

Database Systems project ERD (Phase 2)

23CSIS03C

Project title: CRM Database - Customer Relationship Management system

Group Members:

ID	Name	Emails	Group
234742	Ahmed Ali Ramadan Abdallah	ahmed234742@bue.edu.eg	A7
235576	Abdelrahman Almakhzangy	Abdelrahman235576@bue.edu.eg	A3
235185	Nouran Mostafa ElKady	Nouran235185@bue.edu.eg	A3
236264	Hala Emad Gaber	Hala236264@bue.edu.eg	A4
230223	Nadeen Ragaee Makram	Nadin230223@bue.edu.eg	A9

Contribution table :

ID:	Name:	Detailed Contribution :
234742	Ahmed Ali Ramadan Abdallah	Support ticket, support representative, Current_customer (code & mapping)
235576	Abdelrahman Almakhzangy	Supplier, Product (code & mapping) Interact With (shared table)
235185	Nouran Mostafa ElKady	Marketer, Campaign, potential customer(code & mapping)
236264	Hala Emad Gaber	Sales representative, sales pipeline (code & mapping) Shared table (product)
230223	Nadeen Ragaae Makram	Employee, Department ,Employee_phone(code & mapping)

Description:

Through the multiple phases of adjustments and comprehensive edits and optimization of our database, we have reached the Final version of our customer relationship management database (CRM_DB). The database is designed to empower an SaaS Company that we have assumed to be Co.lab. For which we have assumed the products, regions and full company theme so that we can accurately curate a solid database. Below you will find more about our database including the tables, relationships, insertions, alters, queries, and scenarios.

Attributes and relationships:

Table name (Super Class)	Table names (Sub Class)	Relationship name	Relationship with & type
Employee	Support_rep	Work_on	Support_ticket , 1 to M
	Sales_rep	Working_on	Sales_pipeline (check Working_on Sales_pipeline with Sales_rep)
	Marketer	Manage	Campaign(check Manage Campaign with Marketer)
Customer	Potential customer	Is Target by	Campaign(check Target Campaign with Potential customer)
	Current customer	Submit	Support_ticket , 1 to M
Campaign		Target,Manage	Potential_customer, 1 to M Marketer, M to 1
Department		Works_in	Employee , 1 to M
Sales_pipeline		Working_on, Sell	Sales_rep of type 1 to M Product of type 1 to M
Support_ticket		Works_on,Submit	Support_rep (check Work_on Supp_rep with Support Ticket) Current_customer (check Submit Support_rep with Support Ticket)
Product		Interact_with	Current_customer , M to M
Supplier		Supply	Product , M to M

Scope:

The scope and purpose of our database is to help SaaS (Software as a Service) companies enhance their business and operations by providing better customer relationship management (CRM) and supporting their sales and marketing processes. The database stores data for the purpose of providing analytical insights about customers, support representatives, and support tickets, enabling the company to track and optimize customer interactions. The database also generates reports on how to target potential customers, follow up with current campaigns, follow up on company products on an ongoing basis, identify the best agents to handle new areas, and track company financial gains. This information helps facilitate company decision-making and maximize ROI. Furthermore, the database handles team information and management to facilitate teamwork.

Functionality:

Analyze customer behavior, preferences, and demographics to recommend products, segment customers, and improve products. Route customer support tickets to the best representatives and identify employees eligible for promotions. In addition we can also, Track customer lifetime value, product profitability, sales funnel performance, marketing campaign performance, product usage, top employees, and customer support ticket resolution times.

Below are some of the queries that can be used to reach the above functionalities:

Queries Description:

Ahmed 234742

Query 1:

Description : This query retrieves all data from the Support_rep table for employees located in the 'Cairo' region and their salary is above 2000.

Actual code:

create view A as

SELECT Support_rep.Employee_id, Employee.Fname, Employee.Lname, Support_rep.Skill,
Support_rep.Region

FROM Support_rep

INNER JOIN Employee ON Support_rep.Employee_id = Employee.Employee_id

WHERE Region = 'Cairo' and Salary > 2000;

select * from A;



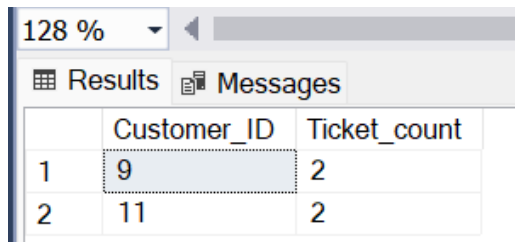
	Employee_id	Fname	Lname	Skill	Region
1	15	Yasmin	Abdelrahman	Customer Support	Cairo
2	16	Moha...	Gaber	Customer Support	Cairo

Query 2:

Description : This query counts the number of open tickets for each customer and returns only customers with more than two open tickets.

