**Database Management & Database Design**

**Final Project Proposal**

**Name:** Raj R Phadke

**NUID:** 001814155

1. **Topic:**

Database for an Automobile Manufacturing Unit (Factory)

1. **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

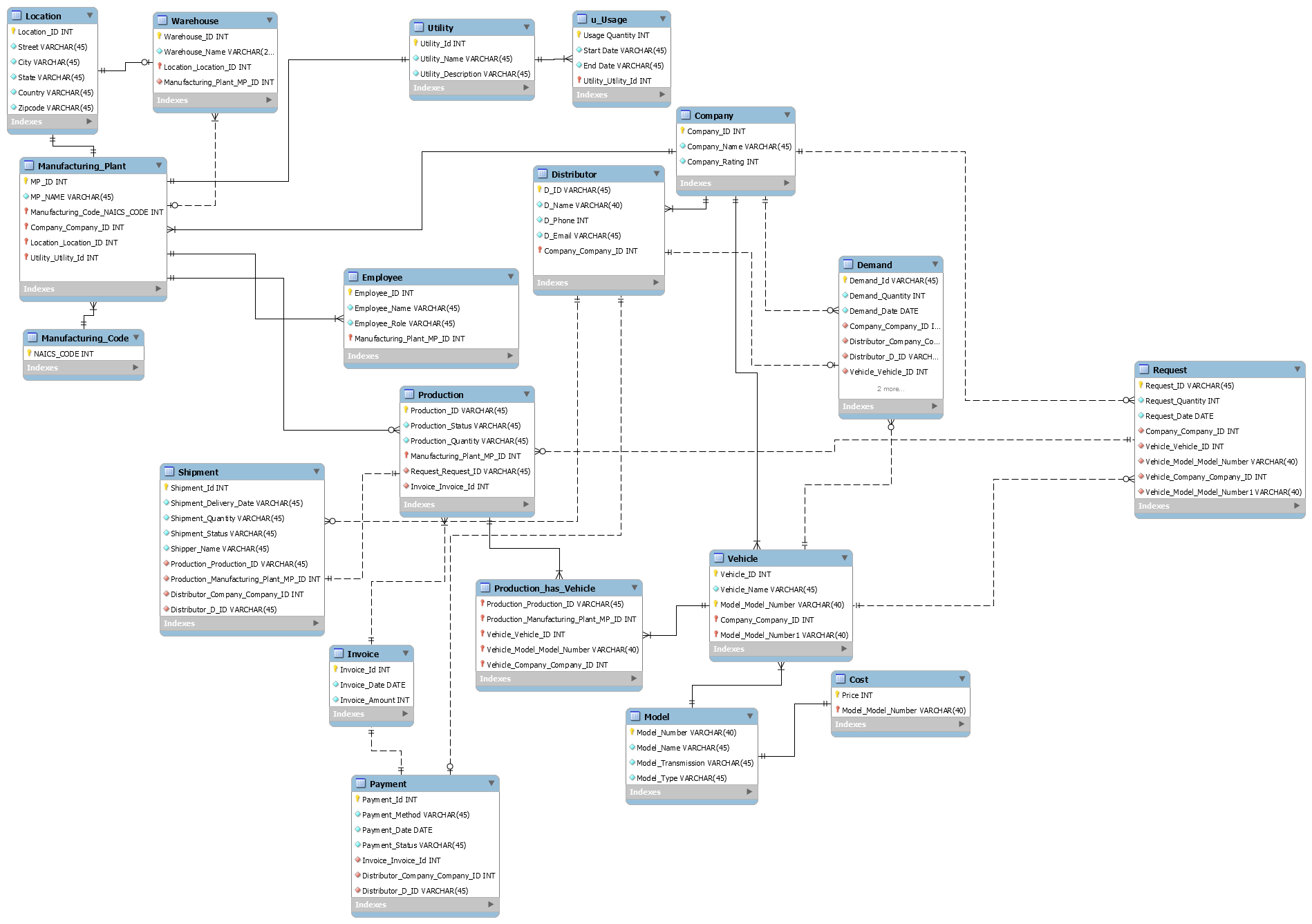
* + - Triggers
    - Stored Procedures
    - Access Control
    - Views etc.

1. **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.
2. **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.

* 1. Demands are based on a vehicle and a vehicle may have zero or many demands.
  2. In case of need, a company may send zero or many requests for production but every request has to mandatory come from a company.
  3. If there are no requests, there won’t be any production. If requests exist, there may be many productions.
  4. Every vehicle is mandatory based on one model and make. Whereas there can be multiple vehicles which are based on the same model.
  5. Every model has a cost associated with it and the cost is selective only to that model.
  6. 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.
  7. Production makes the need of shipment and every shipment is based on one production.
  8. Invoice is associated to a payment and every payment has one invoice based on which the payment is made.
  9. 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).
  10. 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).

1. **Final ER diagram**



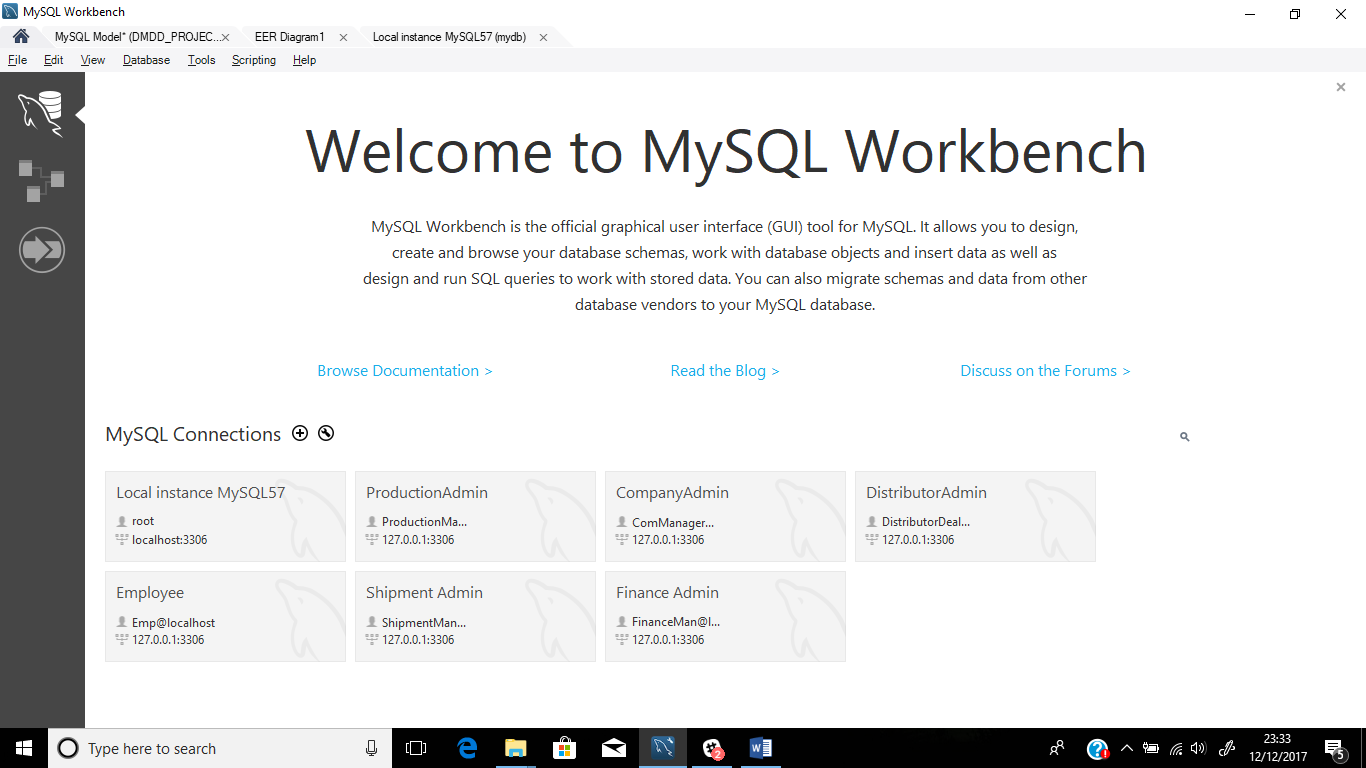
**Implementation**

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.**



1. **Creating users and their passwords**

**SQL CODE:**

-create user 'ProductionManager@localhost' @'localhost' identified by 'productionManager';

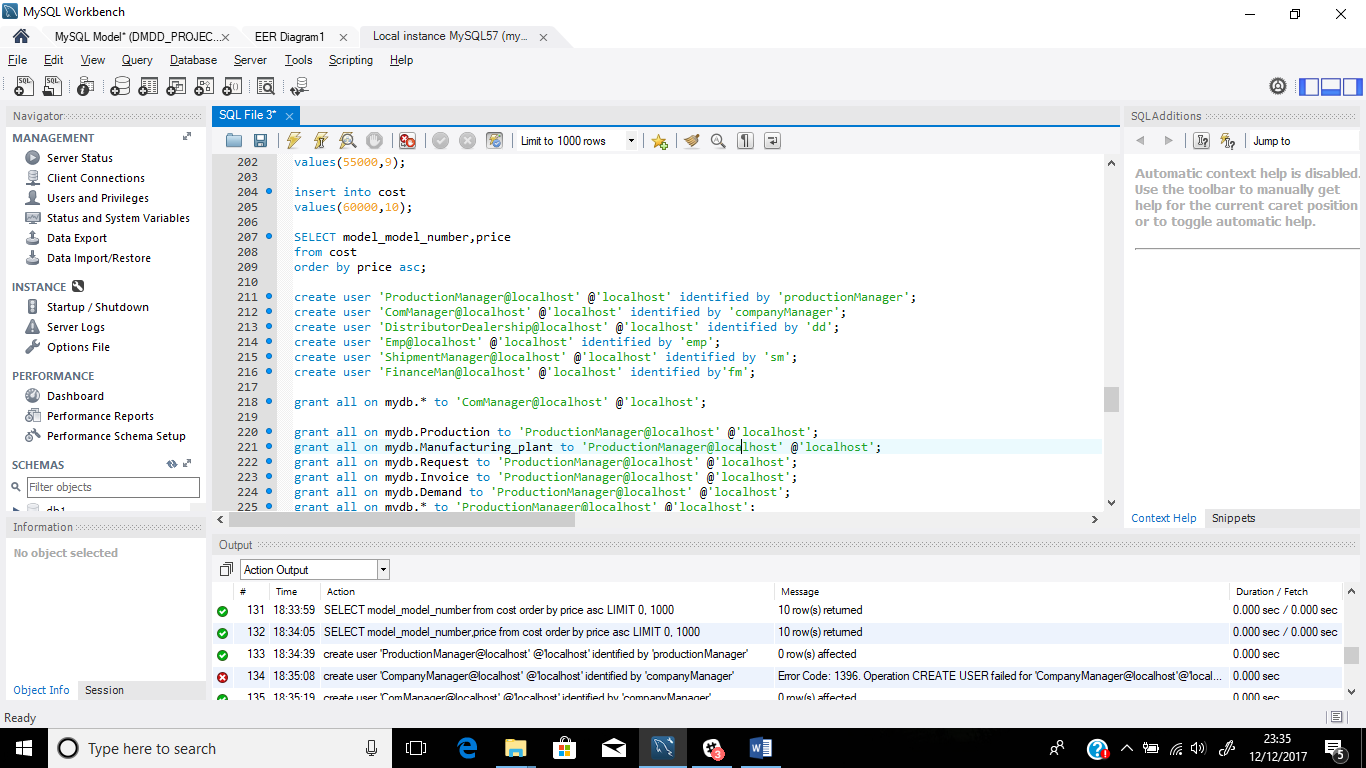
-create user 'ComManager@localhost' @'localhost' identified by 'companyManager';

-create user 'DistributorDealership@localhost' @'localhost' identified by 'dd';

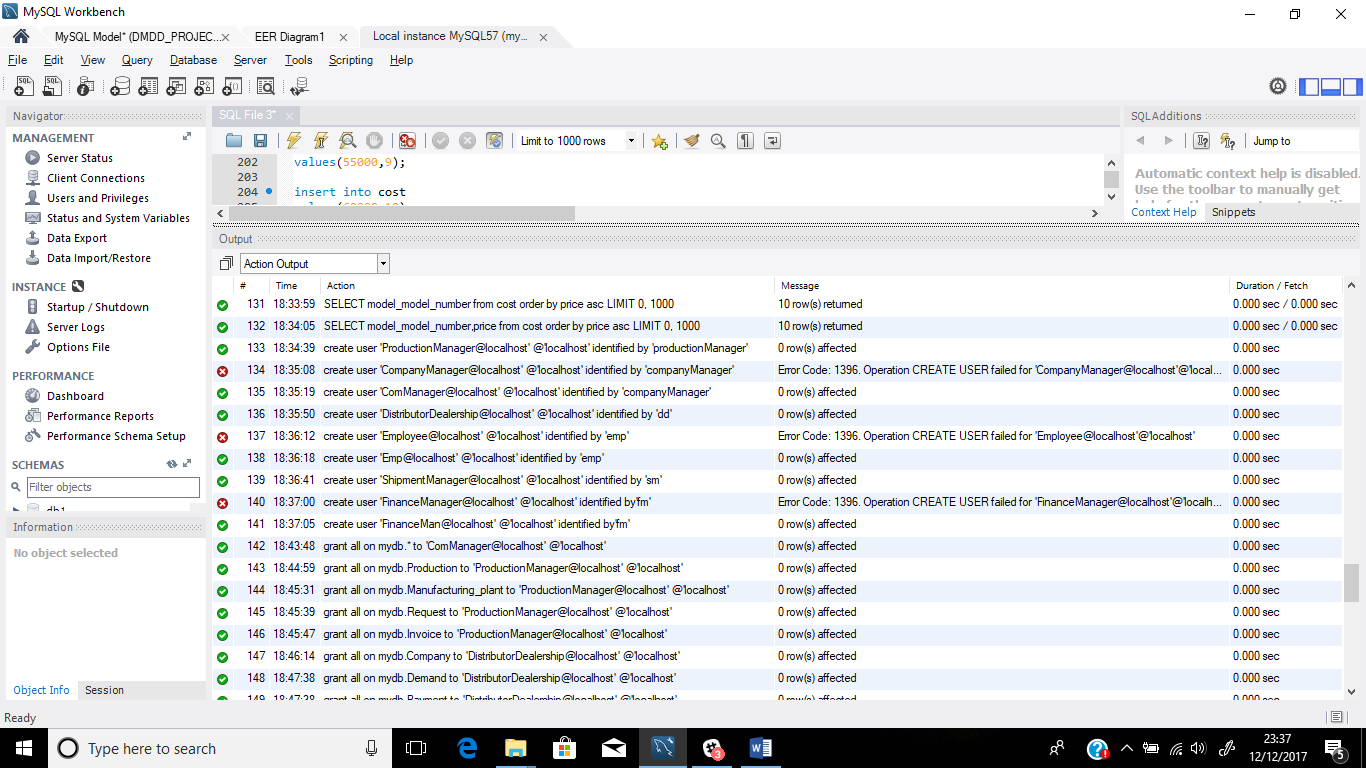
-create user 'Emp@localhost' @'localhost' identified by 'emp';

-create user 'ShipmentManager@localhost' @'localhost' identified by 'sm';

-create user 'FinanceMan@localhost' @'localhost' identified by'fm'**;**



**OUTPUT:**



1. **Using subquery to insert data**

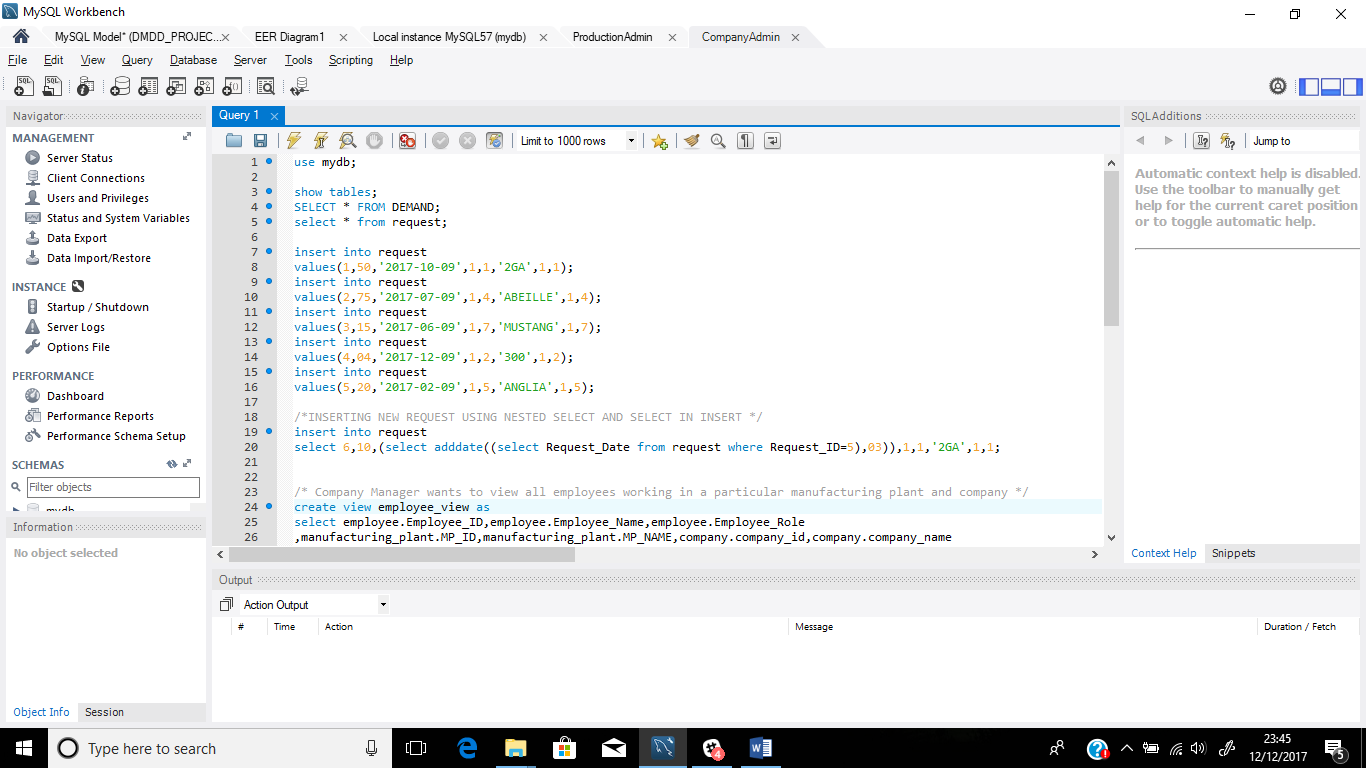
**1. Nested query and select subqueries to insert data into the request table**

