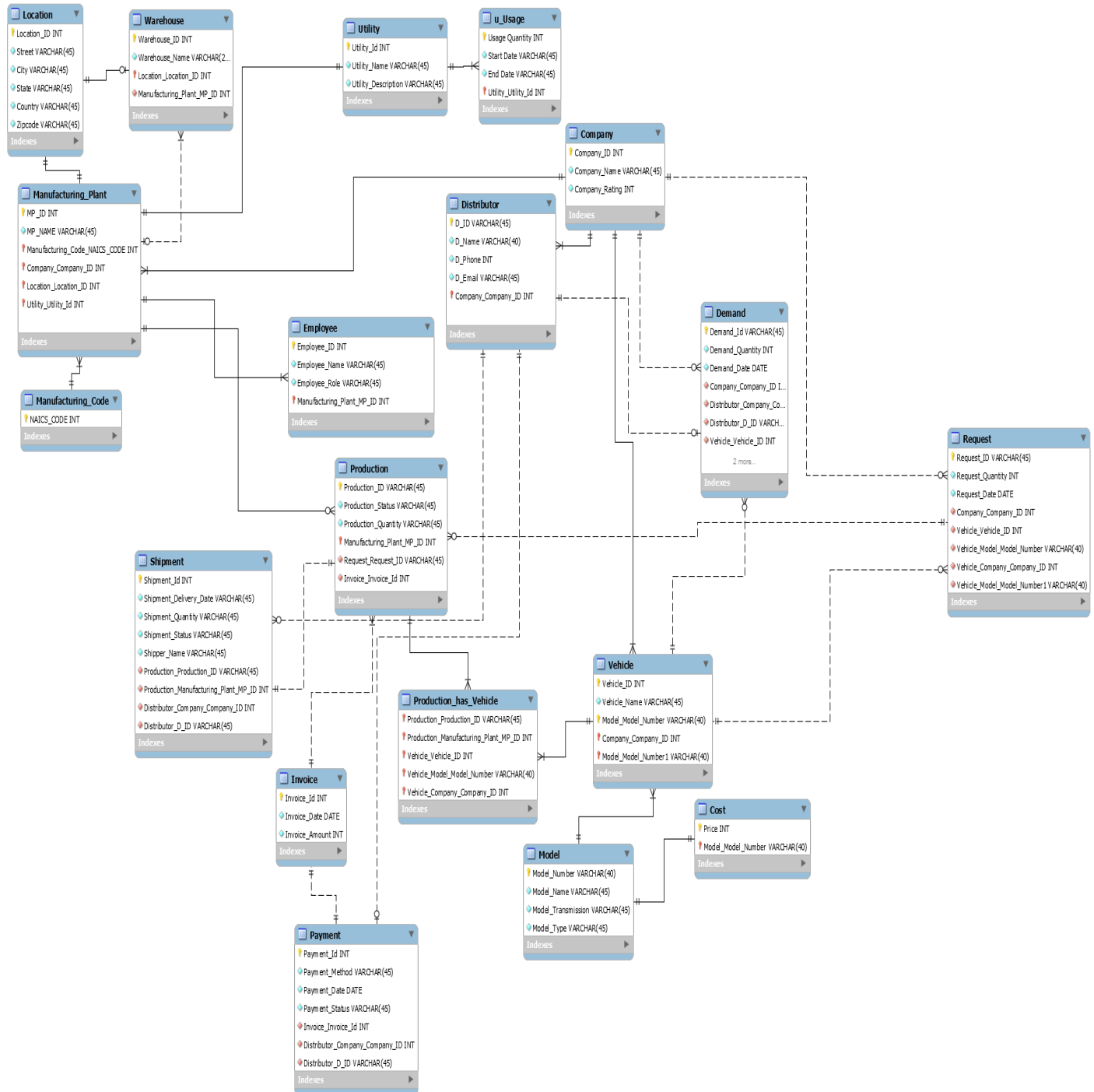
3. Following is the list of tables that the database includes:

1. Manufacturing Plant(Factory)- factory id, name etc.
2. Company – Ford in our example.
3. Location – address of the factory.
4. Manufacturing Code - Establishments primarily engaged in manufacturing car bodies and assembling vehicles on a chassis and manufacturing kit cars for highway use are classified in U.S. Industry by “**NAICS**” code.
5. Employees – workers/assemblers with a role.
6. Production – production related statistics such as per day production.
7. Utility – type of utility used
8. Usage- amount of electricity, gas or other utility used from a date to a date.
9. Demand- need of the distributor/dealership.
10. Model/Make – type of vehicle.
11. Distributor – an organization or a dealership that focuses on selling the manufactured cars.
12. Shipment – Shipment process table
13. Invoices – payment bill generated.
14. Cost – cost per vehicle model/make.
15. Warehouses – a depository where extra components are stored.
16. Payment – payment related information.
17. Request – amount of vehicles requested by the company to be produced
18. Vehicle -- in our example, vehicle is the product manufactured.
19. Production_has_Vehicle – linkage table between production table and vehicle table in a many to many manner.

4. **Relationships**

1. Manufacturing plant to use a utility is mandatory and every utility is used by a manufacturing plant.
2. Utility has one or more usage records and every there is a mandatory usage of utility.
3. Manufacturing plant may have one or many warehouses. Warehouse may be assigned to zero or one manufacturing plant.
4. Every manufacturing plant has one location, but warehouse won't have any location if the manufacturing plant has no location.
5. Every manufacturing plant works on one manufacturing code whereas the manufacturing code may be assigned to one or many manufacturing plants.
6. Manufacturing plant has one or many employees and every employee is assigned to one manufacturing plant only.
7. Every manufacturing plant exists for a company (Ford in our case). Whereas a company may have one or more manufacturing plants.
8. Manufacturing plant may have zero or many productions depending on the need of the distributor/dealership. But every production is manufacturing plant specific.
9. A distributor is based on a company whereas a company may have one or many distributors/ dealerships.
10. A distributor provides its needs to the company. Example. Request of vehicles to sell through the dealership.
In which sense, distributor may have zero or one demand and the company would have zero or many demands.
11. Demands are based on a vehicle and a vehicle may have zero or many demands.
12. In case of need, a company may send zero or many requests for production but every request has to mandatory come from a company.
13. If there are no requests, there won't be any production. If requests exist, there may be many productions.
14. Every vehicle is mandatory based on one model and make. Whereas there can be multiple vehicles which are based on the same model.
15. Every model has a cost associated with it and the cost is selective only to that model.
16. Productions of multiple vehicles is possible and it is possible for a vehicle to exist in multiple productions. Production generates an invoice and an invoice can be generated for one or many productions.
17. Production makes the need of shipment and every shipment is based on one production.
18. Invoice is associated to a payment and every payment has one invoice based on which the payment is made.
19. Payment is done by a distributor/ dealership and a distributor can make one payment or no payment (if the distributor has no needs and places no request for production).
20. Shipment must be delivered to the dealership/ distributor who ordered it and the distributor may have one or no shipment (if it doesn't request a production).