Actual code:

```
create view A2 AS
SELECT Customer_ID, COUNT(*) AS Ticket_count
FROM Support_ticket
WHERE Status = 'Open'
GROUP BY Customer_ID
HAVING COUNT(*) > 1;
select * From A2;
```



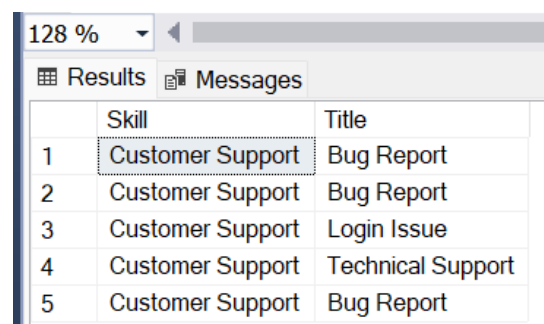
	Customer_ID	Ticket_count
1	9	2
2	11	2

Query 3:

Description : This query joins the 'Support_rep' and 'Support_ticket' tables based on the 'Emp_ID' and retrieves the skills of support representatives who are assigned to high-priority tickets.

Actual code:

```
create view A3 as
SELECT Support_rep.Skill, Support_ticket.Title
FROM Support_rep
INNER JOIN Support_ticket ON
Support_rep.Employee_id =
Support_ticket.Emp_ID
WHERE Support_ticket.Priority = 'High';
select *from A3;
```



	Skill	Title
1	Customer Support	Bug Report
2	Customer Support	Bug Report
3	Customer Support	Login Issue
4	Customer Support	Technical Support
5	Customer Support	Bug Report

Query 4:

Description : This query joins three tables to retrieve the usernames, job titles, and descriptions of support tickets containing the phrase "Bug Report" in their titles.

Actual code:

```
SELECT customer.customer_id,Current_Customer.Username, Customer.Job_title,  
Support_ticket.Description,Support_ticket.Status  
FROM Current_Customer  
INNER JOIN Customer ON Current_Customer.Customer_ID = Customer.Customer_ID  
INNER JOIN Support_ticket ON Customer.Customer_ID = Support_ticket.Customer_ID  
WHERE Support_ticket.Title LIKE '%Bug Report%';
```

	customer_id	Username	Job_title	Description	Status
1	9	Mona_ElMahdy	Engineer	Bug in co-sheets calculations	Open
2	12	Farid_Salem	Software Developer	Error in co-note features	Closed
3	12	Farid_Salem	Software Developer	Bug in co-calendar notifications	Closed

Query 5:

Description : This query uses a nested query to find the Emp_ID of the support representative who closed the most recent ticket, then retrieves the first and last name of that representative it saves those and full name., their ID, and the number of tickets they have recently closed.

Actual code:

```
SELECT Employee.employee_id, Employee.Fname + ' ' + Employee.Lname AS 'Employee Name', COUNT(Support_ticket.Emp_ID) AS 'Number of Closed Tickets'
FROM Employee
INNER JOIN Support_rep ON Employee.Employee_id = Support_rep.Employee_id
INNER JOIN Support_ticket ON Support_ticket.Emp_ID = Support_rep.Employee_id
WHERE Support_ticket.Status = 'Closed'
GROUP BY Employee.employee_id, Employee.Fname + ' ' + Employee.Lname,
Employee.Lname
ORDER BY Employee.Lname ASC;
```

	employee_id	Employee Name	Number of Closed Tickets
1	15	Yasmin Abdelrahman	2
2	16	Mohammed Gaber	2
3	17	Nada Ibrahim	1

Query 6:

Description :

Actual code:

```
SELECT Customer.Fname, Customer.Lname, Support_ticket.Title,  
Support_ticket.Priority,Support_ticket.Due_date  
FROM Customer  
INNER JOIN Support_ticket ON Customer.Customer_ID = Support_ticket.Customer_ID  
WHERE Support_ticket.Status = 'Open' AND Support_ticket.Emp_ID IN (  
    SELECT Employee_id  
    FROM Support_rep  
    WHERE Skill LIKE '%Customer Support%'  
);
```