**-**insert into request

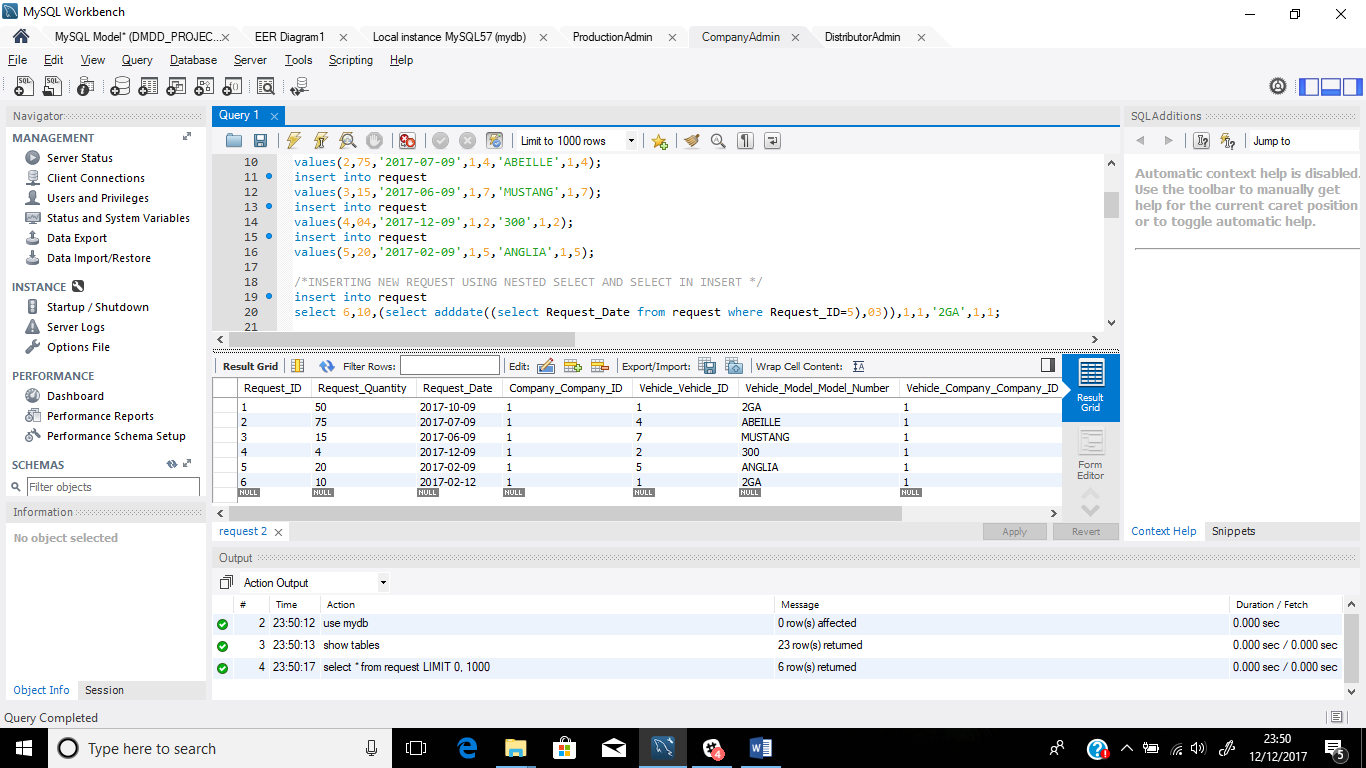
select 6, 10,

(select adddate ( ( select Request\_Date from request where Request\_ID = 5 ) , 03 ) ),

1, 1, '2GA', 1, 1;



**Output table:**

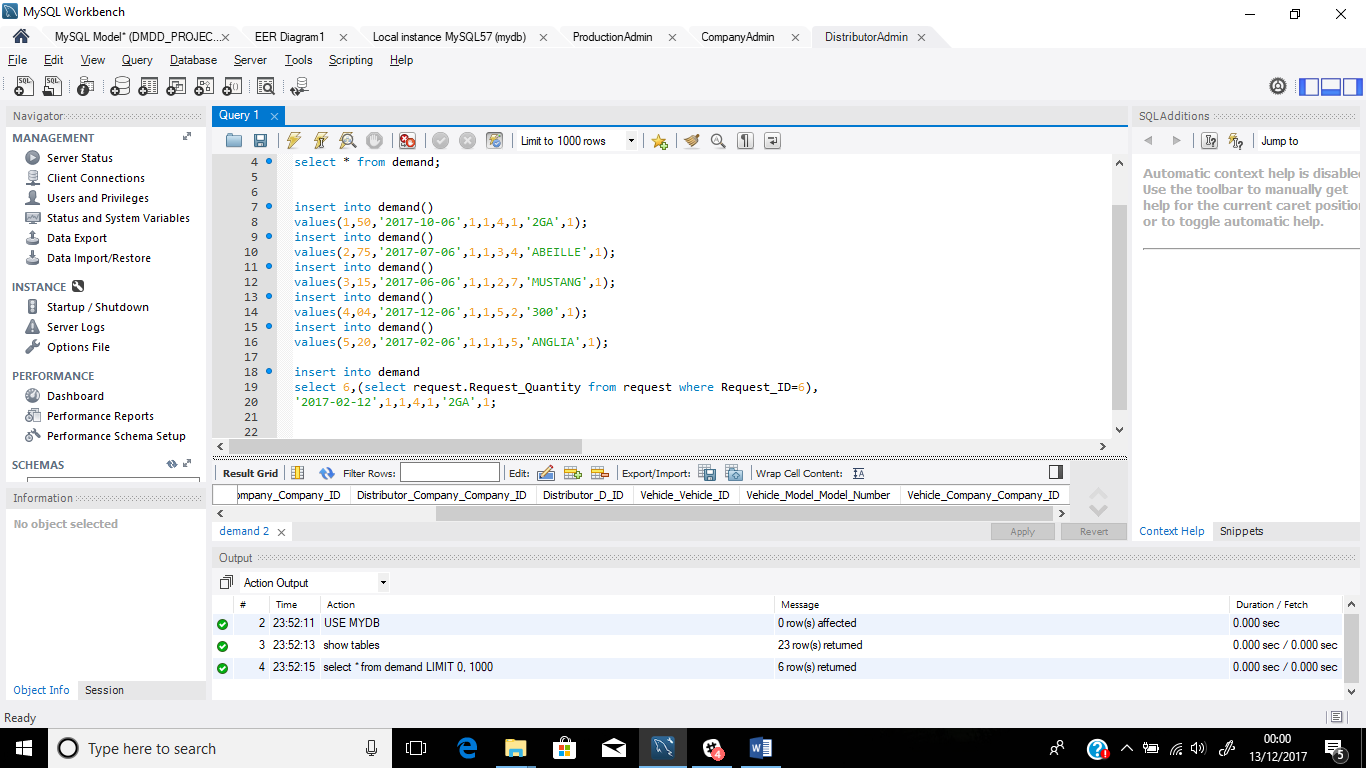


**2. Entering data into the demand table using nested query and select subquery**

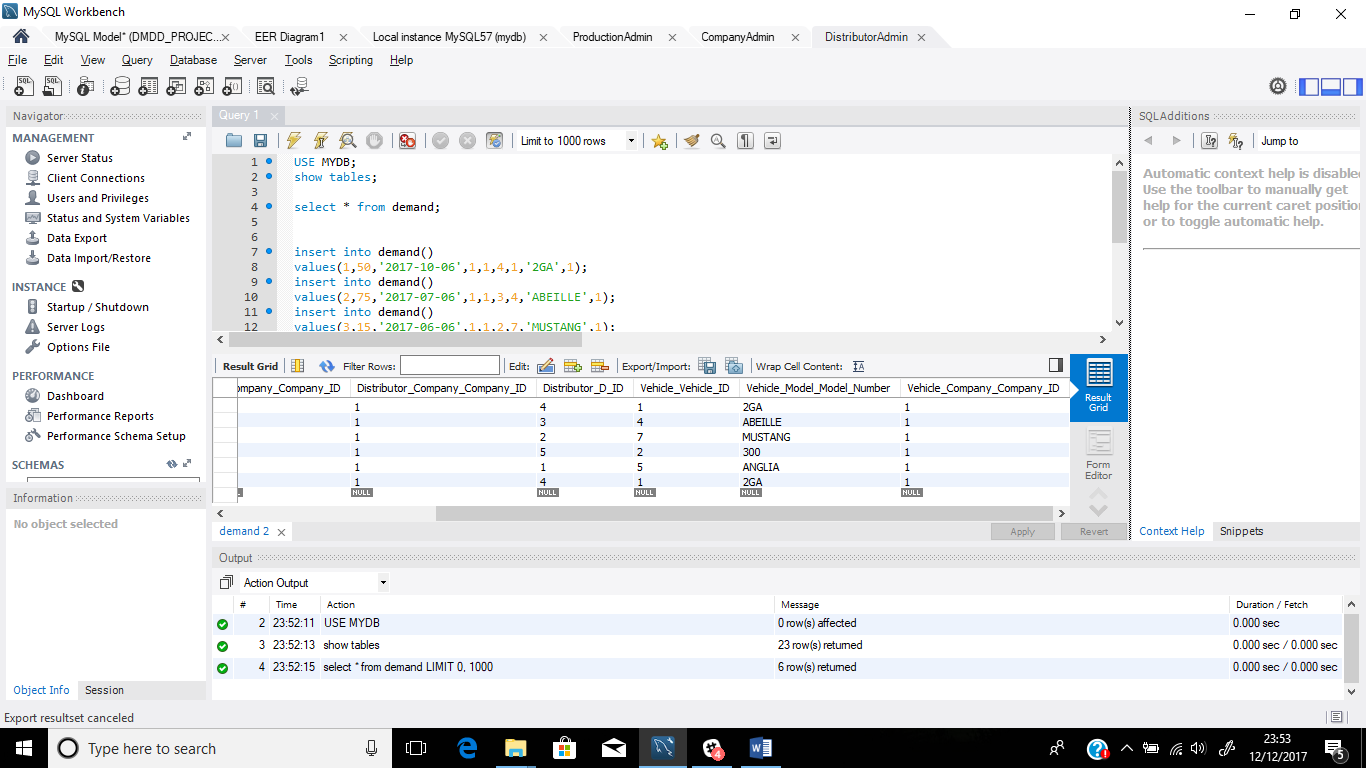
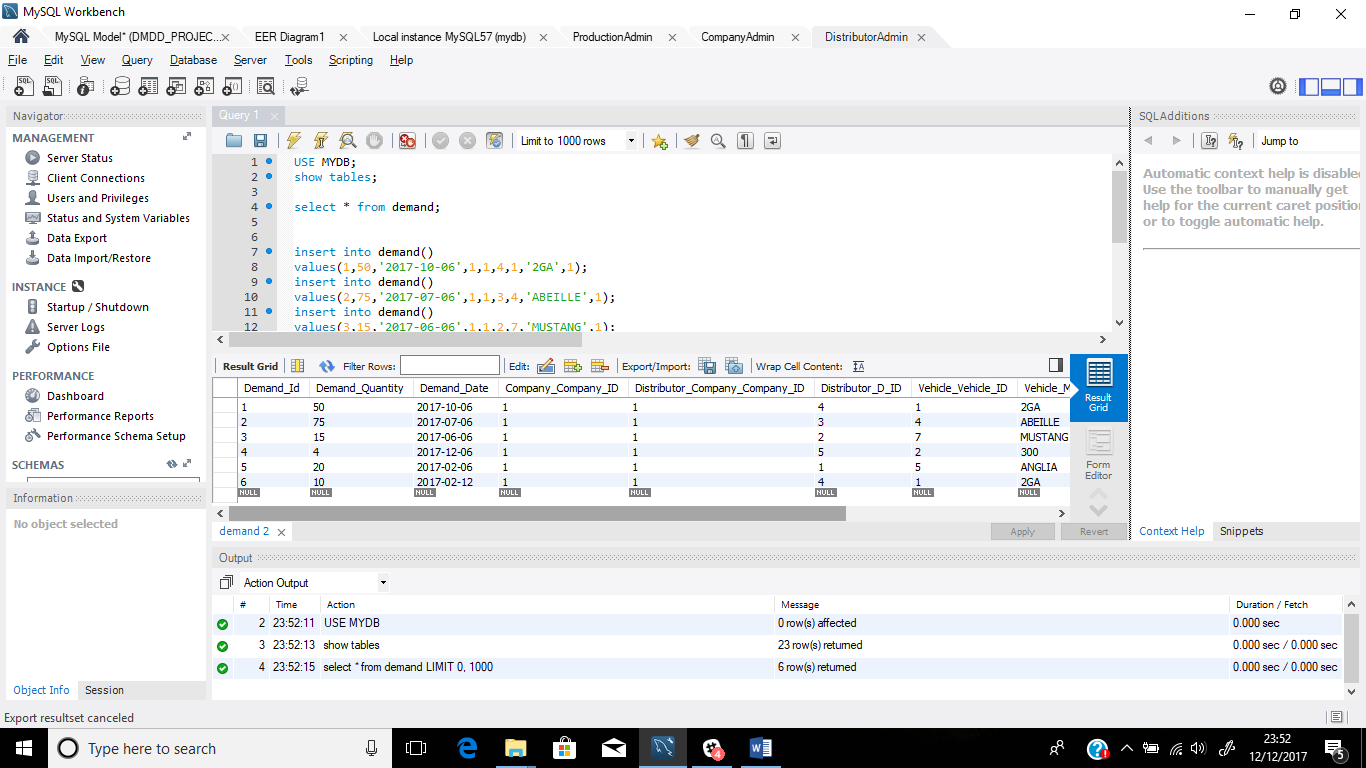
**-**insert into demand

select 6 , (select request.Request\_Quantity from request where Request\_ID=6),

'2017-02-12',1,1,4,1,'2GA',1;



**Output:**



1. **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))

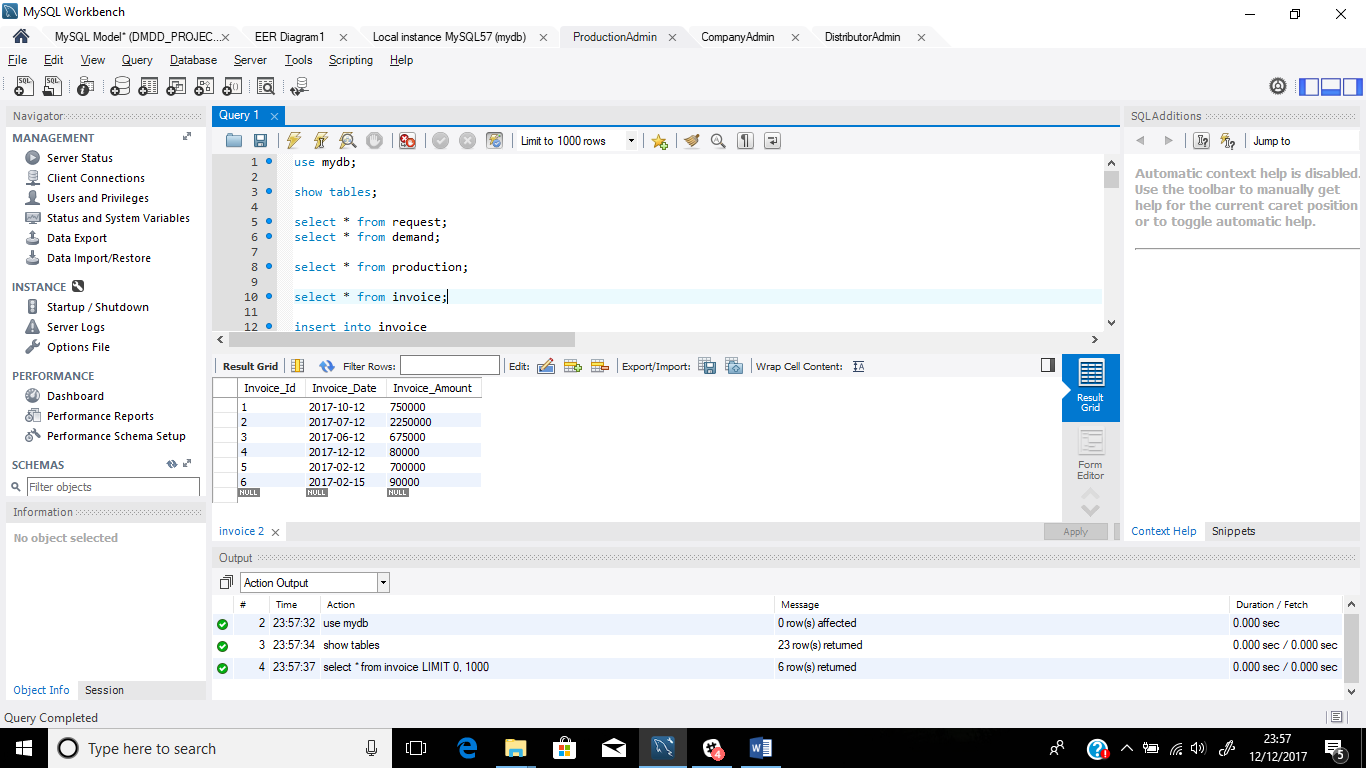
,700000;