5. Final ER diagram



Implementation















1. The project includes multiple user roles, not in the form of tables but individual connections created at the MYSQL -Workbench.

These users are:

1. Production Admin which manages data related to the production side of the database.
2. Company Admin which manages data related to the company, its warehouse, its distributors/dealership and all the demands, needs and request.
3. Distributor Admin which manages the distributor side, generating demands and sending requests.
4. Employee Admin which adds new employees to the database as soon as they are recruited.
5. Shipment Admin which manages the shipment side of the database.
6. Finance Admin which manages the invoices and the payment process.

And obviously the System ADMIN which I predefined as the root localhost.

MySQL Connections

Local instance MySQL57  root  localhost:3306	ProductionAdmin  ProductionMa...  127.0.0.1:3306	CompanyAdmin  ComManager...  127.0.0.1:3306	DistributorAdmin  DistributorDeal...  127.0.0.1:3306
Employee  Emp@localhost  127.0.0.1:3306	Shipment Admin  ShipmentMan...  127.0.0.1:3306	Finance Admin  FinanceMan@l...  127.0.0.1:3306	

2. Creating users and their passwords

SQL CODE:

```
-create user 'ProductionManager@localhost' @'localhost' identified by 'productionManager';
-create user 'ComManager@localhost' @'localhost' identified by 'companyManager';
-create user 'DistributorDealership@localhost' @'localhost' identified by 'dd';
-create user 'Emp@localhost' @'localhost' identified by 'emp';
-create user 'ShipmentManager@localhost' @'localhost' identified by 'sm';
-create user 'FinanceMan@localhost' @'localhost' identified by 'fm';
```

```
211 • create user 'ProductionManager@localhost' @'localhost' identified by 'productionManager';
212 • create user 'ComManager@localhost' @'localhost' identified by 'companyManager';
213 • create user 'DistributorDealership@localhost' @'localhost' identified by 'dd';
214 • create user 'Emp@localhost' @'localhost' identified by 'emp';
215 • create user 'ShipmentManager@localhost' @'localhost' identified by 'sm';
216 • create user 'FinanceMan@localhost' @'localhost' identified by 'fm';
217
```

OUTPUT:

✓	133	18:34:39	create user 'ProductionManager@localhost' @'localhost' identified by 'productionManager'	0 row(s) affected	0.000 sec
✗	134	18:35:08	create user 'CompanyManager@localhost' @'localhost' identified by 'companyManager'	Error Code: 1396. Operation CREATE USER failed for 'CompanyManager@localhost'@'local...	0.000 sec
✓	135	18:35:19	create user 'ComManager@localhost' @'localhost' identified by 'companyManager'	0 row(s) affected	0.000 sec
✓	136	18:35:50	create user 'DistributorDealership@localhost' @'localhost' identified by 'dd'	0 row(s) affected	0.000 sec
✗	137	18:36:12	create user 'Employee@localhost' @'localhost' identified by 'emp'	Error Code: 1396. Operation CREATE USER failed for 'Employee@localhost'@'localhost'	0.000 sec
✓	138	18:36:18	create user 'Emp@localhost' @'localhost' identified by 'emp'	0 row(s) affected	0.000 sec
✓	139	18:36:41	create user 'ShipmentManager@localhost' @'localhost' identified by 'sm'	0 row(s) affected	0.000 sec
✗	140	18:37:00	create user 'FinanceManager@localhost' @'localhost' identified by 'fm'	Error Code: 1396. Operation CREATE USER failed for 'FinanceManager@localhost'@'localh...	0.000 sec
✓	141	18:37:05	create user 'FinanceMan@localhost' @'localhost' identified by 'fm'	0 row(s) affected	0.000 sec

2. Entering data into the demand table using nested query and select subquery

-insert into demand

select 6, (select request.Request_Quantity from request where Request_ID=6),
'2017-02-12',1,1,4,1,'2GA',1;

```
insert into demand()
values(1,50,'2017-10-06',1,1,4,1,'2GA',1);
insert into demand()
values(2,75,'2017-07-06',1,1,3,4,'ABEILLE',1);
insert into demand()
values(3,15,'2017-06-06',1,1,2,7,'MUSTANG',1);
insert into demand()
values(4,04,'2017-12-06',1,1,5,2,'300',1);
insert into demand()
values(5,20,'2017-02-06',1,1,1,5,'ANGLIA',1);

insert into demand
select 6,(select request.Request_Quantity from request where Request_ID=6),
'2017-02-12',1,1,4,1,'2GA',1;
```

Output:

Demand_Id	Demand_Quantity	Demand_Date	Company_Company_ID	Distributor_Company_Company_ID	Distributor_D_ID	Vehicle_Vehicle_ID
1	50	2017-10-06	1	1	4	1
2	75	2017-07-06	1	1	3	4
3	15	2017-06-06	1	1	2	7
4	4	2017-12-06	1	1	5	2
5	20	2017-02-06	1	1	1	5
6	10	2017-02-12	1	1	4	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Vehicle_Model_Model_Number	Vehicle_Company_Company_ID
2GA	1
ABEILLE	1
MUSTANG	1
300	1
ANGLIA	1
2GA	1
NULL	NULL

3. Inserting data into the invoice table

Using subqueries

-insert into invoice

```
select 1,(select adddate((Select request.Request_Date from request where
Request_ID=1),03))
```

```
,750000;
```

-insert into invoice

```
select 2,(select adddate((Select request.Request_Date from request where
Request_ID=2),03))
```

```
,2250000;
```

-insert into invoice

```
select 3,(select adddate((Select request.Request_Date from request where
Request_ID=3),03))
```

```
,675000;
```

-insert into invoice

```
select 4,(select adddate((Select request.Request_Date from request where
Request_ID=4),03))
```

```
,80000;
```

-insert into invoice

```
select 5,(select adddate((Select request.Request_Date from request where
Request_ID=5),03))
```

```
,700000;
```

-insert into invoice

```
select 6,(select adddate((Select request.Request_Date from request where
Request_ID=6),03))
```

```
,90000;
```

Output:

Invoice_Id	Invoice_Date	Invoice_Amount
1	2017-10-12	750000
2	2017-07-12	2250000
3	2017-06-12	675000
4	2017-12-12	80000
5	2017-02-12	700000
6	2017-02-15	90000
NULL	NULL	NULL

4. Inserting data into the production table using subqueries

```
select * from production;

insert into production
select 1,'starting',(select request.Request_Quantity from request where Request_ID=1),1,1,1;
insert into production
select 2,'starting',(select request.Request_Quantity from request where Request_ID=2),2,2,2;
insert into production
select 3,'starting',(select request.Request_Quantity from request where Request_ID=3),3,3,3;
insert into production
select 4,'starting',(select request.Request_Quantity from request where Request_ID=4),4,4,4;
insert into production
select 5,'starting',(select request.Request_Quantity from request where Request_ID=5),5,5,5;
insert into production
select 6,'starting',(select request.Request_Quantity from request where Request_ID=6),1,6,6;
```

Output:

Result Grid						
Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:
Production_ID	Production_Status	Production_Quantity	Manufacturing_Plant_MP_ID	Request_Request_ID	Invoice_Invoice_Id	
1	completed	50	1	1	1	
2	completed	75	2	2	2	
3	completed	15	3	3	3	
4	completed	4	4	4	4	
5	completed	20	5	5	5	
6	completed	10	1	6	6	
NULL	NULL	NULL	NULL	NULL	NULL	

5. Inserting data into the linkage table (production_has_vehicle) (many-to-many relationship)

```
--insert into production_has_vehicle
```

```
select
```

```
(select Production_ID from production where Production_ID=1),  
(select production.Manufacturing_Plant_MP_ID from production where  
production.Production_ID=1),  
(select vehicle.Vehicle_ID from vehicle where Vehicle_ID in (select  
request.Vehicle_Vehicle_ID from request where Request_ID=1)),  
'2GA',
```

```
1;
```

```
--insert into production_has_vehicle
```

```
select
```

```
(select Production_ID from production where Production_ID=2),  
(select production.Manufacturing_Plant_MP_ID from production where  
production.Production_ID=2),  
(select vehicle.Vehicle_ID from vehicle where Vehicle_ID in (select  
request.Vehicle_Vehicle_ID from request where Request_ID=2)),  
'ABEILLE',
```

```
1;
```

```
--insert into production_has_vehicle
```

```
select
```

```
(select Production_ID from production where Production_ID=3),  
(select production.Manufacturing_Plant_MP_ID from production where  
production.Production_ID=3),  
(select vehicle.Vehicle_ID from vehicle where Vehicle_ID in (select  
request.Vehicle_Vehicle_ID from request where Request_ID=3)),  
'MUSTANG',
```

```
1;
```

```
--insert into production_has_vehicle
```

```
select
```

```
(select Production_ID from production where Production_ID=4),  
(select production.Manufacturing_Plant_MP_ID from production where  
production.Production_ID=4),  
(select vehicle.Vehicle_ID from vehicle where Vehicle_ID in (select  
request.Vehicle_Vehicle_ID from request where Request_ID=4)),  
'300',
```

```
1;
```

```
--insert into production_has_vehicle
```

```
select
```

```
(select Production_ID from production where Production_ID=5),
```

```

(select production.Manufacturing_Plant_MP_ID from production where
production.Production_ID=5),
(select vehicle.Vehicle_ID from vehicle where Vehicle_ID in (select
request.Vehicle_Vehicle_ID from request where Request_ID=5)),
'ANGLIA',
1;

-- insert into production_has_vehicle
select
(select Production_ID from production where Production_ID=6),
(select production.Manufacturing_Plant_MP_ID from production where
production.Production_ID=6),
(select vehicle.Vehicle_ID from vehicle where Vehicle_ID in (select
request.Vehicle_Vehicle_ID from request where Request_ID=6)),
'2GA',
1;

```

Output:

Production_Production_ID	Production_Manufacturing_Plant_MP_ID	Vehicle_Vehicle_ID	Vehicle_Model_Model_Number	Vehicle_Company_Company_ID
1	1	1	2GA	1
6	1	1	2GA	1
4	4	2	300	1
2	2	4	ABEILLE	1
5	5	5	ANGLIA	1
3	3	7	MUSTANG	1
NULL	NULL	NULL	NULL	NULL

6. Adding data to the shipment table using subqueries

```

insert into shipment
select 1,(select adddate((select Request.Request_Date from Request where Request.request_id=1),04))
,(select Request.Request_Quantity from request where request.Request_ID=1)
,'dispatched','FedX',1,1,1,
(select demand.Distributor_D_ID from demand where Demand_Id=1);
insert into shipment
select 2,(select adddate((select Request.Request_Date from Request where Request.request_id=2),04))
,(select Request.Request_Quantity from request where request.Request_ID=2)
,'dispatched','FedX',2,2,1,(select demand.Distributor_D_ID from demand where Demand_Id=2);
insert into shipment
select 3,(select adddate((select Request.Request_Date from Request where Request.request_id=3),04))
,(select Request.Request_Quantity from request where request.Request_ID=3)
,'dispatched','FedX',3,3,1,(select demand.Distributor_D_ID from demand where Demand_Id=3);
insert into shipment
select 4,(select adddate((select Request.Request_Date from Request where Request.request_id=4),04))
,(select Request.Request_Quantity from request where request.Request_ID=4)
,'dispatched','FedX',4,4,1,(select demand.Distributor_D_ID from demand where Demand_Id=4);
insert into shipment
select 5,(select adddate((select Request.Request_Date from Request where Request.request_id=5),04))
,(select Request.Request_Quantity from request where request.Request_ID=5)
,'dispatched','FedX',5,5,1,(select demand.Distributor_D_ID from demand where Demand_Id=5);
insert into shipment
select 6,(select adddate((select Request.Request_Date from Request where Request.request_id=6),04))
,(select Request.Request_Quantity from request where request.Request_ID=6)
,'dispatched','FedX',5,5,1,(select demand.Distributor_D_ID from demand where Demand_Id=6);

```

Output:

Shipment_ID	Shipment_Delivery_Date	Shipment_Quantity	Shipment_Status	Shipper_Name	Production	Production_Manufacturi	Distributor_Company_Company_ID	Distributor_D_ID
1	2017-10-13	50	delivered	FedX	1	1	1	4
2	2017-07-13	75	delivered	FedX	2	2	1	3
3	2017-06-13	15	delivered	FedX	3	3	1	2
4	2017-12-13	4	delivered	FedX	4	4	1	5
5	2017-02-13	20	delivered	FedX	5	5	1	1
6	2017-02-16	10	dispatched	FedX	5	5	1	4
NULL	NULL	NULL	NULL	NULL	NULL	5	NULL	NULL