	Fname	Lname	Title	Priority	Due_date
1	Mona	ElMahdy	Bug Report	High	2023-01-10
2	Amina	Youssef	Billing Inquiry	Low	2023-01-20
3	Yasmine	Rizk	Login Issue	High	2023-02-10
4	Mona	ElMahdy	Technical Support	High	2023-02-20
5	Amina	Youssef	Product Inquiry	Low	2023-03-01

Query 7:

Description : this query finds customers who have the most open tickets assigned to support representatives with the highest average resolution time

Actual code:

```
SELECT c.Fname + ' ' + c.Lname AS 'Full Name', COUNT(st.Ticket_id) AS Open_ticket_count,  
c.Customer_ID  
from Customer c  
INNER JOIN Support_ticket st ON c.Customer_ID = st.Customer_ID  
WHERE st.Status = 'Open'  
GROUP BY c.Fname, c.Lname, c.Customer_ID  
ORDER BY Open_ticket_count DESC;
```

	Full Name	Open_ticket_count	Customer_ID
1	Mona ElMahdy	2	9
2	Amina Youssef	2	11
3	Yasmine Rizk	1	15

Query 8:

Description : this query is used to get the employee_id and full names ('Employee Name') of employees -who are support representatives and have closed support tickets, sorted by their last names in ascending order. it shows also the priority of tickets they have worked on and the tickets ID as this query is going to be used to promote the person with works the most on high priority tickets to a supervisor.

Actual code:

```
SELECT Employee.employee_id ,Employee.Fname+ ' ' + Employee.Lname as 'Employee Name'
, Support_ticket.priority, Support_ticket.Ticket_id
FROM Employee
INNER JOIN Support_rep ON Employee.Employee_id = Support_rep.Employee_id
inner join Support_ticket on employee.employee_id = Support_ticket.Emp_ID
WHERE Support_rep.Employee_id IN (
    SELECT Support_ticket.Emp_ID
    FROM Support_ticket
    WHERE Status = 'closed'
)
ORDER BY Employee.Lname ASC;
```

	employee_id	Employee Name	priority	Ticket_id
1	15	Yasmin Abdelrahman	High	1
2	15	Yasmin Abdelrahman	High	4
3	15	Yasmin Abdelrahman	High	7
4	15	Yasmin Abdelrahman	High	9
5	15	Yasmin Abdelrahman	High	12
6	16	Mohammed Gaber	Medium	2
7	16	Mohammed Gaber	Medium	5
8	16	Mohammed Gaber	Medium	8
9	16	Mohammed Gaber	Medium	10
10	17	Nada Ibrahim	Low	3
11	17	Nada Ibrahim	Low	6
12	17	Nada Ibrahim	Low	11

Hala 236264

Query 1:

Description: This query retrieves the distinct contact name of contacts that in a stage less than 3 from the same table (sales pipeline) and present stage under the name “contact name is” to refer to which stage they are currently in

Actual code :

```
SELECT DISTINCT Contact_name, stage AS  
'The contact name is'  
FROM Sales_Pipeline  
where stage <'3';
```

89 %	Results	Messages
	Contact_name	The contact name is
1	Ahmed Hassan	1
2	Mohamed hussein	2

Query 2:

Description : This query retrieves the Fname, Lname and the Employee ID , from 2 tables sales pipeline and Employee, the output will display to the user the number of people (using their first and last name)that each sales pipeline are working with .

Actual code:

```
SELECT COUNT(P.sales_pipeline_ID), Fname,Lname ,E.Employee_ID  
FROM Sales_Pipeline P JOIN Sales_Rep R on P.sales_pipeline_ID= R.Employee_id JOIN  
Employee E on E.Employee_id = R.Employee_id  
group by E.Employee_ID,Fname,Lname;
```