-insert into invoice

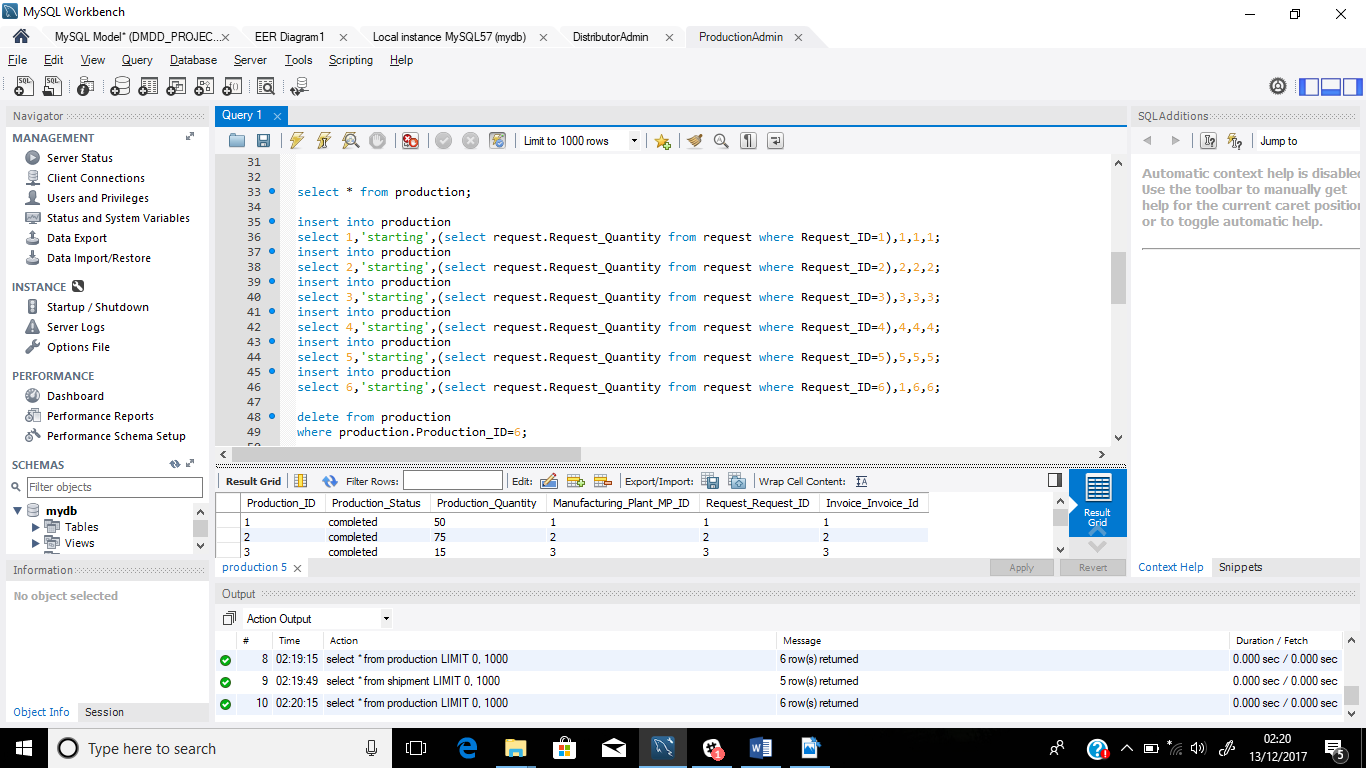
select 6,(select adddate((Select request.Request\_Date from request where Request\_ID=6),03))

,90000;

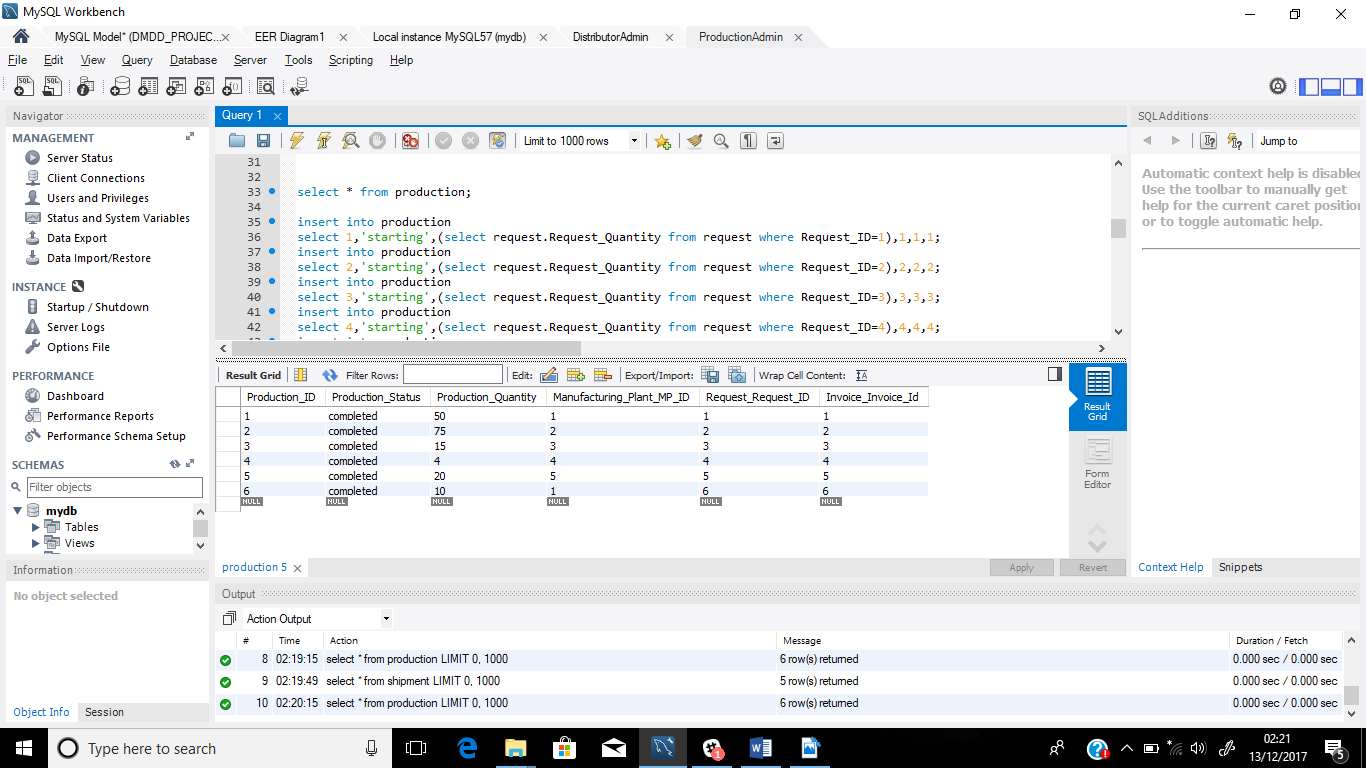
**Output:**



1. **Inserting data into the production table using subqueries**



**Output:**



1. **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),

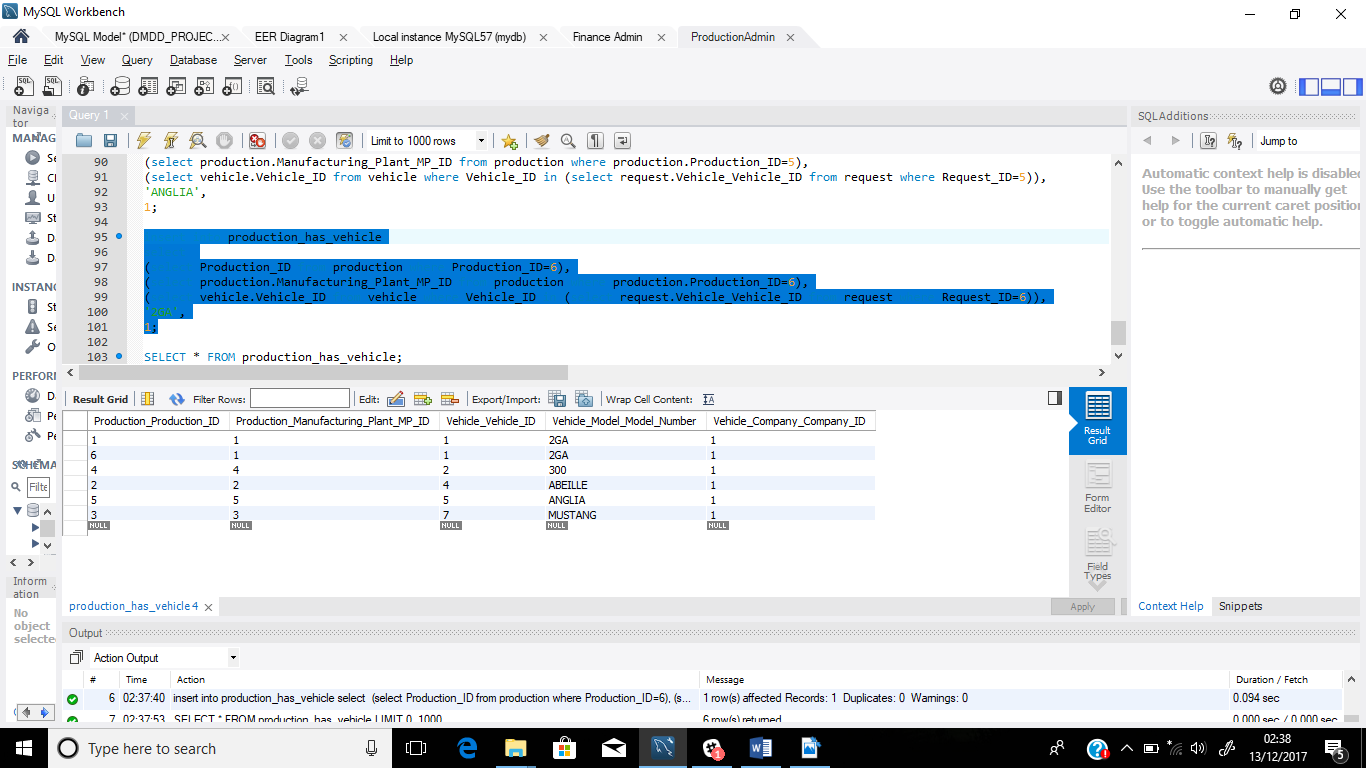
(select production.Manufacturing\_Plant\_MP\_ID from production where production.Production\_ID=6),

(select vehicle.Vehicle\_ID from vehicle where Vehicle\_ID in (select request.Vehicle\_Vehicle\_ID from request where Request\_ID=6)),

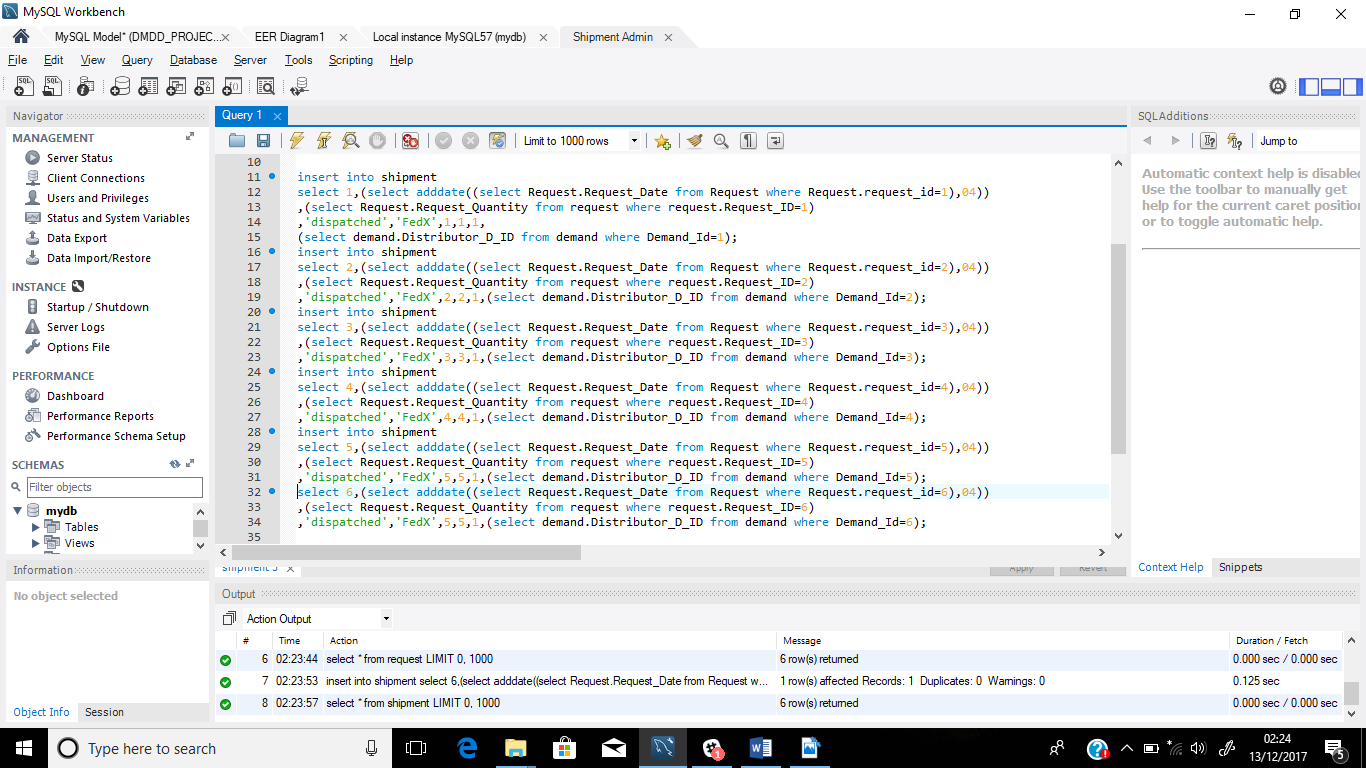
'2GA',

1;

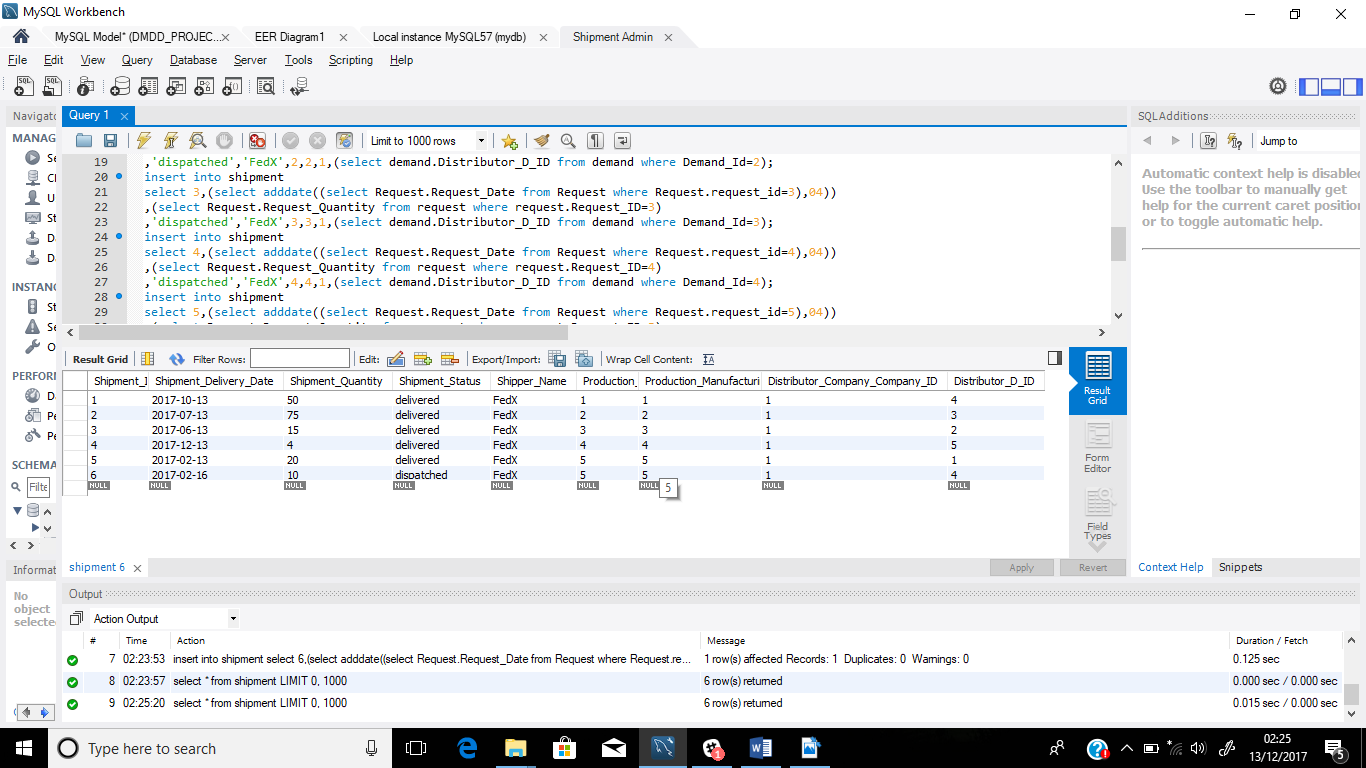
**Output:**



1. **Adding data to the shipment table using subqueries**



**Output:**



1. **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),

(select shipment.Distributor\_D\_ID from shipment where shipment.Shipment\_Id=5);