7. Inserting data into the payment table using subqueries

```
--insert into payment
select 1,'Cheque',
(select adddate((Select request.Request_Date from request where Request_ID=1),10)),
'Paid',
1,
(select shipment.Distributor_Company_Company_ID from shipment where
shipment.Shipment_Id=1),
(select shipment.Distributor_D_ID from shipment where shipment.Shipment_Id=1);
```

```
--insert into payment
select 2,'Cheque',
(select adddate((Select request.Request_Date from request where Request_ID=2),10)),
'Paid',
2,
(select shipment.Distributor_Company_Company_ID from shipment where
shipment.Shipment_Id=2),
(select shipment.Distributor_D_ID from shipment where shipment.Shipment_Id=2);
```

```
--insert into payment
select 3,'Cheque',
(select adddate((Select request.Request_Date from request where Request_ID=3),10)),
'Paid',
3,
(select shipment.Distributor_Company_Company_ID from shipment where
shipment.Shipment_Id=3),
(select shipment.Distributor_D_ID from shipment where shipment.Shipment_Id=3);
```

```
--insert into payment
select 4,'Cheque',
(select adddate((Select request.Request_Date from request where Request_ID=4),10)),
'Paid',
4,
(select shipment.Distributor_Company_Company_ID from shipment where
shipment.Shipment_Id=4),
(select shipment.Distributor_D_ID from shipment where shipment.Shipment_Id=4);
```

```
--insert into payment
select 5,'Cheque',
(select adddate((Select request.Request_Date from request where Request_ID=5),10)),
'Paid',
5,
(select shipment.Distributor_Company_Company_ID from shipment where
shipment.Shipment_Id=5),
```


4. Created Views

1. Company Manager wants to view all employees working in a manufacturing plant and company

SQL query:

```
/* Company Manager wants to view all employees working in a particular manufacturing plant and company */
create view employee_view as
select employee.Employee_ID,employee.Employee_Name,employee.Employee_Role
,manufacturing_plant.MP_ID,manufacturing_plant.MP_NAME,company.company_id,company.company_name
from employee as employee inner join
manufacturing_plant as manufacturing_plant inner join
company as company
on employee.Manufacturing_Plant_MP_ID=manufacturing_plant.MP_ID and company.company_id=manufacturing_plant.Company_Company_ID;
```

```
--
create view employee_view as
select employee.Employee_ID, employee.Employee_Name, employee.Employee_Role
,manufacturing_plant.MP_ID,manufacturing_plant.MP_NAME,company.company_id,
company.company_name
from employee as employee inner join
manufacturing_plant as manufacturing_plant inner join
company as company
on employee.Manufacturing_Plant_MP_ID=manufacturing_plant.MP_ID
and
company.company_id=manufacturing_plant.Company_Company_ID;
```

Output:

Executing the view as:

```
select * from employee_view;
```

	Employee_ID	Employee_Name	Employee_Role	MP_ID	MP_NAME	company_id	company_name
	1	TUSHAR	Company Manager	1	Ford Motors Division 1	1	Ford Motors
	2	DHARIT	Company Manager	2	Ford Motors Division alpha	1	Ford Motors
	3	SAMEER	Company Manager	3	Ford Motors Division beta	1	Ford Motors
	4	SAMI	Company Manager	4	Ford Motors Division gama	1	Ford Motors
	5	VIKAS	Company Manager	5	Ford Motors Division pico	1	Ford Motors
	6	NAMAN	Employee	1	Ford Motors Division 1	1	Ford Motors Division pico
	7	JAY	Employee	2	Ford Motors Division alpha	1	Ford Motors
	8	RAJ	Employee	3	Ford Motors Division beta	1	Ford Motors
	9	ASHISH	Employee	4	Ford Motors Division gama	1	Ford Motors
	10	ADITYA	Employee	5	Ford Motors Division pico	1	Ford Motors
	11	SUSHANT	Employee	2	Ford Motors Division alpha	1	Ford Motors

2. View to display company managers

```
/* Company Admin wants to see only Company Managers */
create view Company_Manager_view as
select employee.Employee_ID,employee.Employee_Name,employee.Employee_Role
,manufacturing_plant.MP_ID,manufacturing_plant.MP_NAME,company.company_id,company.company_name
from employee as employee inner join
manufacturing_plant as manufacturing_plant inner join
company as company
on employee.Manufacturing_Plant_MP_ID=manufacturing_plant.MP_ID and company.company_id=manufacturing_plant.Company_Company_ID
where employee.Employee_Role='Company Manager';
```

```
--
create view Company_Manager_view as
select employee.Employee_ID,employee.Employee_Name,employee.Employee_Role
,manufacturing_plant.MP_ID,manufacturing_plant.MP_NAME,company.company_id,
company.company_name
from employee as employee inner join
manufacturing_plant as manufacturing_plant inner join
company as company
on employee.Manufacturing_Plant_MP_ID=manufacturing_plant.MP_ID and
company.company_id=manufacturing_plant.Company_Company_ID
where employee.Employee_Role='Company Manager';
```

Output:

Executing the view as

```
select * from Company_Manager_view;
```

	Employee_ID	Employee_Name	Employee_Role	MP_ID	MP_NAME	company_id	company_name
	1	TUSHAR	Company Manager	1	Ford Motors Division 1	1	Ford Motors
	2	DHARIT	Company Manager	2	Ford Motors Division alpha	1	Ford Motors
	3	SAMEER	Company Manager	3	Ford Motors Division beta	1	Ford Motors
	4	SAMI	Company Manager	4	Ford Motors Division gama	1	Ford Motors
	5	VIKAS	Company Manager	5	Ford Motors Division dico	1	Ford Motors