9 %

Results

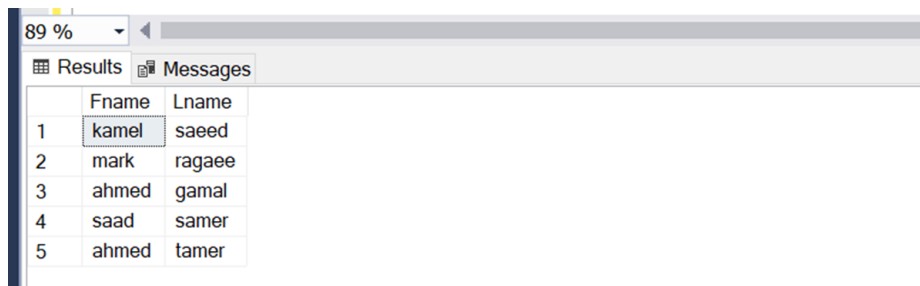
Messages

	(No column name)	Fname	Lname	Employee_ID
1	1	saad	samer	12467
2	1	amer	omar	14789
3	1	ahmed	tamer	15798
4	1	kamel	saeed	19203
5	1	mark	ragae	19872
6	1	tamer	aly	22567
7	1	ahmed	gamal	23793

Query 3:

Description : This query retrieves the Fname , Lname from the employee and sales rep tables ,where the territory (is between A and S) And in order of territory

Actual code : SELECT Fname,Lname
FROM Employee E, Sales_Rep S
Where E.Employee_id = S.Employee_id AND Territory BETWEEN 'A' AND 'S'
Order by Territory;



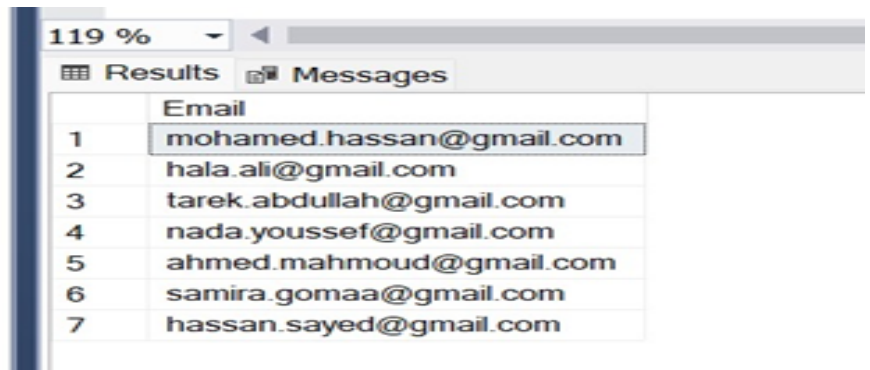
A screenshot of a SQL query results window. The window has a zoom level of 89% and two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a table with two columns: 'Fname' and 'Lname'. There are five rows of data, numbered 1 to 5 in the first column. The data is as follows:

	Fname	Lname
1	kamel	saeed
2	mark	ragae
3	ahmed	gamal
4	saad	samer
5	ahmed	tamer

Query 4:

Description : This query Retrieves the mails of the customers that sales pipeline works with , That their current stage =3

Actual code : SELECT Email
FROM Customer
WHERE Customer_ID IN
(SELECT Customer_ID
FROM Potential_customer
WHERE Campaign_ID IN (
SELECT ID
FROM Campaign
WHERE Employee_ID_ IN
(SELECT Employee_ID_
FROM Sales_Pipeline
WHERE Stage = 3))
);



A screenshot of a SQL query results window. The window has a zoom level of 119% and two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a table with one column: 'Email'. There are seven rows of data, numbered 1 to 7 in the first column. The data is as follows:

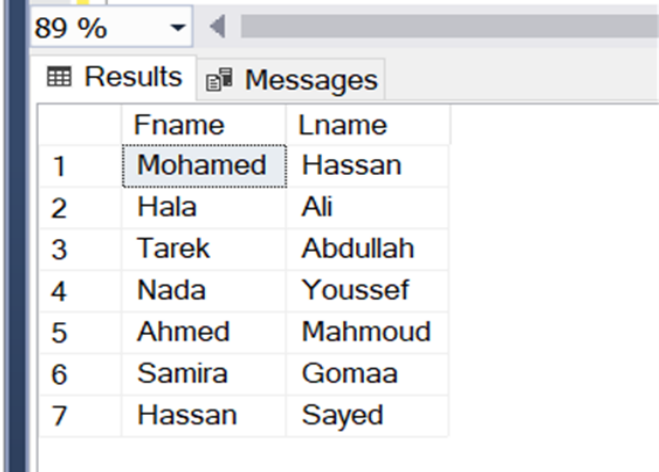
	Email
1	mohamed.hassan@gmail.com
2	hala.ali@gmail.com
3	tarek.abdullah@gmail.com
4	nada.youssef@gmail.com
5	ahmed.mahmoud@gmail.com
6	samira.gomaa@gmail.com
7	hassan.sayed@gmail.com