-- insert into payment

select 6,'Cheque',

(Select adddate((Select request.Request\_Date from request where Request\_ID=6),10)),

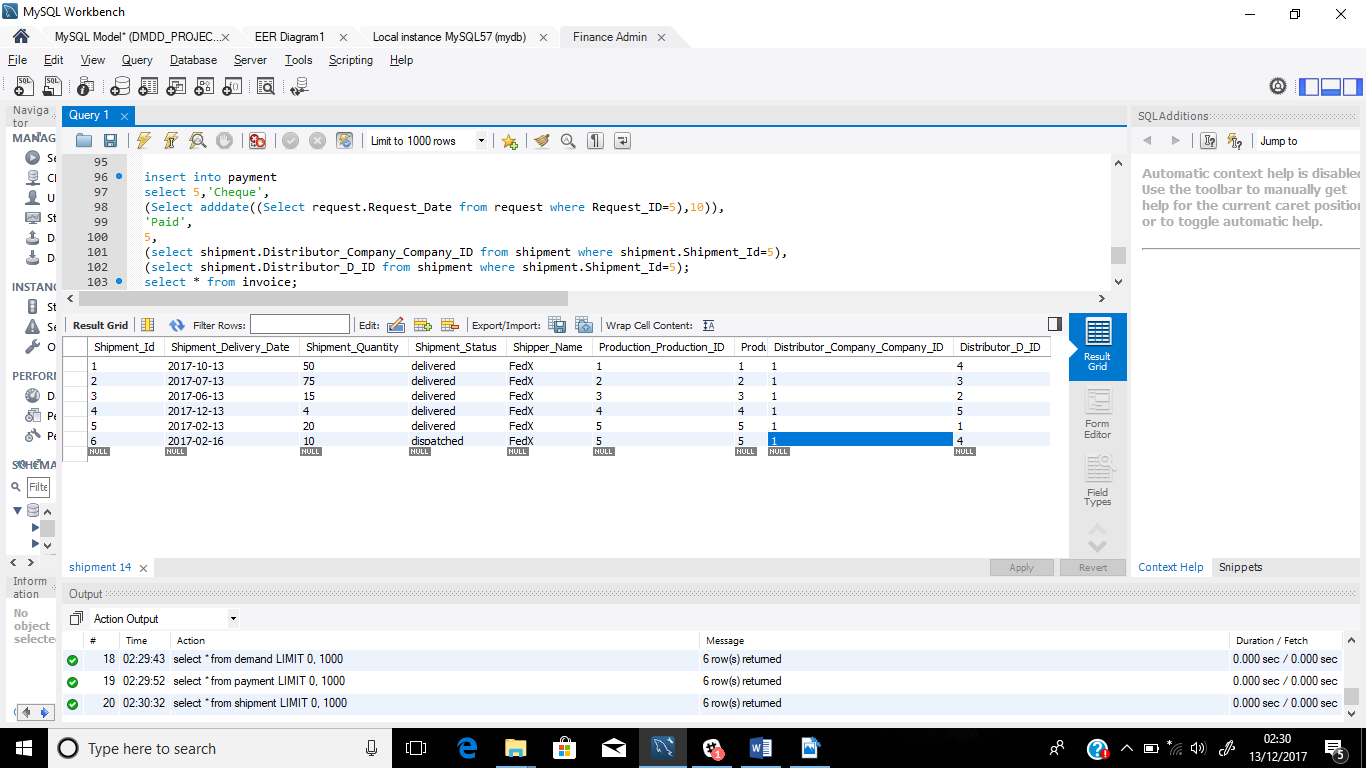
'Paid',

6,

(select shipment.Distributor\_Company\_Company\_ID from shipment where shipment.Shipment\_Id=6),

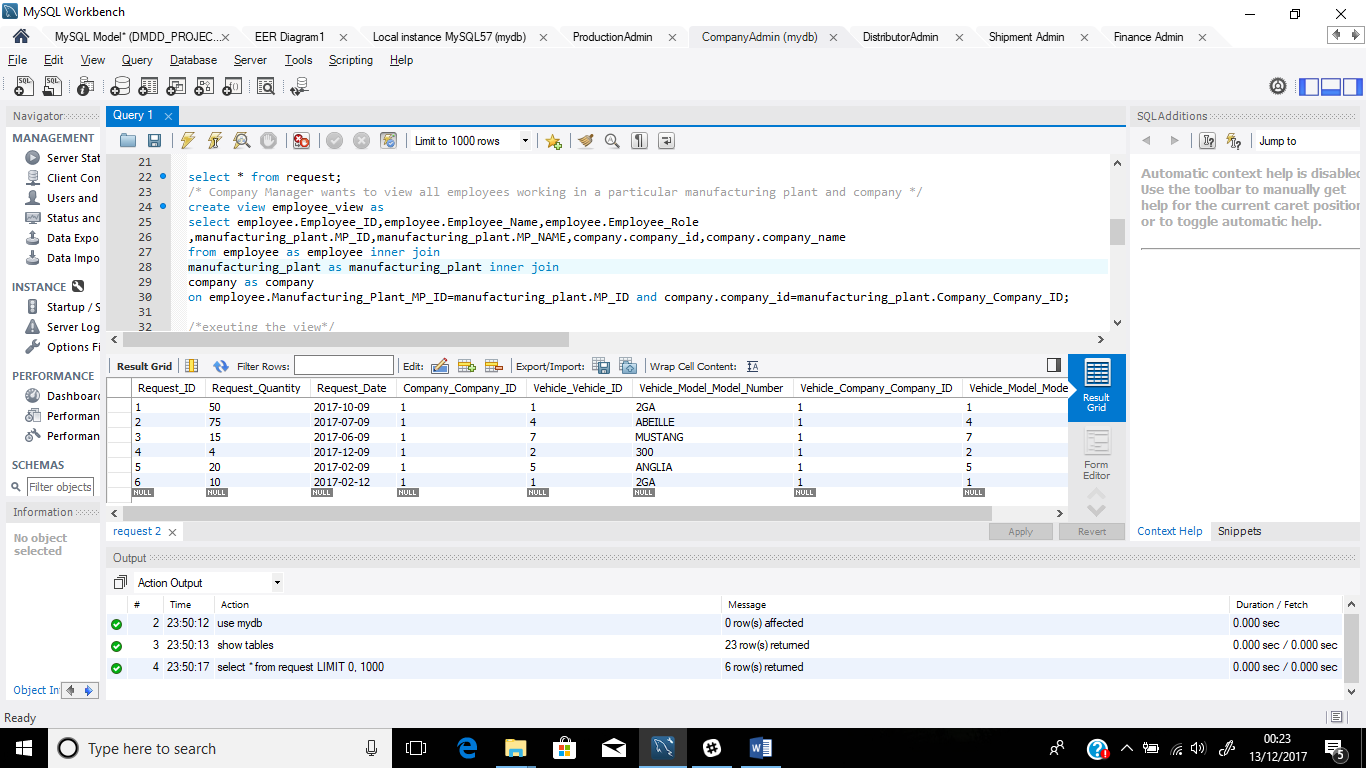
(select shipment.Distributor\_D\_ID from shipment where shipment.Shipment\_Id=6);

**Output:**



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

SQL query:



--

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;

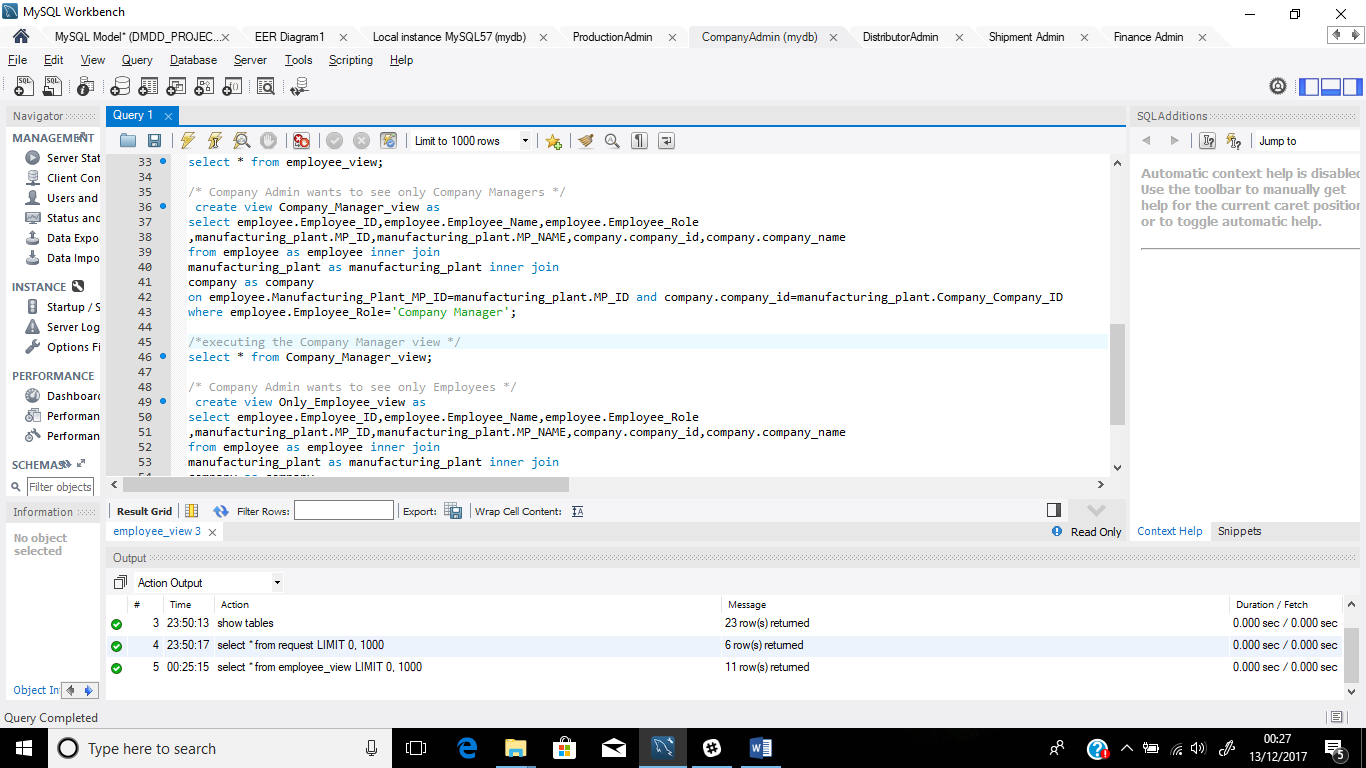
**Output**:

**Executing the view as**:

select \* from employee\_view;



1. View to display 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

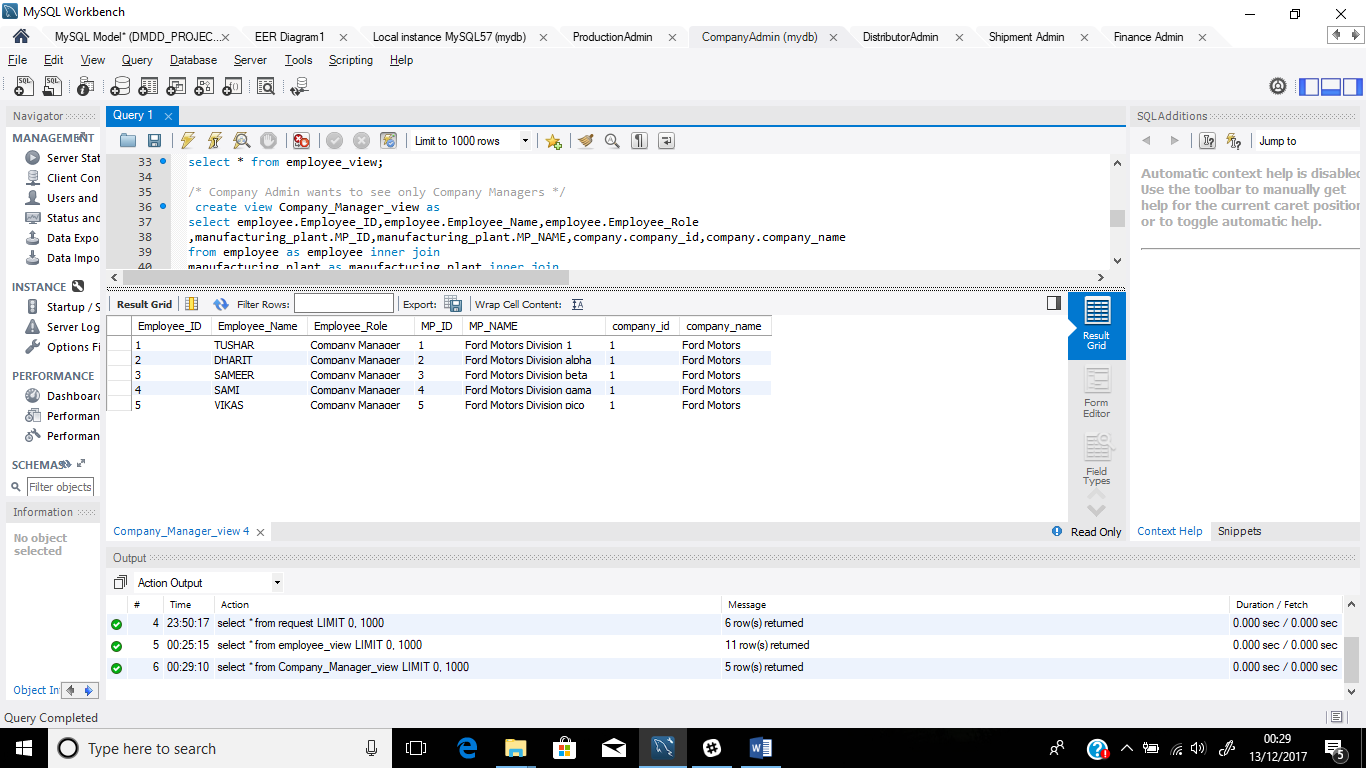
on employee.Manufacturing\_Plant\_MP\_ID=manufacturing\_plant.MP\_ID and company.company\_id=manufacturing\_plant.Company\_Company\_ID

where employee.Employee\_Role='Company Manager';

**Output:**

**Executing the view as**

select \* from Company\_Manager\_view;