3. View to display all the vehicles available

--

create view all_vehicle

as

select

vehicle.Vehicle_ID,vehicle.Vehicle_Name,model.Model_Name,model.Model_Number,
model.Model_Transmission,model.Model_Type,cost.Price
from vehicle inner join model inner join cost
on vehicle.Model_Model_Number1=model.Model_Number and
model.Model_Number=cost.Model_Model_Number ;

output:

```

290
291 • create view all_vehicle
292 as
293 select vehicle.Vehicle_ID,vehicle.Vehicle_Name,model.Model_Name,model.Model_Number,model.Model_Transmission,model.Model_Type,cost.Price
294 from vehicle inner join model inner join cost
295 on vehicle.Model_Model_Number1=model.Model_Number and model.Model_Number=cost.Model_Model_Number ;
296
297 • select * from all_vehicle;
298

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Vehicle_ID	Vehicle_Name	Model_Name	Model_Number	Model_Transmission	Model_Type	Price
1	Ford 20a	2GA	1	AUTO	HATCHBACK	15000
2	Ford 300	300	2	AUTO	HATCHBACK	20000
3	Ford 7W	7W	3	AUTO	HATCHBACK	25000
4	Ford ABEILLE	ABEILLE	4	AUTO	HATCHBACK	30000
5	Ford ANGLIA	ANGLIA	5	AUTO	SPORT	35000
6	Ford CAPRI	CAPRI	6	AUTO	SPORT	40000
7	Ford MUSTANG	MUSTANG	7	AUTO	SEDAN	45000
8	Ford EXP	EXP	8	AUTO	HATCHBACK	50000
9	Ford ELITE	ELITE	9	AUTO	SEDAN	55000
10	Ford PAMPA	PAMPA	10	AUTO	SPORT	60000

5. Other part

1. Maximum requested vehicles sorted by quantity ascending(not grouped by)
SQL QUERY and output:

```

304 • select request.Request_ID,vehicle.Vehicle_ID,vehicle.Vehicle_Name as vehicle ,request.Request_Quantity as quantity
305 from request left outer join vehicle
306 on request.Vehicle_Vehicle_ID=vehicle.Vehicle_ID
307 order by request.Request_Quantity asc;
308

```

Request_ID	Vehicle_ID	vehicle	quantity
4	2	Ford 300	4
6	1	Ford 20a	10
3	7	Ford MUSTANG	15
5	5	Ford ANGLIA	20
1	1	Ford 20a	50
2	4	Ford ABEILLE	75

2. Maximum requested vehicles sorted by quantity ascending
SQL QUERY and output:

```

319
320 • select demand.Demand_Id,vehicle.Vehicle_ID,vehicle.Vehicle_Name,vehicle.Model_Model_Number1,demand.Demand_Quantity
321 from demand left outer join vehicle
322 on demand.vehicle_vehicle_id=vehicle.vehicle_id
323
324 order by demand.Demand_Quantity asc;
325

```

Demand_Id	Vehicle_ID	Vehicle_Name	Model_Model_Number1	Demand_Quantity
4	2	Ford 300	2	4
6	1	Ford 20a	1	10
3	7	Ford MUSTANG	7	15
5	5	Ford ANGLIA	5	20
1	1	Ford 20a	1	50
2	4	Ford ABEILLE	4	75

3. Total amount paid by distributor to the company for demanding and requesting production of vehicle

```

367
368
369 • select payment.Payment_Id,invoice.Invoice_Amount as 'Amount Paid',distributor.D_Name as 'Paid From',
370      company.Company_Name as 'Paid to'
371 from payment inner join invoice inner join distributor inner join company
372 on payment.Invoice_Invoice_Id=invoice.Invoice_Id and payment.Distributor_D_ID=distributor.D_ID
373 and distributor.Company_Company_ID=company.Company_ID
374 order by Invoice_Amount,Payment_Id asc;

```

<				
Result Grid				
Filter Rows: <input type="text"/>				
Export: <input type="button" value="Export"/>				
Wrap Cell Content: <input type="button" value="Wrap"/>				
Payment_Id	Amount Paid	Paid From	Paid to	
4	80000	Optimum Dealers	Ford Motors	
6	90000	Speedv Dealers	Ford Motors	
3	675000	Ace Dealers	Ford Motors	
5	700000	Symphony Dealers	Ford Motors	
1	750000	Speedv Dealers	Ford Motors	
2	2250000	Star Dealers	Ford Motors	

6. Trigger

1. Code:

```
delimiter //
create trigger insert_a_cost
after insert on model
for each row
begin
insert into cost
values(75000,11);
end//
```

```
391
392 delimiter //
393 • create trigger insert_a_cost
394   after insert on model
395   for each row
396   begin
397     insert into cost
398     values(75000,11);
399   end//
400
```

✓ 302 14:51:05 create trigger insert_a_cost after insert on model for each row begin insert into cost values(75000,11);... 0 row(s) affected

Now I created a new entry into the model table

--insert into model

values('11','Figo','AUTO','HATCHBACK');

now when I see the cost table, I see a new entry with trigger specified values

SELECT * FROM COST;

	Price	Model_Model_Number
	15000	1
	60000	10
	75000	11
	20000	2
	25000	3
	30000	4
	35000	5
	40000	6
	45000	7
	50000	8
	55000	9
	NULL	NULL

7. Stored Procedure

1. To check vehicle details by input vehicle Id