Query 5:

Description : This query displays the first and the last name of the customers that have sales pipeline which have an opportunity amount greater than 100

Actual code :

```
SELECT Fname, Lname
FROM Customer
WHERE Customer_ID IN
(SELECT Customer_ID
FROM Potential_customer
WHERE Campaign_ID IN
(SELECT ID
FROM Campaign
WHERE Employee_ID_ IN
(SELECT Employee_ID_
FROM Sales_Pipeline
WHERE Opportunity_Amount > 100))
);
```



The screenshot shows a database application window with a query results grid. The grid has two columns, 'Fname' and 'Lname', and seven rows of data. The first row is highlighted. Above the grid, there are tabs for 'Results' and 'Messages', and a progress indicator showing '89 %'.

	Fname	Lname
1	Mohamed	Hassan
2	Hala	Ali
3	Tarek	Abdullah
4	Nada	Youssef
5	Ahmed	Mahmoud
6	Samira	Gomaa
7	Hassan	Sayed

Abdelrahman 235576

Query 1:

Description : This query counts the number of products and retrieves the highest , lowest , average and total prices of the products. Also it retrieves the maximum and minimum expiration date.

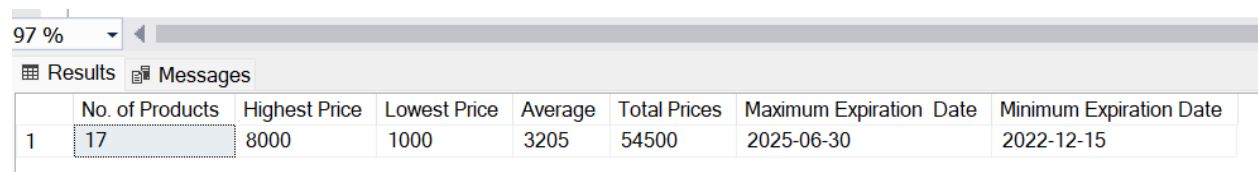
Actual code:

Create View v AS

Select

```
COUNT (*) AS "No. of Products",  
MAX(Price) AS "Highest Price" ,  
MIN(Price) AS "Lowest Price" ,  
AVG (Price) AS "Average" ,  
SUM (Price) AS "Total Prices",  
MAX (Expire_date) AS "Maximum Expiration Date" ,  
MIN (Expire_date) AS "Minimum Expiration Date"  
From Product;
```

Select * From v;



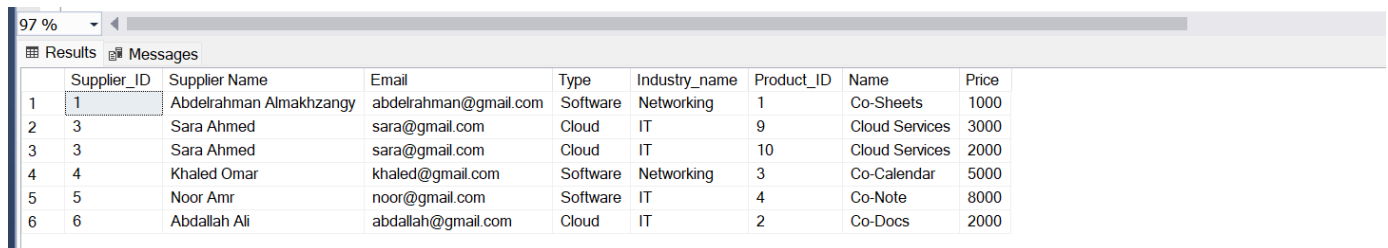
	No. of Products	Highest Price	Lowest Price	Average	Total Prices	Maximum Expiration Date	Minimum Expiration Date
1	17	8000	1000	3205	54500	2025-06-30	2022-12-15

Query 2:

Description : This query retrieves the supplier id , email , name , type of product supplied , industry name , product id , name of the product , and price using the join keyword between product and supply table and supplier with supply table and then the data is sorted by the supplier ID.