1. View to display all the vehicles available

--

create view all\_vehicle

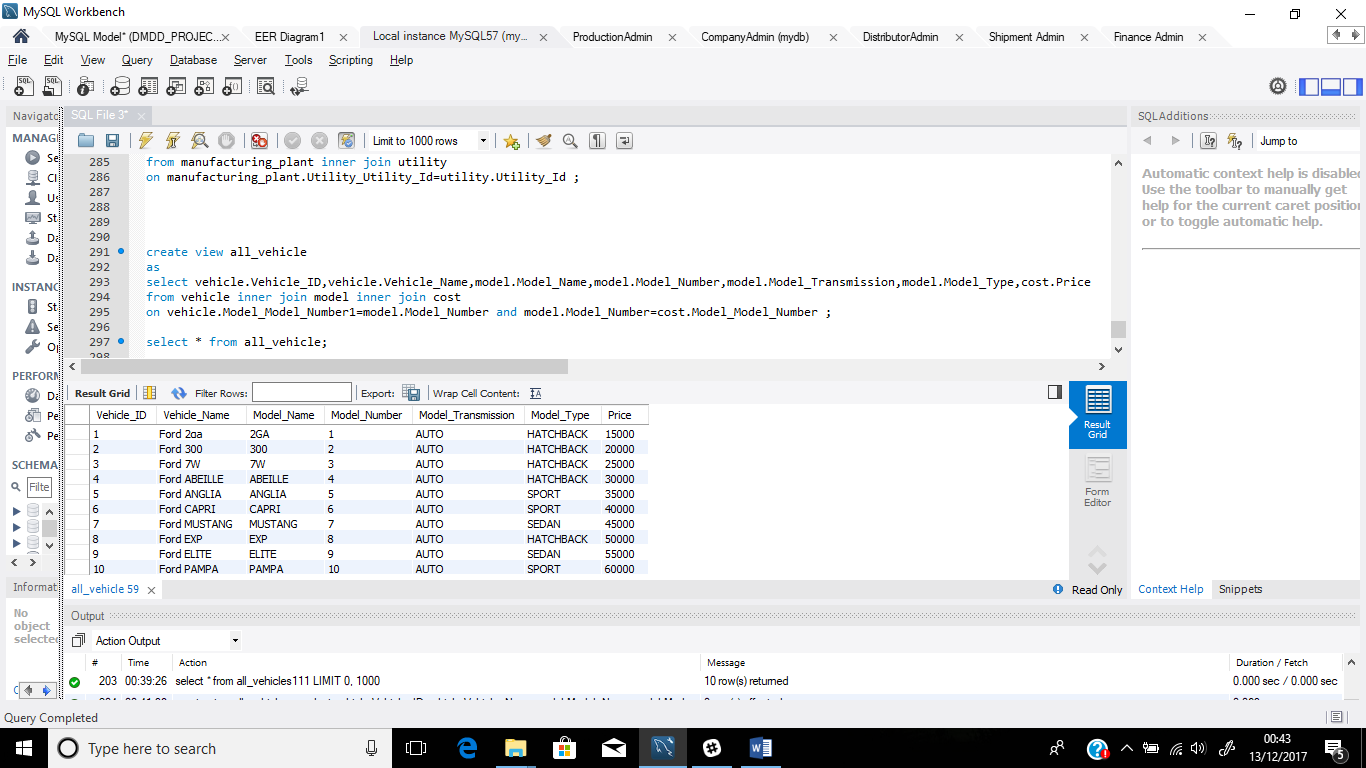
as

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

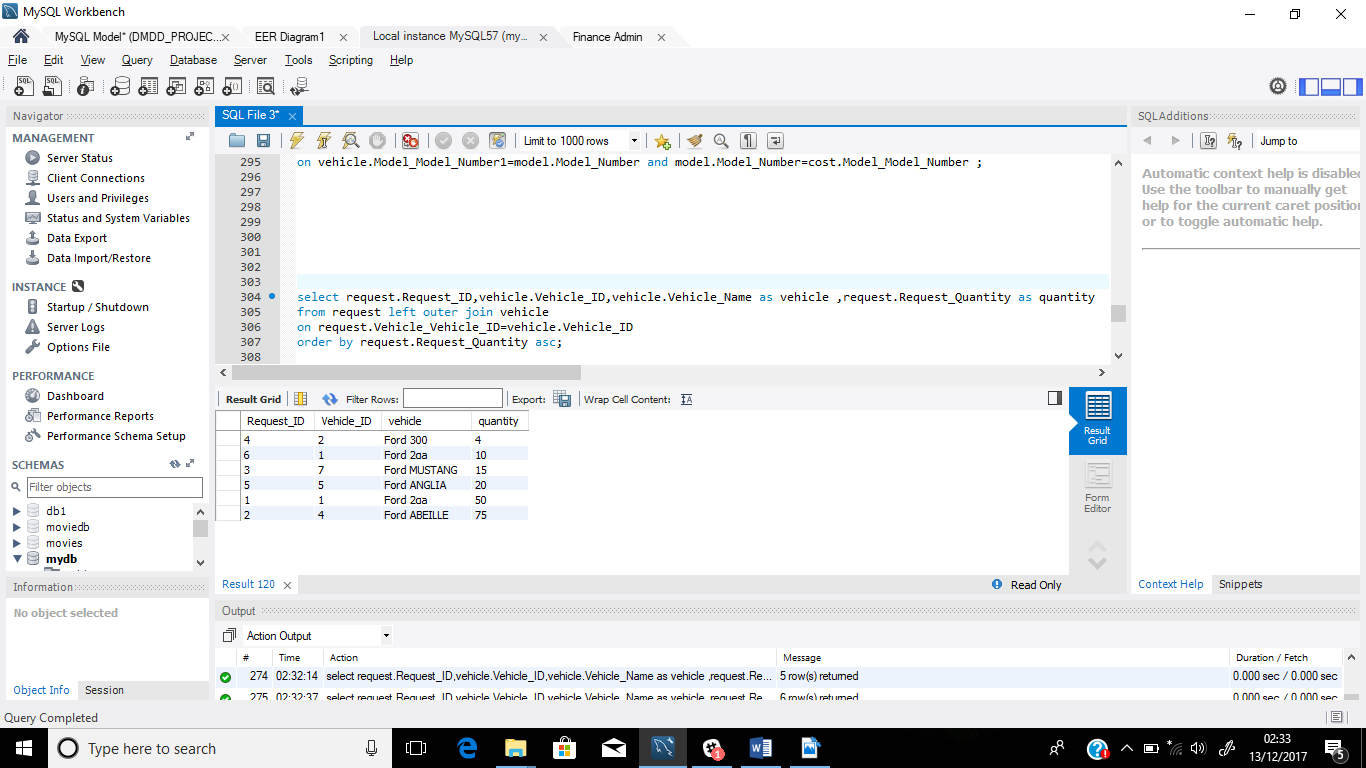
on vehicle.Model\_Model\_Number1=model.Model\_Number and model.Model\_Number=cost.Model\_Model\_Number ;

**output:**



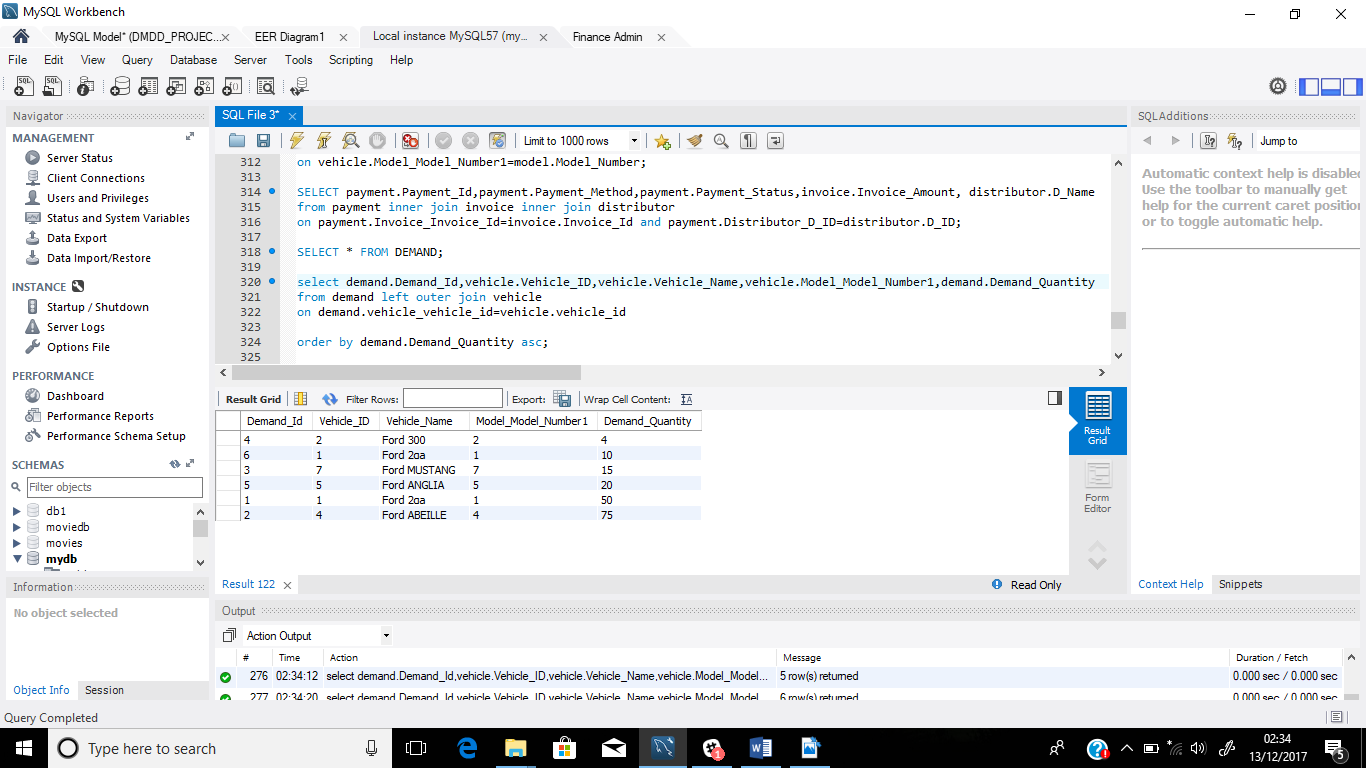
1. **Other part**
2. Maximum requested vehicles sorted by quantity ascending(not grouped by)

SQL QUERY and output:

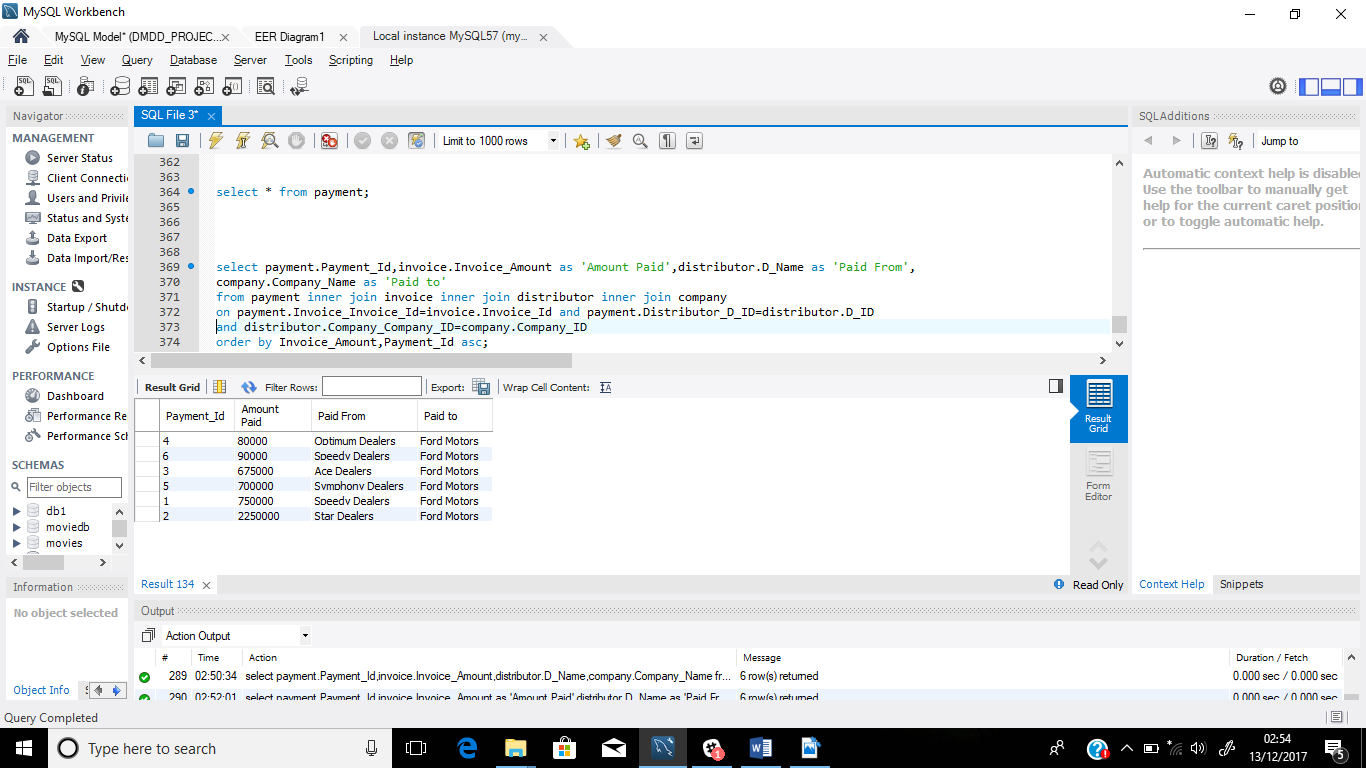


1. Maximum requested vehicles sorted by quantity ascending

SQL QUERY and output:



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



1. **Trigger**
2. Code:

delimiter //

create trigger insert\_a\_cost

after insert on model

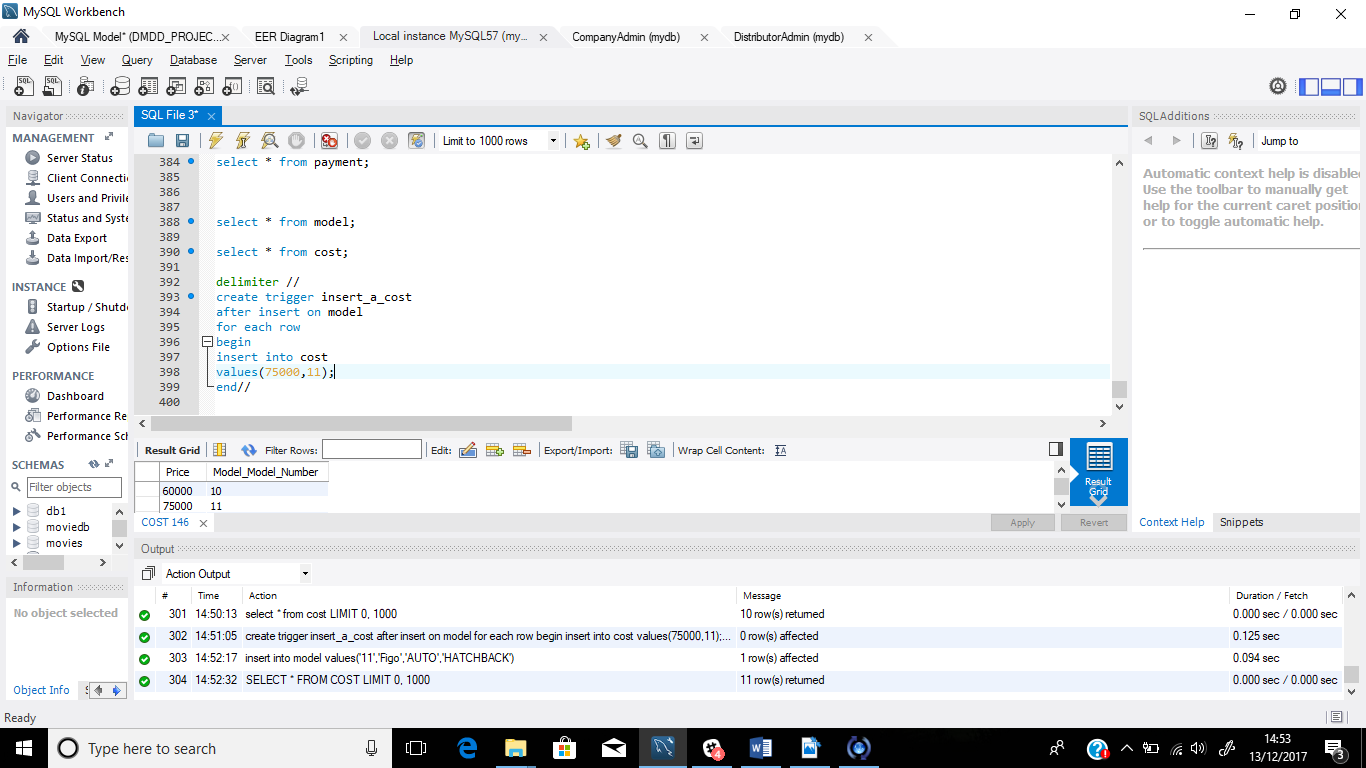
for each row

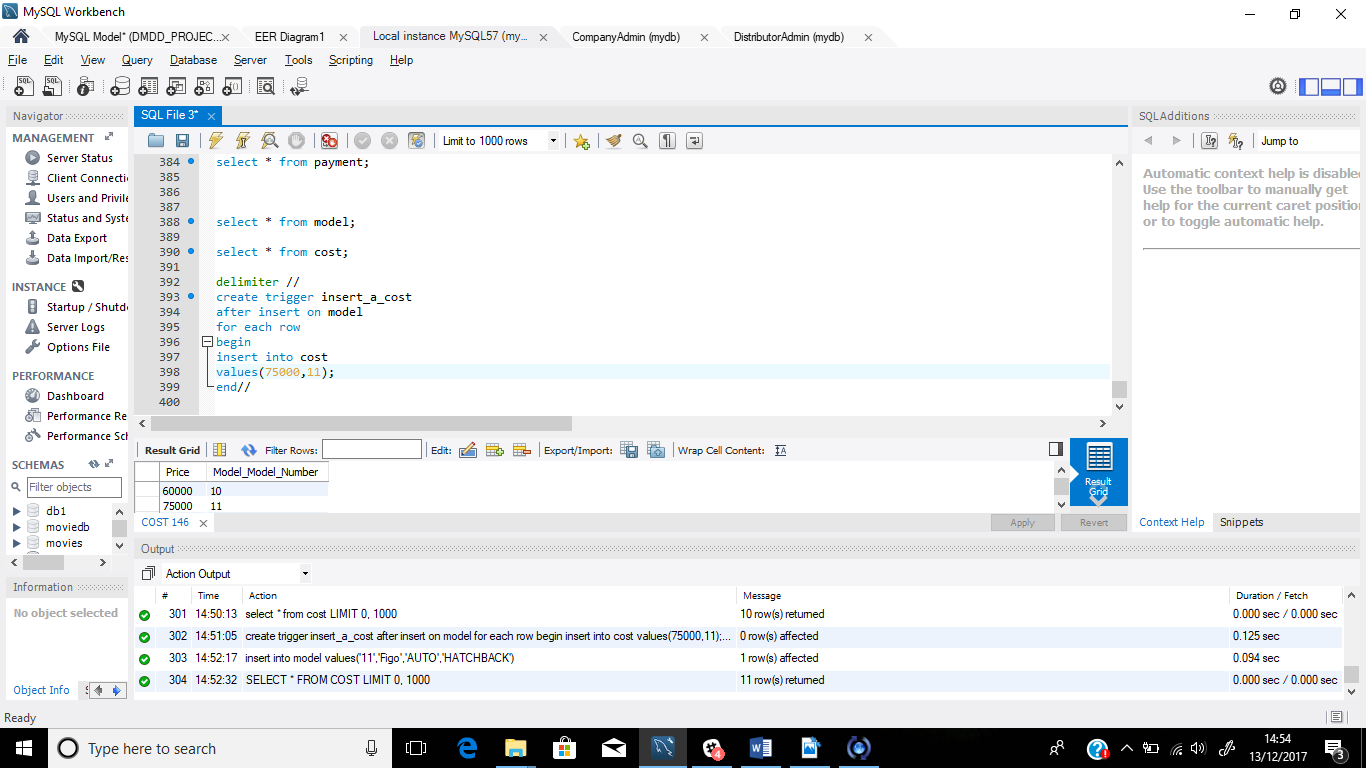
begin

insert into cost

values(75000,11);