```
--create procedure check_price1(IN val int)
begin
select
vehicle.Vehicle_ID,vehicle.Vehicle_Name,model.Model_Name,model.Model_
Number,cost.Price
from vehicle inner join model inner join cost
on vehicle.Model_Model_Number1=model.Model_Number and
model.Model_Number=cost.Model_Model_Number
where Vehicle_ID=val;
end;
//
```

331 15:26:11 create procedure check_price1(IN val int) begin /*declare val int ;*/ select vehicle.Vehicle_ID,veh... 0 row(s) affected

Now to call the procedure we use

- call check_price1(1);//

Here 1 is the vehicle Id and the output we get is

	Vehicle_ID	Vehicle_Name	Model_Name	Model_Number	Price
	1	Ford 20a	2GA	1	15000

8. Locking tables to avoid unwanted access on tables

1. In this project, I have only considered to have 1 company i.e FORD
It shouldn't be allowed for anyone to insert new data into the company table

CODE:

lock tables company read;

✓	341	15:53:26	lock tables company read;	0 row(s) affected	0.000 sec
---	-----	----------	---------------------------	-------------------	-----------

NOW I TRIED TO INSERT DATA INTO THE COMPANY TABLE USING

- INSERT INTO COMPANY VALUES (2, 'NISSAN' , 10);
- AND THIS IS WHAT I GOT AS AN OUTPUT

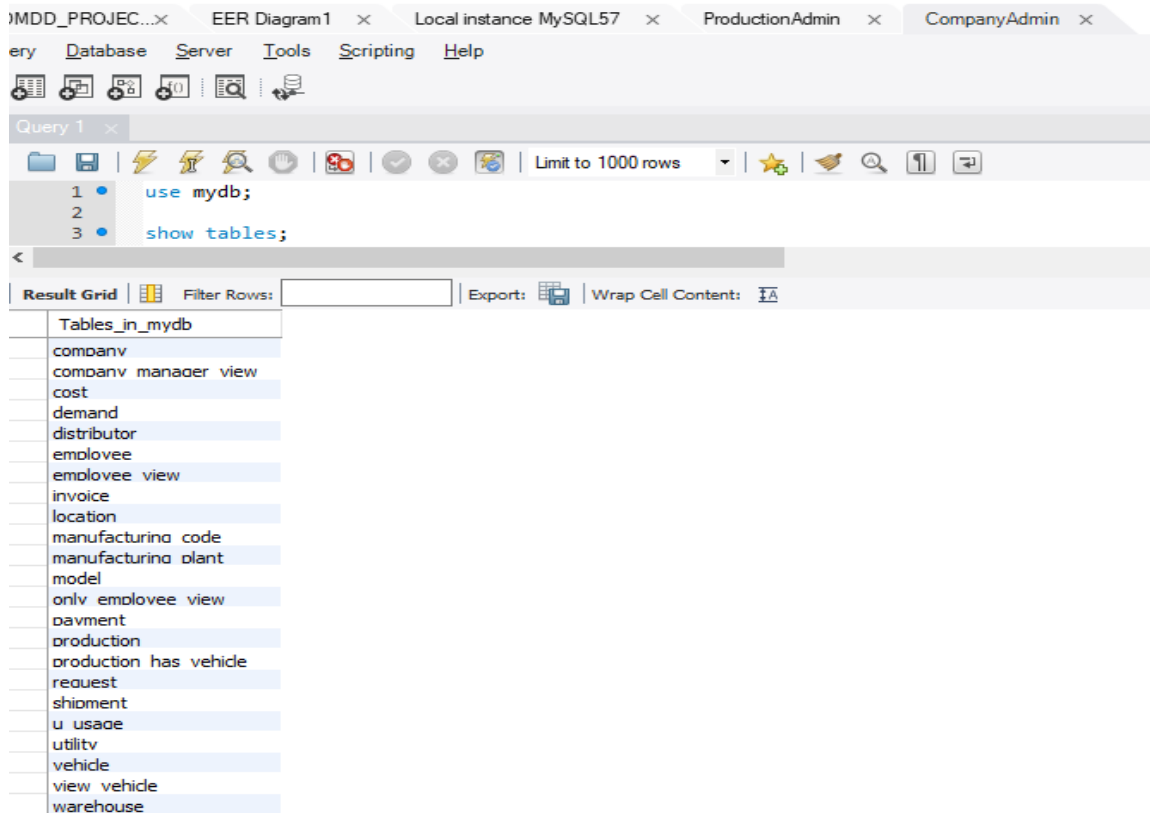
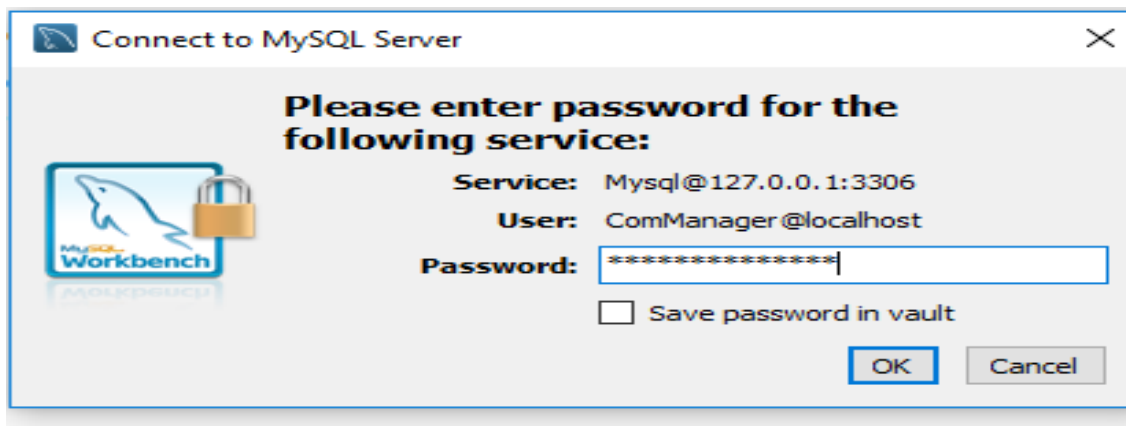
✓	341	15:53:26	lock tables company read;	0 row(s) affected
✗	342	15:58:40	insert into company values(2,'Nissan',10);	Error Code: 1099. Table 'company' was locked with a READ lock and can't be updated

WHICH MEANS THE READ LOCK PLACED ON COMPANY TABLE DID NOT ALLOW ANYTHING TO BE WRITTEN TO IT.

9. User privileges

1. Privilege Granted to Company Manager

- grant all on mydb.* to 'ComManager@localhost' @'localhost';

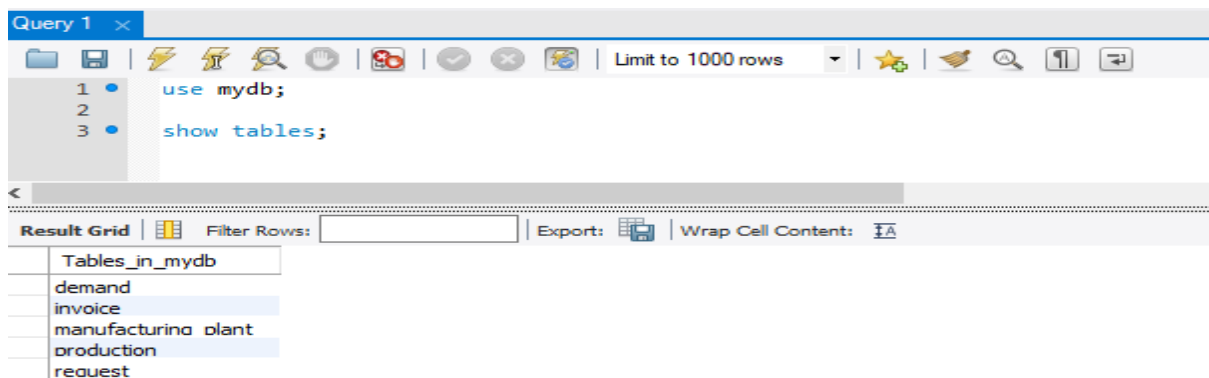
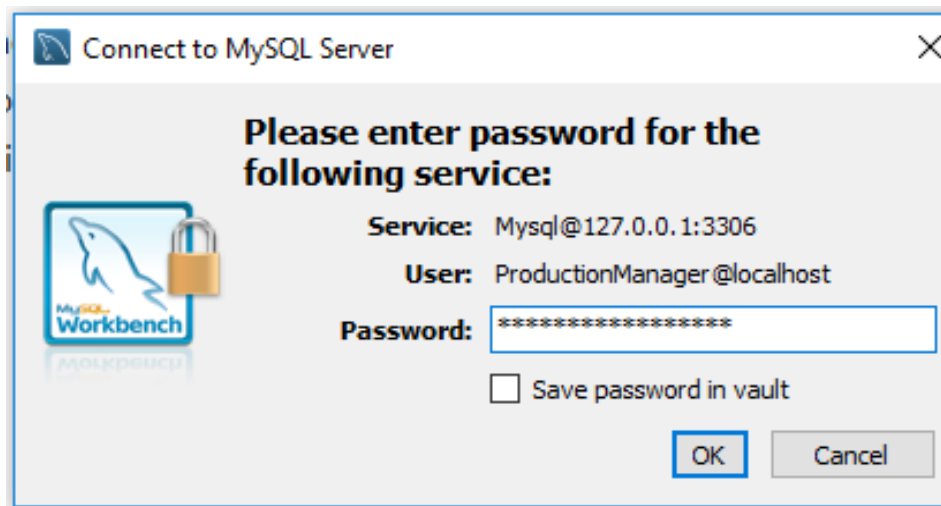


2. Privilege Granted to Company Manager