Actual code:

```
Select s.Supplier_ID , s.Fname + ' ' + s.Lname AS "Supplier Name", s.Email, s.Type,  
s.Industry_name , p.Product_ID, p.Name , p.Price  
from Supplier s  
inner join supply u on s.Supplier_ID = u.Supplier_ID  
inner join Product p on u.Product_ID = p.Product_ID  
Order by s.Supplier_ID ;
```



	Supplier_ID	Supplier Name	Email	Type	Industry_name	Product_ID	Name	Price
1	1	Abdelrahman Almakhzangy	abdelrahman@gmail.com	Software	Networking	1	Co-Sheets	1000
2	3	Sara Ahmed	sara@gmail.com	Cloud	IT	9	Cloud Services	3000
3	3	Sara Ahmed	sara@gmail.com	Cloud	IT	10	Cloud Services	2000
4	4	Khaled Omar	khaled@gmail.com	Software	Networking	3	Co-Calendar	5000
5	5	Noor Amr	noor@gmail.com	Software	IT	4	Co-Note	8000
6	6	Abdallah Ali	abdallah@gmail.com	Cloud	IT	2	Co-Docs	2000

Query 3:

Description : This query retrieves the supplier id , name , email , type of product supplied , industry name , product id , name and price, It uses right outer join to get the data from the supplier and supply table and the matching data from product table also there is some nulls because there is no match for products and suppliers and the data is sorted by the supplier id in descending order.

Actual code:

```
Select s.Supplier_ID , s.Fname + ' ' + s.Lname AS "Supplier Name", s.Email , s.Type,
s.Industry_name , p.Product_ID, p.Name , p.Price
from Supplier s
right outer join supply u on s.Supplier_ID = u.Supplier_ID
right outer join Product p on u.Product_ID = p.Product_ID
Order BY s.supplier_ID DESC;
```

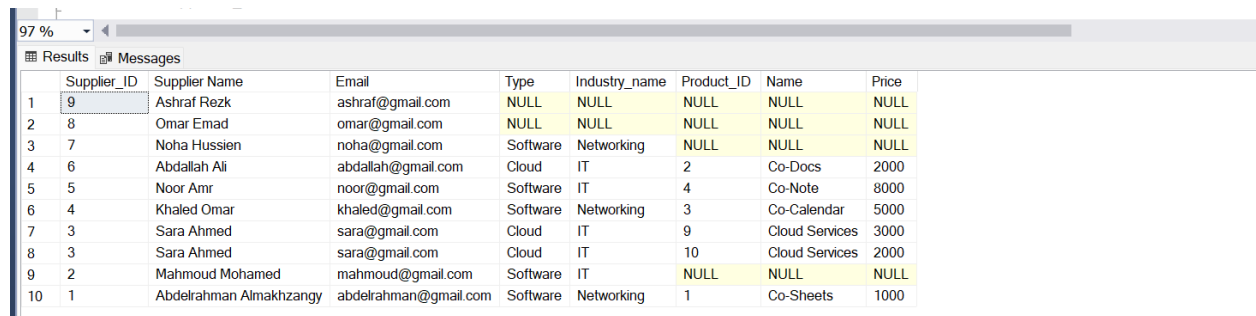
88 %								
Results Messages								
	Supplier_ID	Supplier Name	Email	Type	Industry_name	Product_ID	Name	Price
1	6	Abdallah Ali	abdallah@gmail.com	Cloud	IT	2	Co-Docs	2000
2	5	Noor Amr	noor@gmail.com	Software	IT	4	Co-Note	8000
3	4	Khaled Omar	khaled@gmail.com	Software	Networking	3	Co-Calendar	5000
4	3	Sara Ahmed	sara@gmail.com	Cloud	IT	9	Cloud Services	3000
5	3	Sara Ahmed	sara@gmail.com	Cloud	IT	10	Cloud Services	2000
6	1	Abdelrahman Almakhzangy	abdelrahman@gmail.com	Software	Networking	1	Co-Sheets	1000
7	NULL	NULL	NULL	NULL	NULL	5	Co-Forms	1500
8	NULL	NULL	NULL	NULL	NULL	6	Co-Present	2500
9	NULL	NULL	NULL	NULL	NULL	7	Chat Service	5000
10	NULL	NULL	NULL	NULL	NULL	8	Video Call Service	7000
11	NULL	NULL	NULL	NULL	NULL	11	Co-Calendar	5000
12	NULL	NULL	NULL	NULL	NULL	12	Co-Maps	4000
13	NULL	NULL	NULL	NULL	NULL	13	Co-Weather	1000
14	NULL	NULL	NULL	NULL	NULL	12467	Co-Calendar	1000
15	NULL	NULL	NULL	NULL	NULL	14789	Co-Note	3000
16	NULL	NULL	NULL	NULL	NULL	19203	Co-Docs	2000
17	NULL	NULL	NULL	NULL	NULL	23793	Co-Sheets	1500

Query 4:

Description : This query retrieves the supplier id , name , email , type of the product supplied , industry name , product id , name and price, It uses left outer join to get all data from the supplier table and the matching data from supply and product also there are some nulls because there is no match between with products and supplier and the data is sorted by supplier ID in descending order.