end//





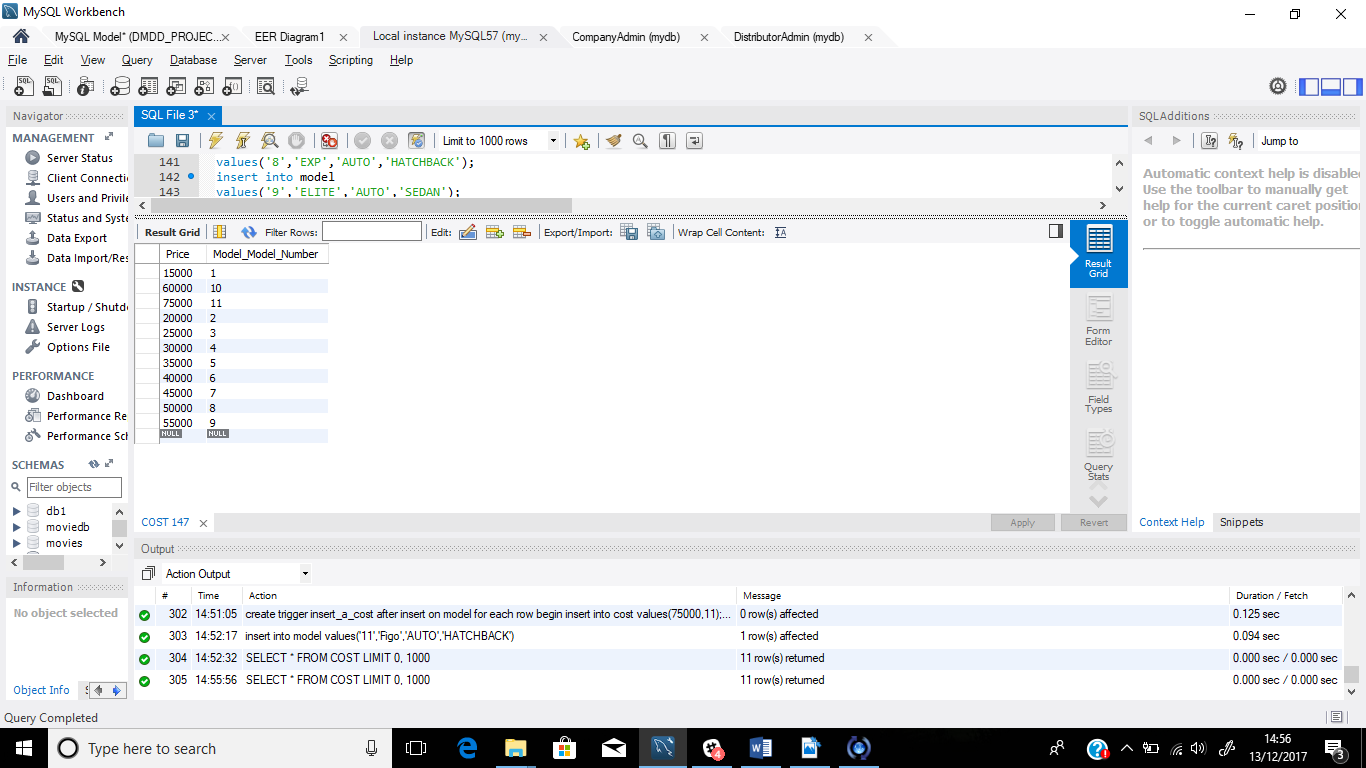
Now I created a new entry into the model table

--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;



1. **Stored Procedure**
2. 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

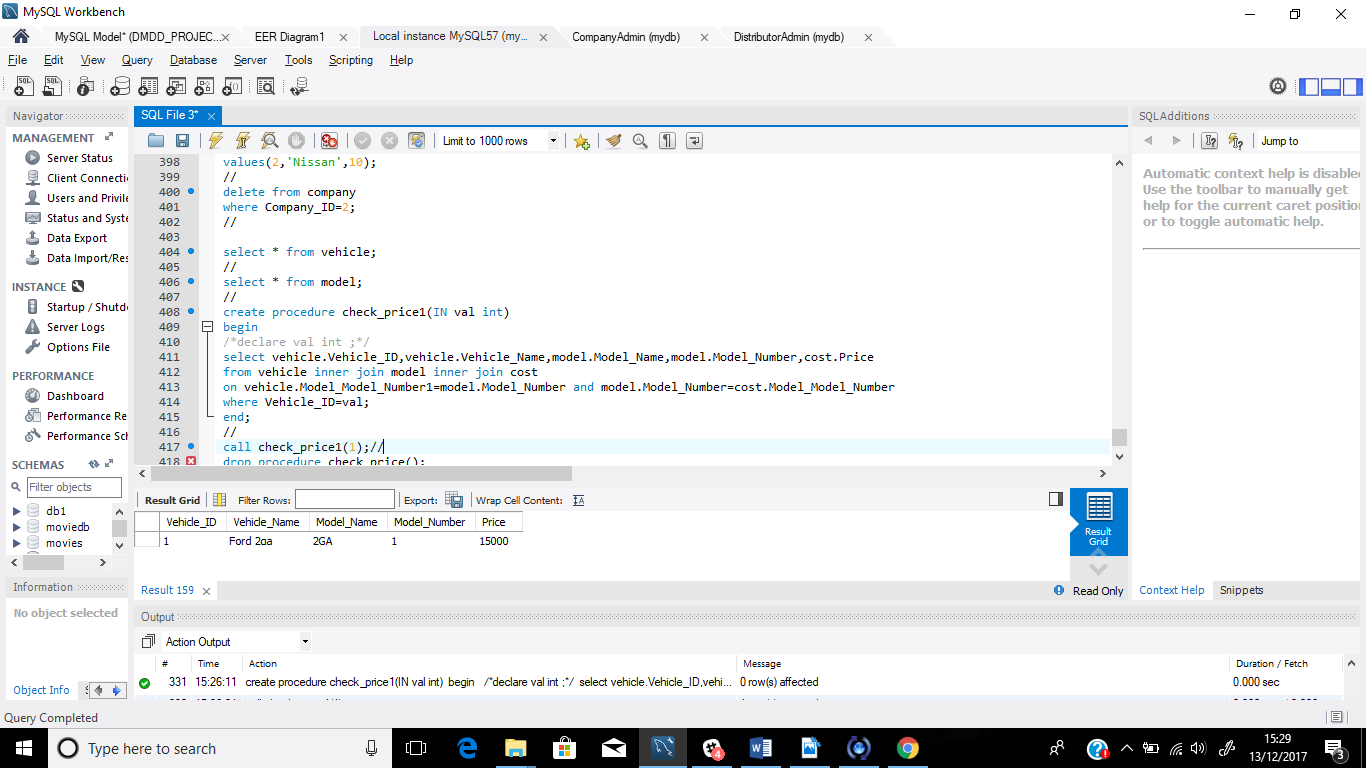
from vehicle inner join model inner join cost

on vehicle.Model\_Model\_Number1=model.Model\_Number and model.Model\_Number=cost.Model\_Model\_Number

where Vehicle\_ID=val;

end;

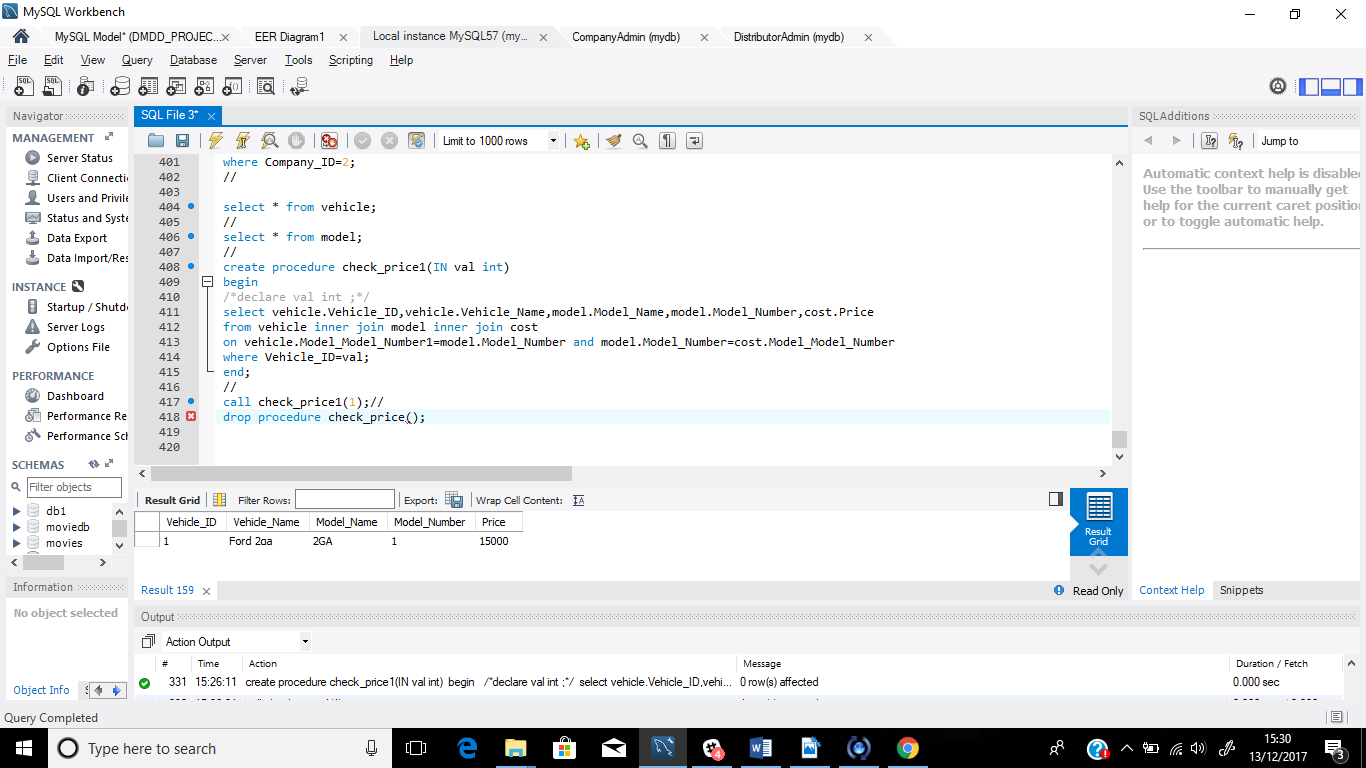
//



Now to call the procedure we use

- call check\_price1(1);//

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

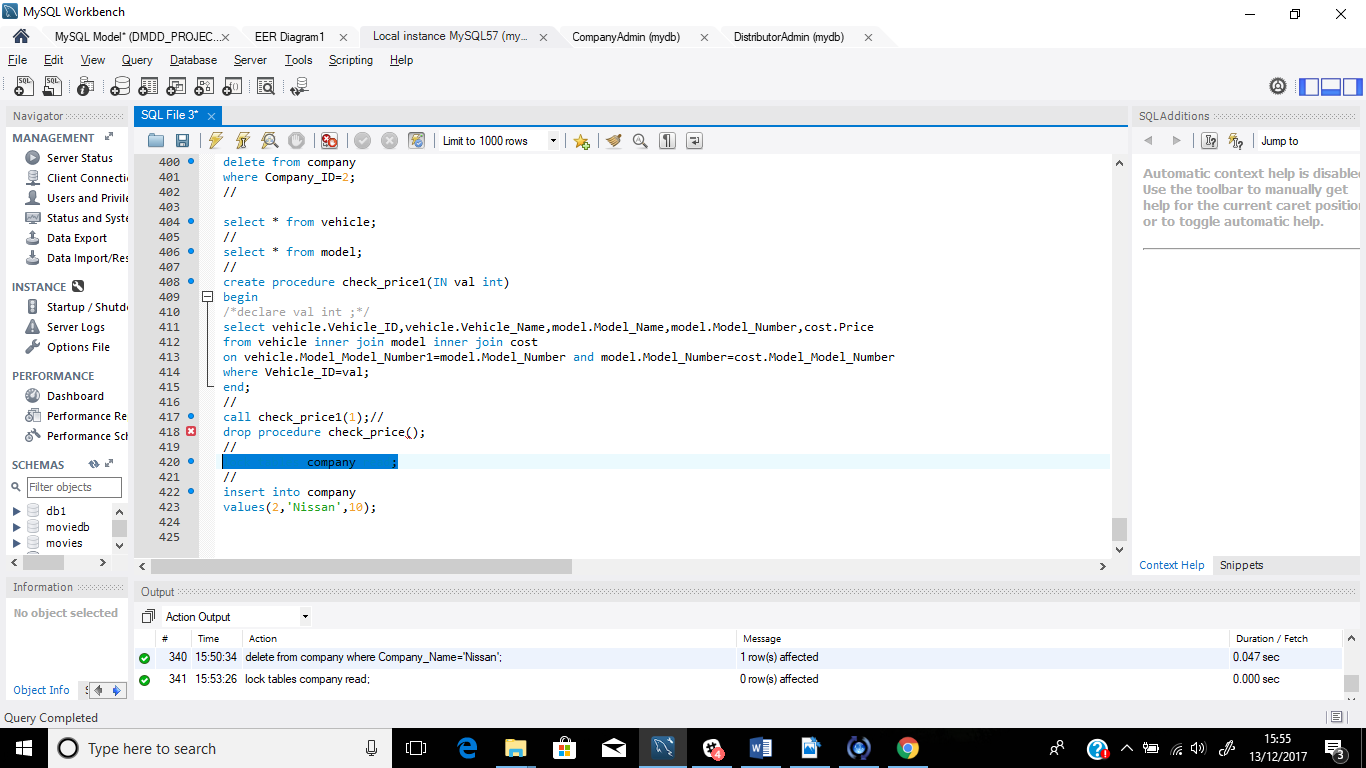


1. **Locking tables to avoid unwanted access on tables**
2. In this project, I have only considered to have 1 company i.e FORD

It shouldn’t be allowed for anyone to insert new data into the company table

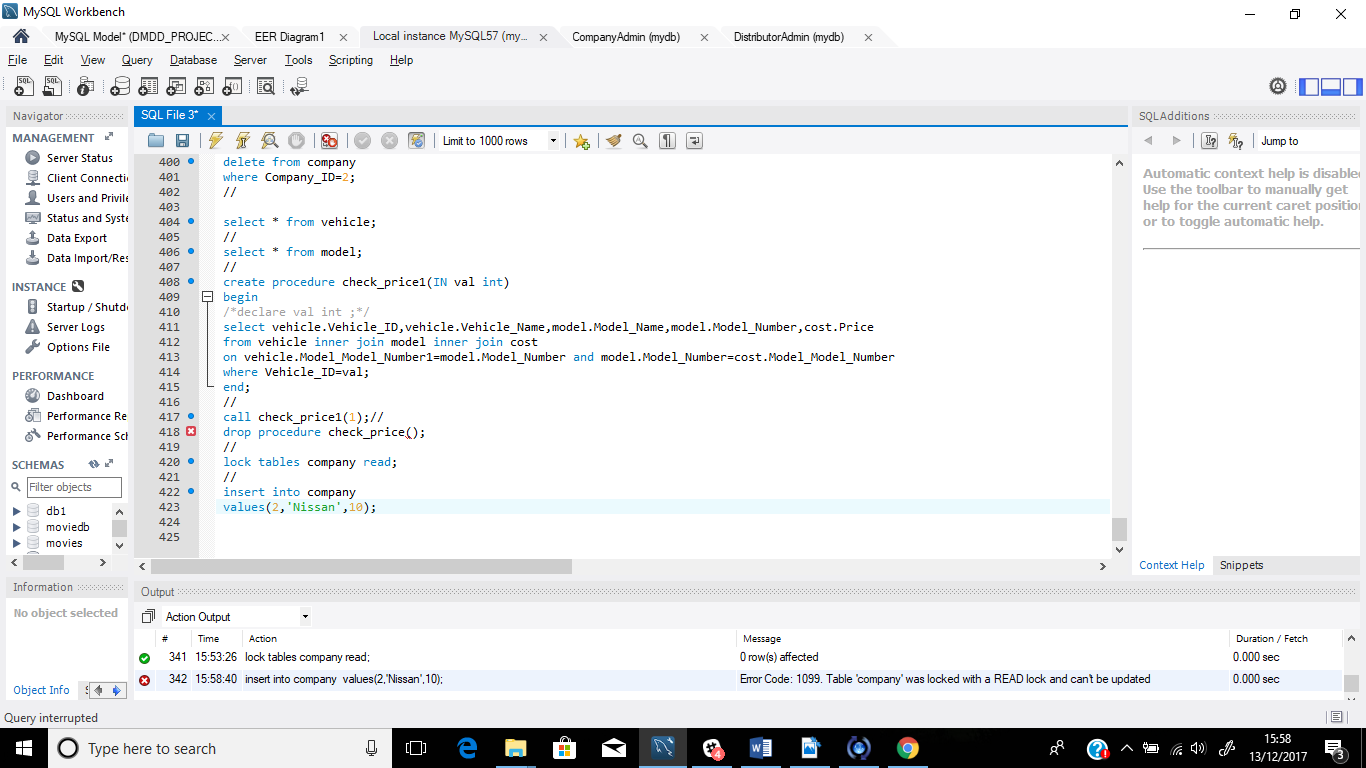
**CODE:**

lock tables company read;