```

grant all on mydb.Production to 'ProductionManager@localhost'
@'localhost';
grant all on mydb.Manufacturing_plant to 'ProductionManager@localhost'
@'localhost';
grant all on mydb.Request to 'ProductionManager@localhost' @'localhost';
grant all on mydb.Invoice to 'ProductionManager@localhost' @'localhost';
grant all on mydb.Demand to 'ProductionManager@localhost' @'localhost';

```



3. Privilege Granted to Distributor

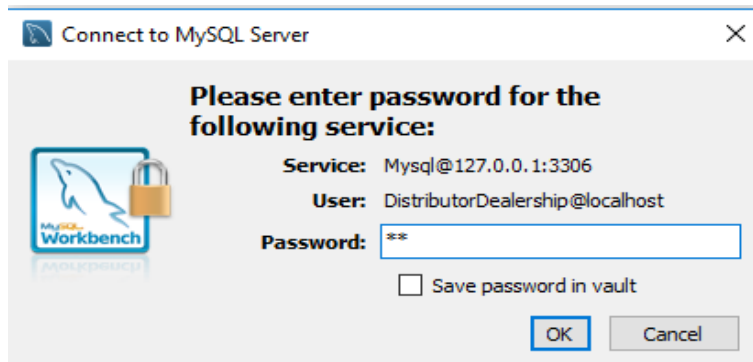
-

grant select on mydb.Manufacturing_plant to 'DistributorDealership@localhost' @'localhost';

grant select on mydb.Company to 'DistributorDealership@localhost' @'localhost';

grant select,insert on mydb.Demand to 'DistributorDealership@localhost' @'localhost';

grant select,insert on mydb.Request to 'DistributorDealership@localhost' @'localhost';



Query 1 x

1 use mydb;
2
3 show tables;

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Tables_in_mydb
company
demand
manufacturing plant
payment
request
shipment

Result 1 x

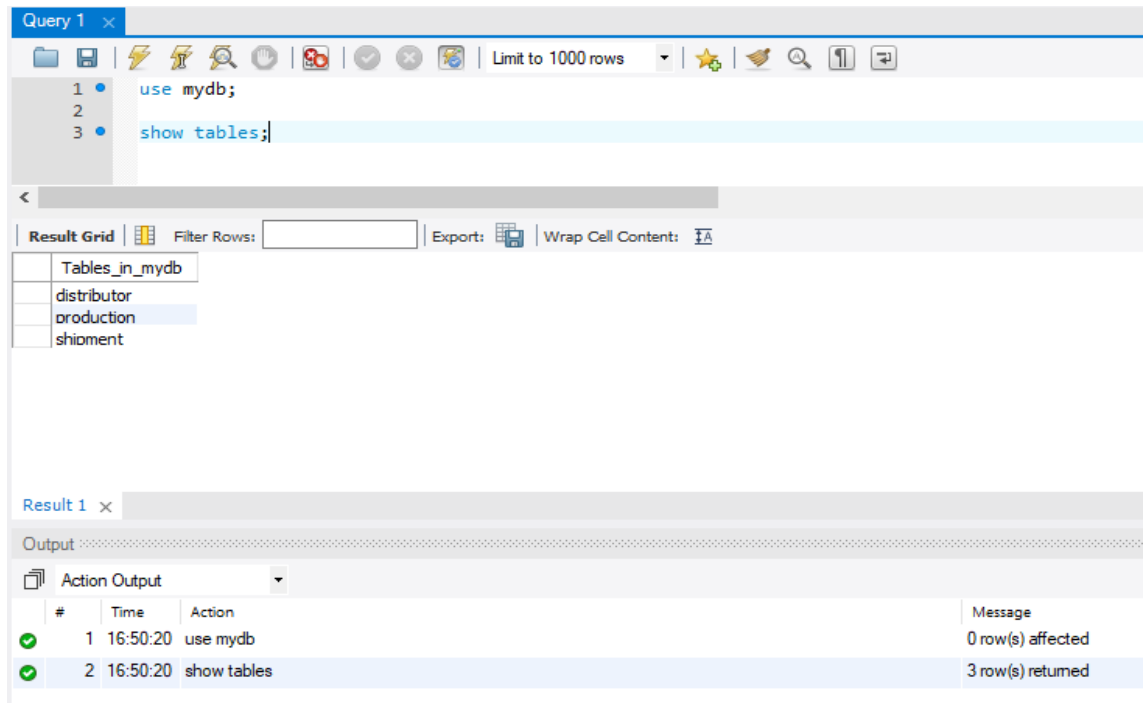
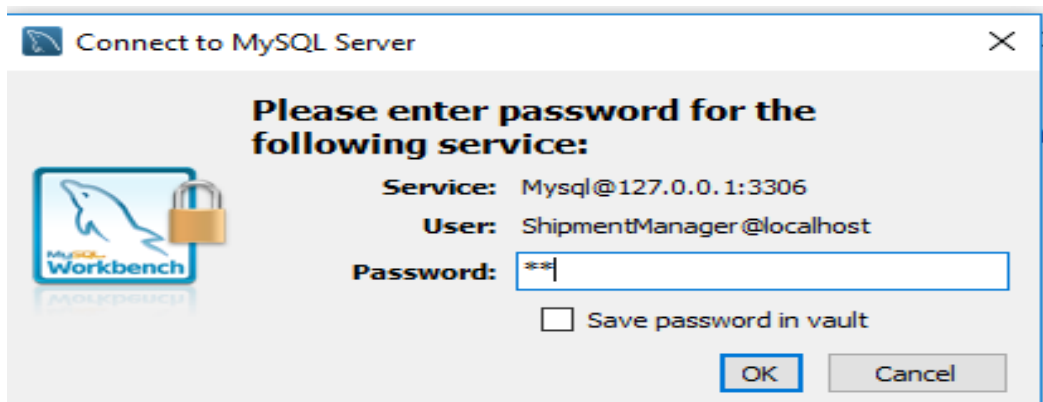
Output