NOW I TRIED TO INSERT DATA INTO THE COMPANY TABLE USING

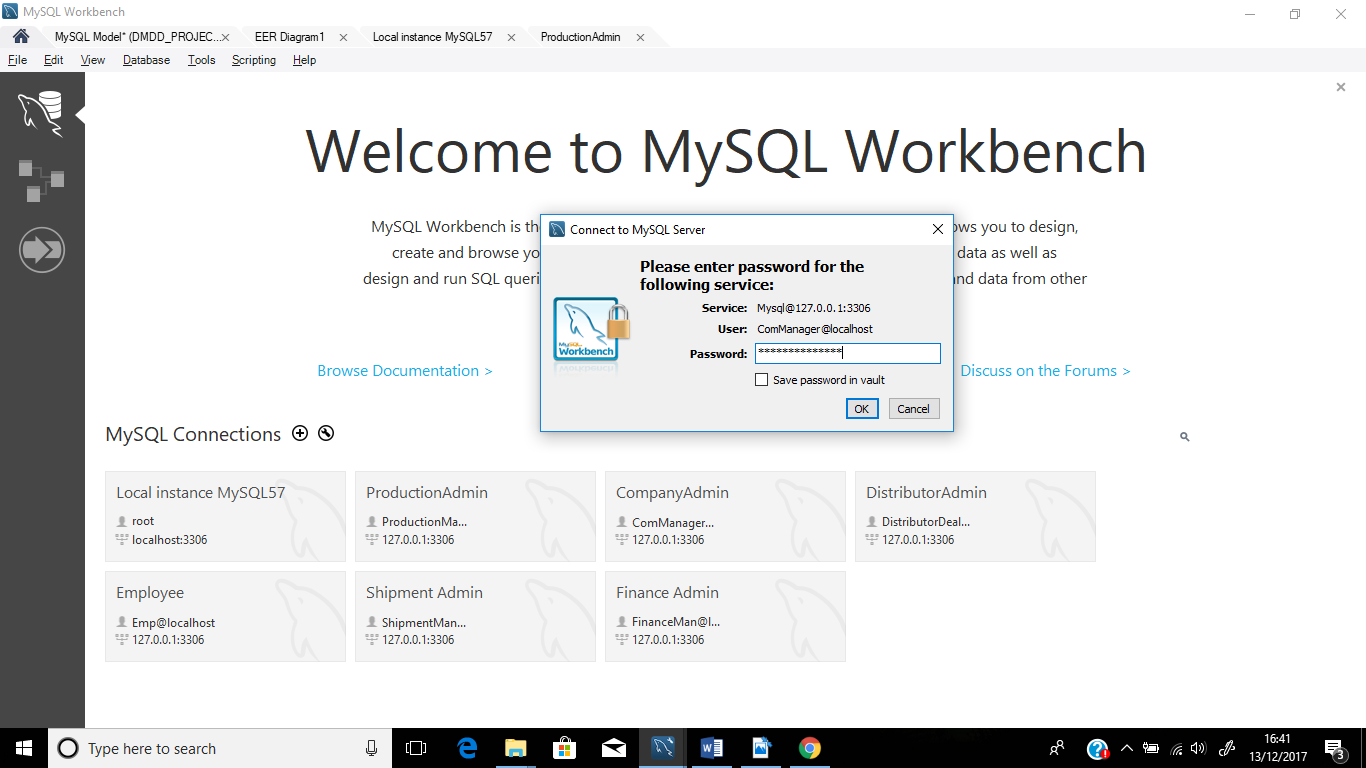
* INSERT INTO COMPANY VALUES ( 2, ‘NISSAN’ , 10 );
* AND THIS IS WHAT I GOT AS AN OUTPUT

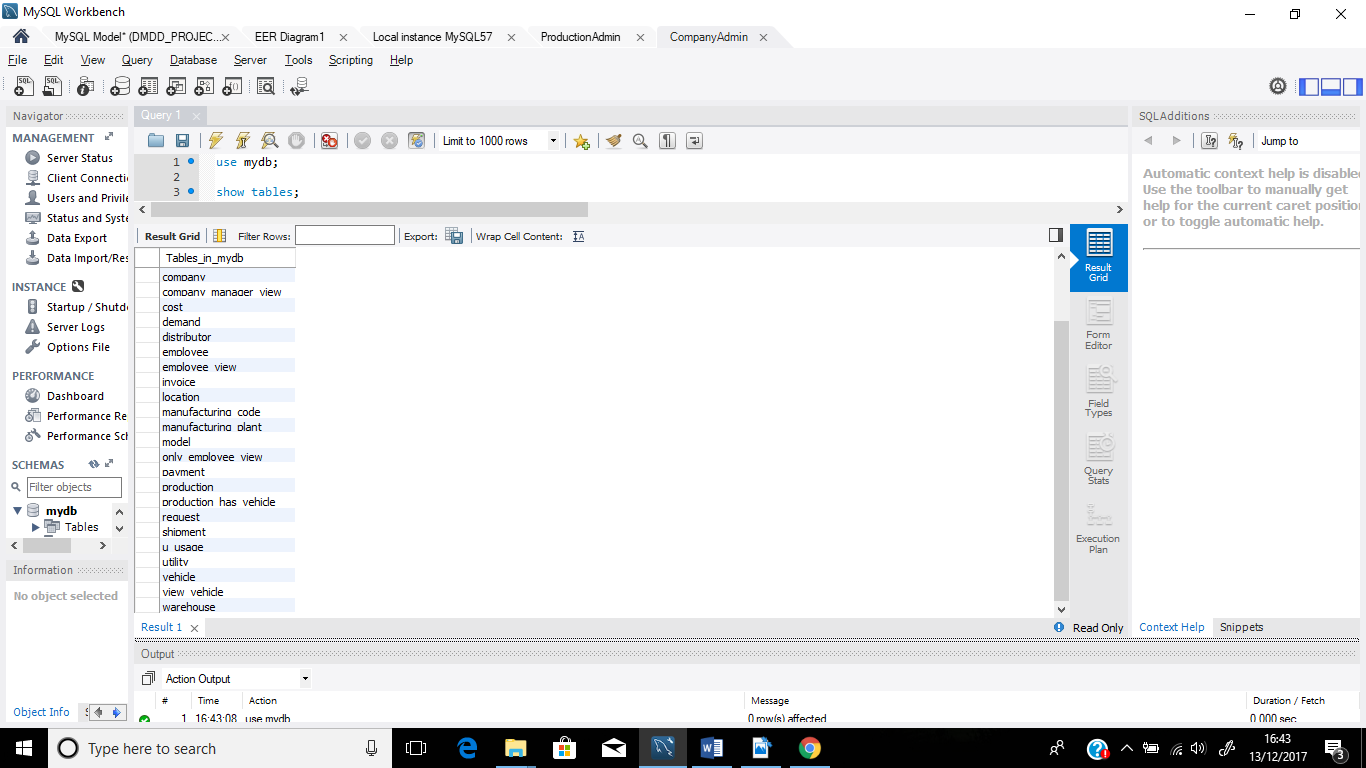


WHICH MEANS THE READ LOCK PLACED ON COMPANY TABLE DID NOT ALLOW ANYTHING TO BE WRITTEN TO IT.

1. **User privileges**
2. **Privilege Granted to Company Manager**

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





1. **Privilege Granted to Company Manager**

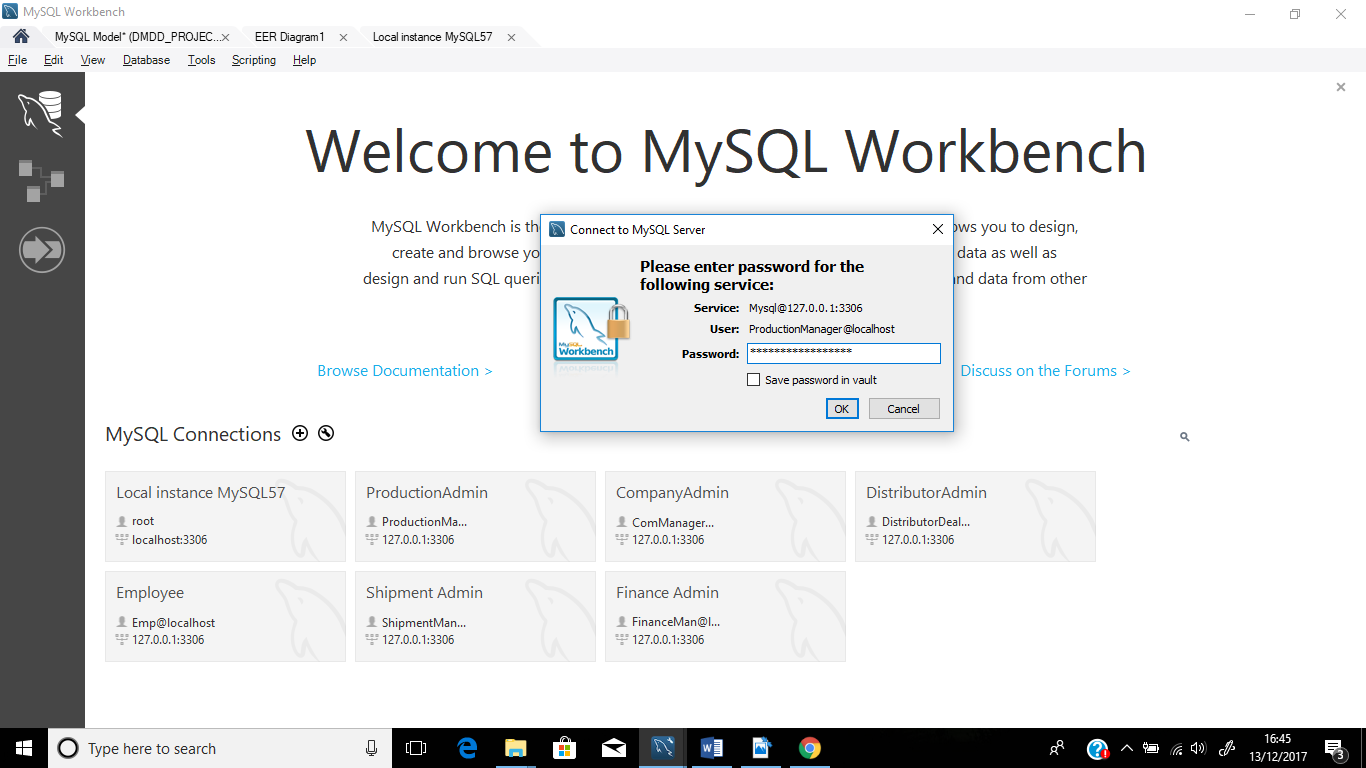
-grant all on mydb.Production to 'ProductionManager@localhost' @'localhost';

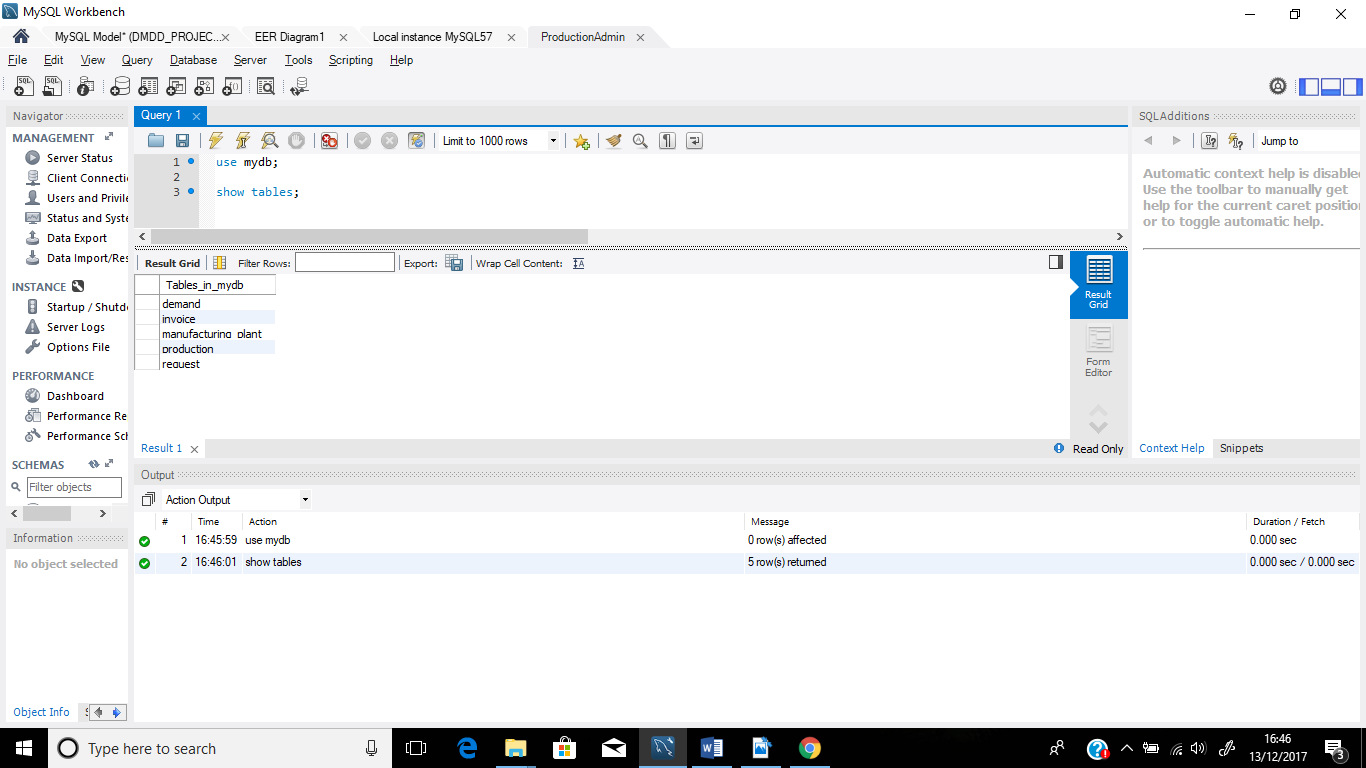
grant all on mydb.Manufacturing\_plant to 'ProductionManager@localhost' @'localhost';

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';





1. **Privilege Granted to Distributor**

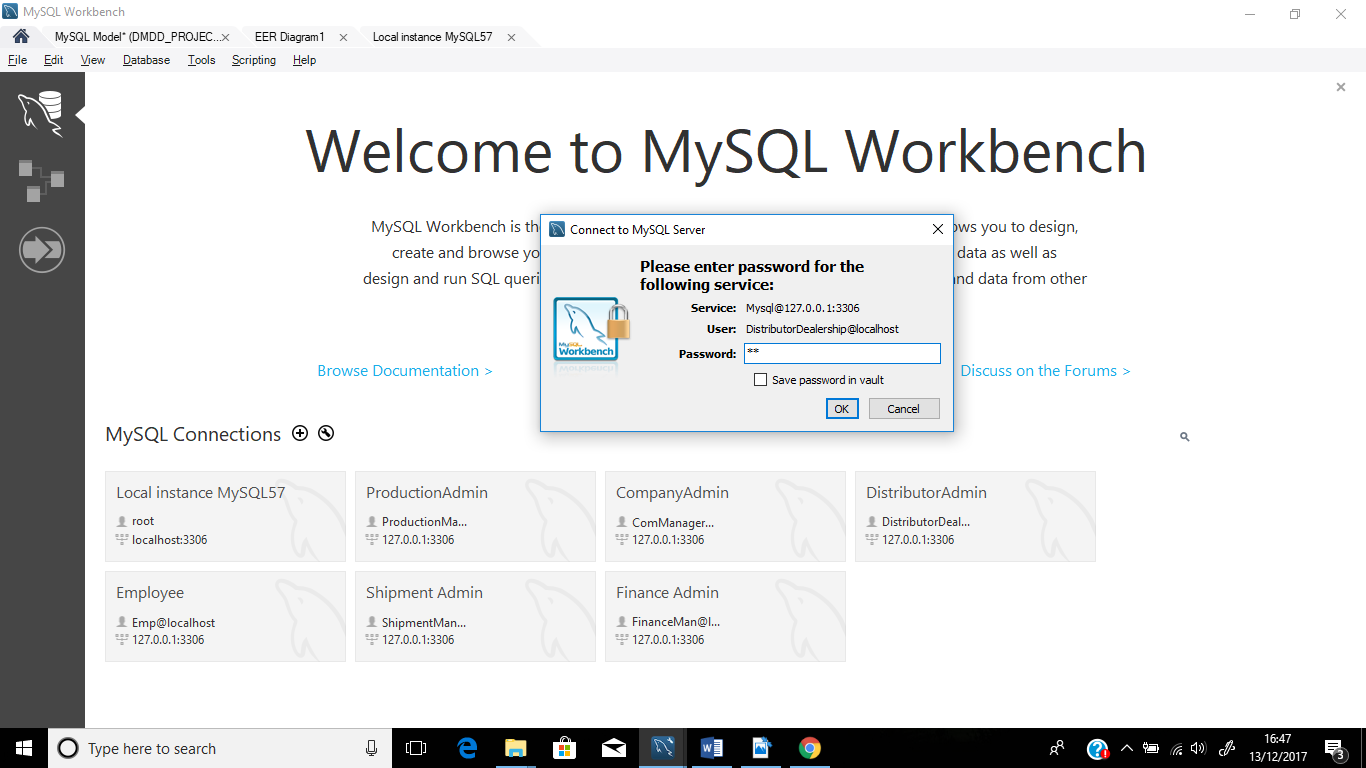
-

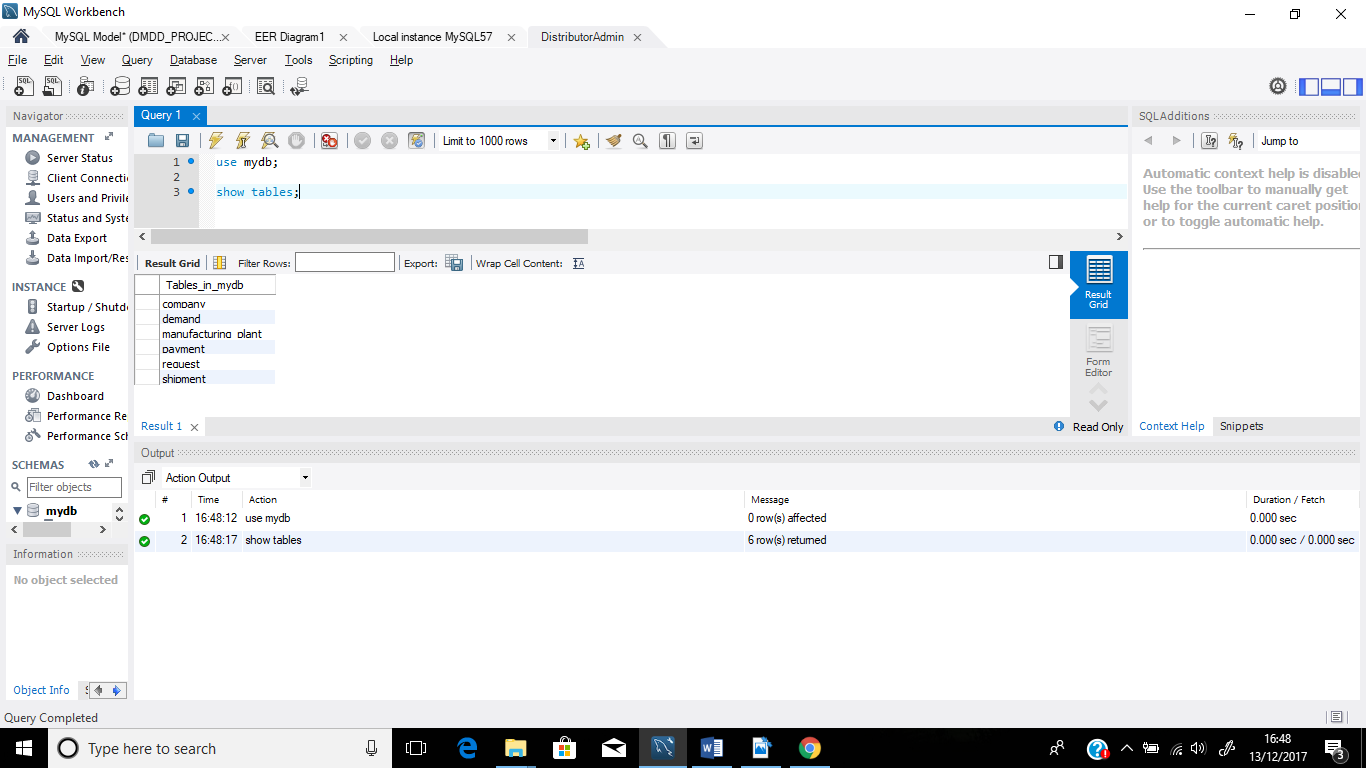
grant select on mydb.Manufacturing\_plant to 'DistributorDealership@localhost' @'localhost';

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';





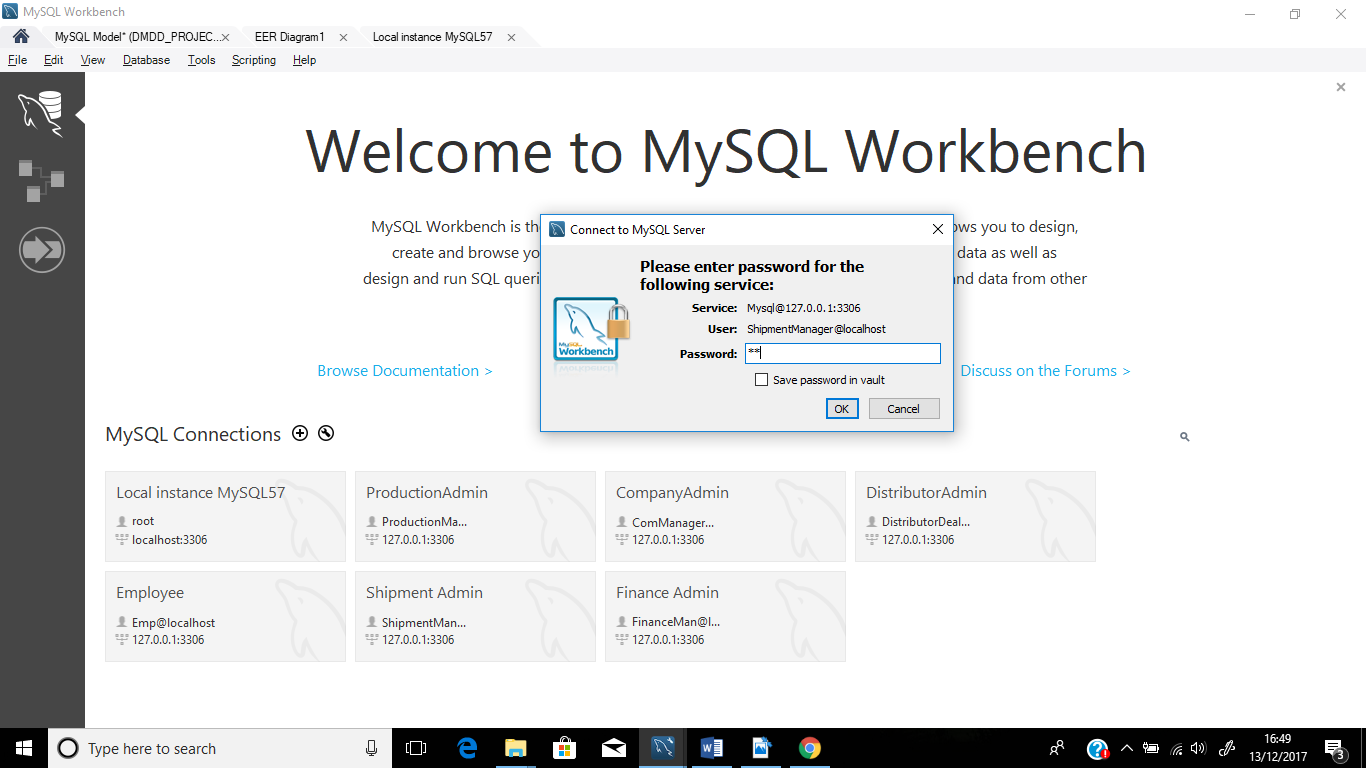
1. **Privilege Granted to Shipment Manager**

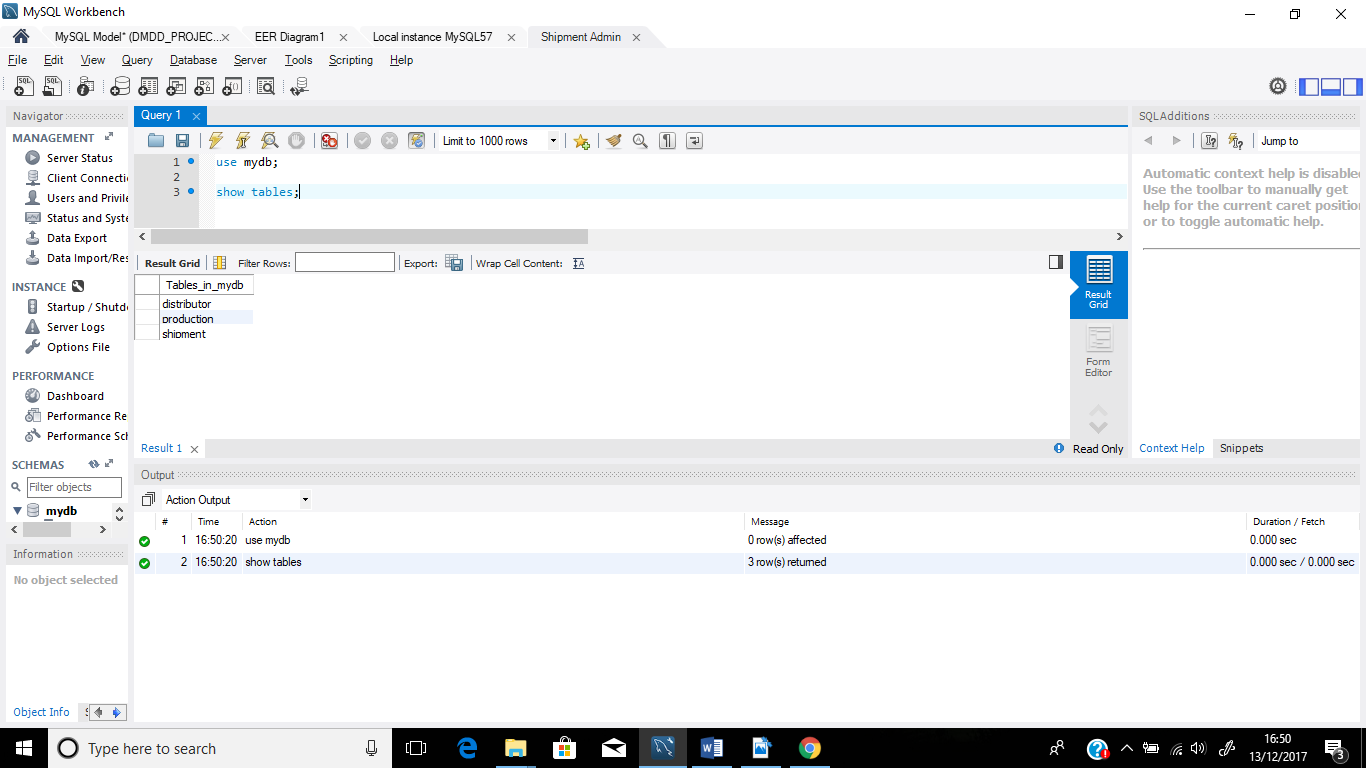
-

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

grant select on mydb.production to 'ShipmentManager@localhost' @'localhost';

grant select on mydb.distributor to 'ShipmentManager@localhost' @'localhost';





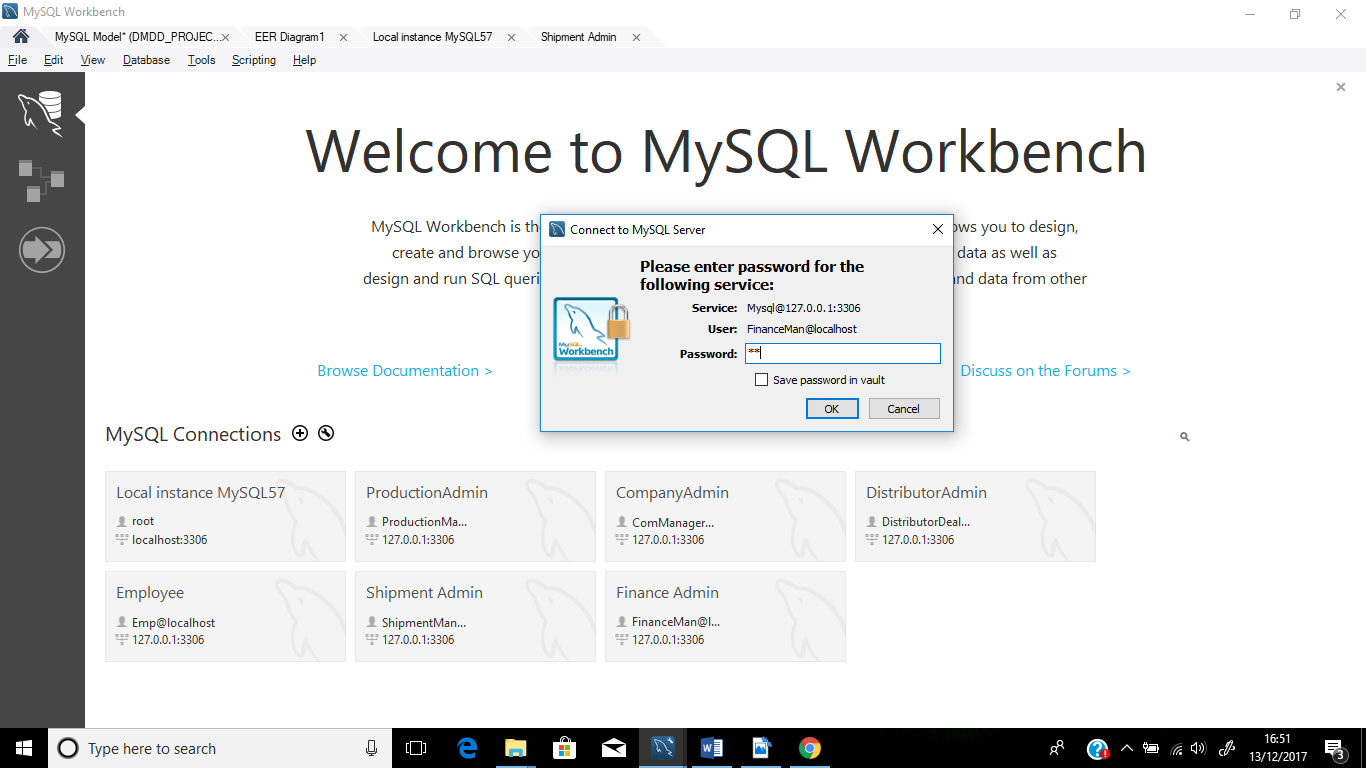
1. **Privilege Granted to Finance Manager**

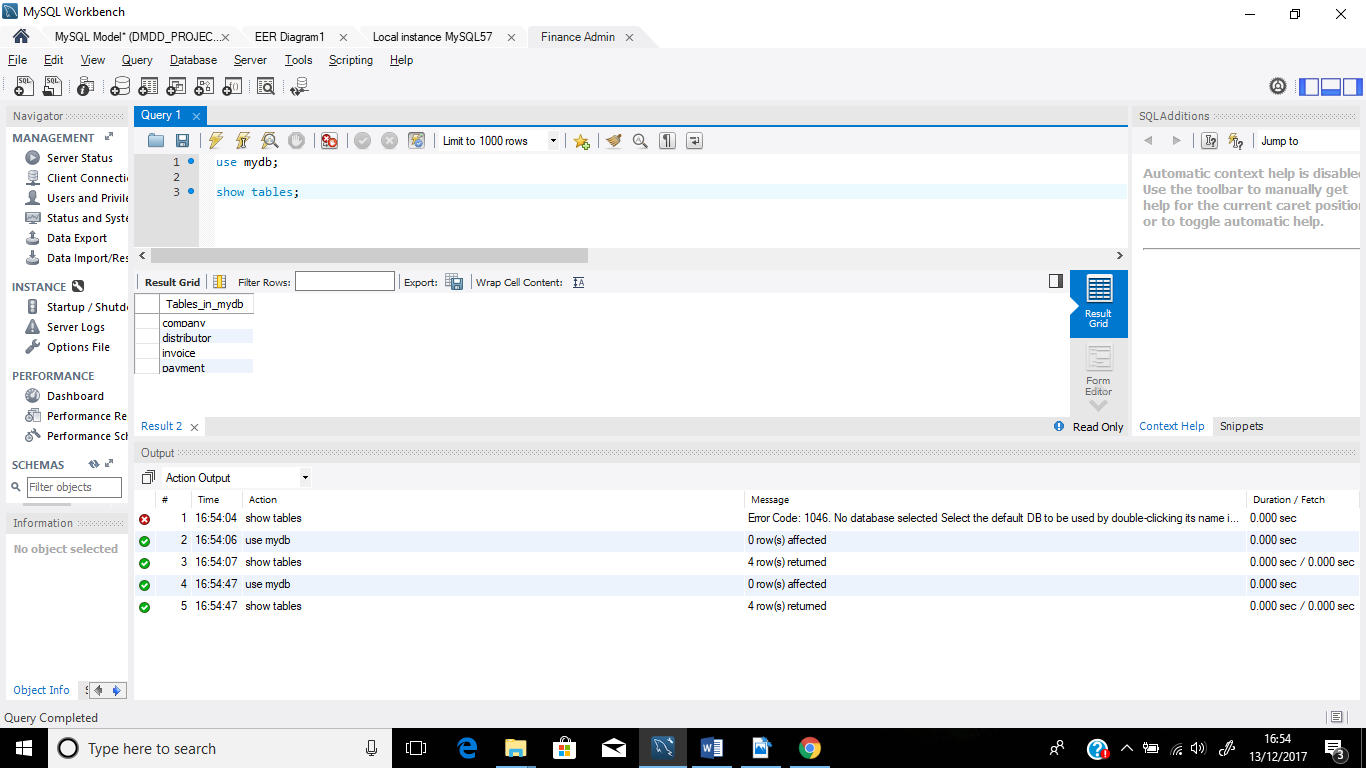
- grant select,insert,update,delete on mydb.payment to 'FinanceMan@localhost' @'localhost';

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';





1. **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';