Action Output

#	Time	Action	Message
✓ 1	16:48:12	use mydb	0 row(s) affected
✓ 2	16:48:17	show tables	6 row(s) returned

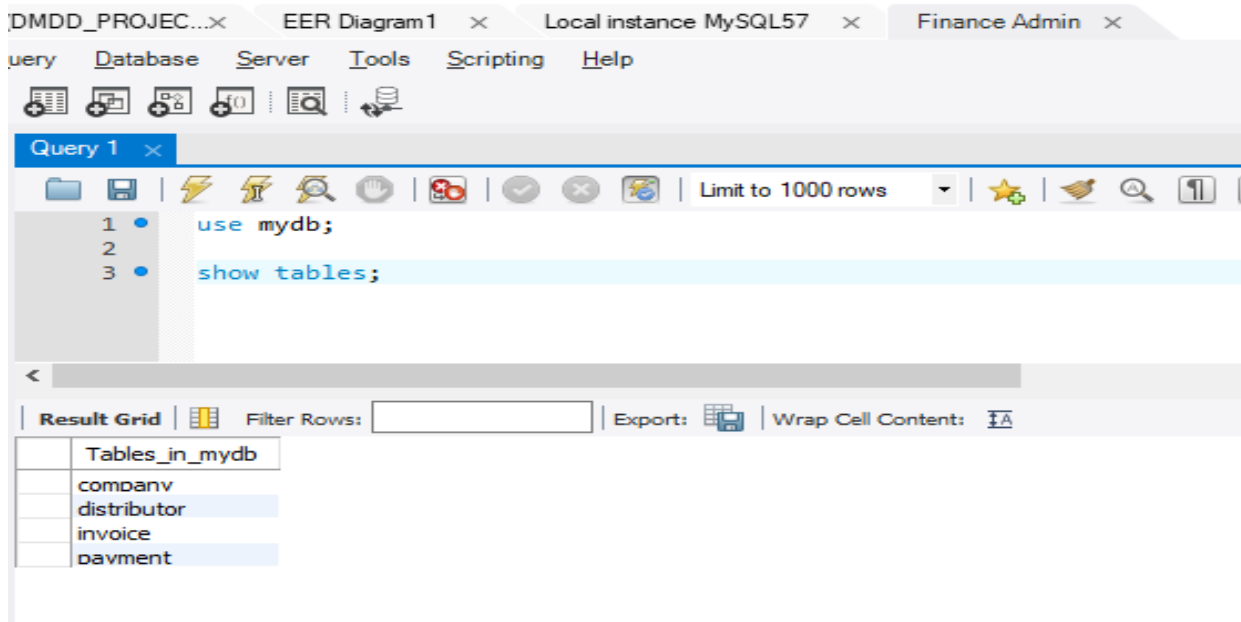
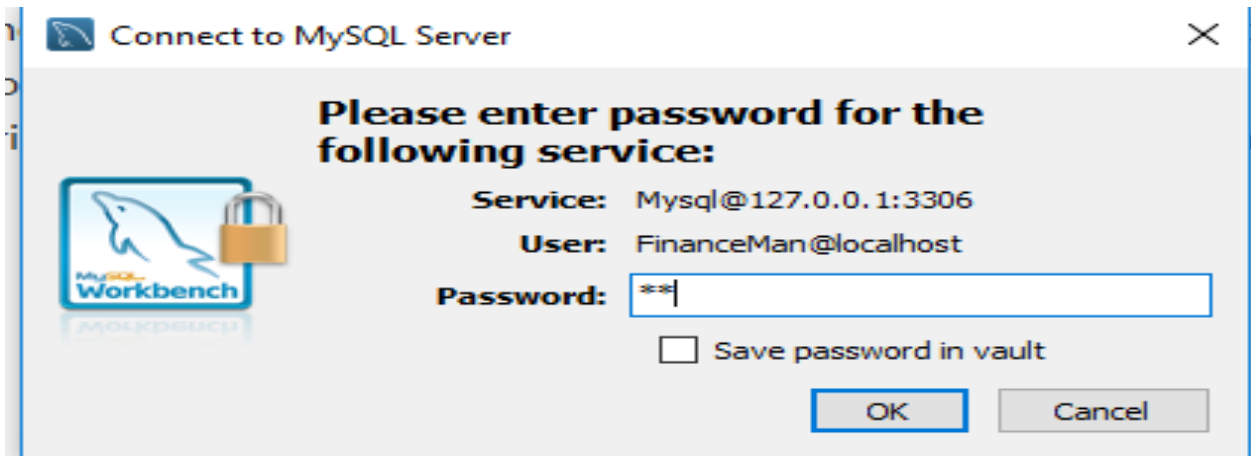
4. Privilege Granted to Shipment Manager

```
grant select,insert,update,delete on mydb.shipment to
'ShipmentManager@localhost' @'localhost';
grant select on mydb.production to 'ShipementManager@localhost'
@'localhost';
grant select on mydb.distributor to 'ShipementManager@localhost'
@'localhost';
```



5. Privilege Granted to Finance Manager

- grant select,insert,update,delete on mydb.payment to
'FinanceMan@localhost' @'localhost';
grant select on mydb.distributor to 'FinanceMan@localhost' @'localhost';
grant select,insert on mydb.invoice to 'FinanceMan@localhost' @'localhost';
grant select on mydb.company to 'FinanceMan@localhost' @'localhost';



6. Privilege Granted to Employee

- grant select,insert on mydb.employee to 'Emp@localhost' @'localhost';
grant select on mydb.Manufacturing_plant to 'Emp@localhost' @'localhost';
grant select on mydb.company to 'Emp@localhost' @'localhost';