Actual code:

```
Select s.Supplier_ID , s.Fname + ' ' + s.Lname AS "Supplier Name", s.Email , s.Type,
s.Industry_name , p.Product_ID, p.Name , p.Price
from Supplier s
left outer join supply u on s.Supplier_ID = u.Supplier_ID
left outer join Product p on u.Product_ID = p.Product_ID
Order BY s.supplier_ID DESC;
```



	Supplier_ID	Supplier Name	Email	Type	Industry_name	Product_ID	Name	Price
1	9	Ashraf Rezk	ashraf@gmail.com	NULL	NULL	NULL	NULL	NULL
2	8	Omar Emad	omar@gmail.com	NULL	NULL	NULL	NULL	NULL
3	7	Noha Hussien	noha@gmail.com	Software	Networking	NULL	NULL	NULL
4	6	Abdallah Ali	abdallah@gmail.com	Cloud	IT	2	Co-Docs	2000
5	5	Noor Amr	noor@gmail.com	Software	IT	4	Co-Note	8000
6	4	Khaled Omar	khaled@gmail.com	Software	Networking	3	Co-Calendar	5000
7	3	Sara Ahmed	sara@gmail.com	Cloud	IT	9	Cloud Services	3000
8	3	Sara Ahmed	sara@gmail.com	Cloud	IT	10	Cloud Services	2000
9	2	Mahmoud Mohamed	mahmoud@gmail.com	Software	IT	NULL	NULL	NULL
10	1	Abdelrahman Almahzangy	abdelrahman@gmail.com	Software	Networking	1	Co-Sheets	1000

Query 5:

Description : This query retrieves the supplier id , name , email , type of product supplied , industry name , product id , name and price. It uses full outer join to get all data from supplier , supply and product table. There are some nulls because there is no match for some suppliers and products and the result is sorted by supplier ID by descending order.

Actual code:

```
Select s.Supplier_ID , s.Fname + ' ' + s.Lname AS "Supplier Name", s.Email , s.Type,
s.Industry_name , p.Product_ID, p.Name , p.Price
from Supplier s
full outer join supply u on s.Supplier_ID = u.Supplier_ID
full outer join Product p on u.Product_ID = p.Product_ID
Order By s.Supplier_ID DESC;
```

73 %								
Results Messages								
	Supplier_ID	Supplier Name	Email	Type	Industry_name	Product_ID	Name	Price
1	9	Ashraf Rezk	ashraf@gmail.com	NULL	NULL	NULL	NULL	NULL
2	8	Omar Emad	omar@gmail.com	NULL	NULL	NULL	NULL	NULL
3	7	Noha Hussien	noha@gmail.com	Software	Networking	NULL	NULL	NULL
4	6	Abdallah Ali	abdallah@gmail.com	Cloud	IT	2	Co-Docs	2000
5	5	Noor Amr	noor@gmail.com	Software	IT	4	Co-Note	8000
6	4	Khaled Omar	khaled@gmail.com	Software	Networking	3	Co-Calendar	5000
7	3	Sara Ahmed	sara@gmail.com	Cloud	IT	9	Cloud Services	3000
8	3	Sara Ahmed	sara@gmail.com	Cloud	IT	10	Cloud Services	2000
9	2	Mahmoud Mohamed	mahmoud@gmail.com	Software	IT	NULL	NULL	NULL
10	1	Abdelrahman Almakhzangy	abdelrahman@gmail.com	Software	Networking	1	Co-Sheets	1000
11	NULL	NULL	NULL	NULL	NULL	23793	Co-Sheets	1500
12	NULL	NULL	NULL	NULL	NULL	12	Co-Maps	4000
13	NULL	NULL	NULL	NULL	NULL	6	Co-Present	2500
14	NULL	NULL	NULL	NULL	NULL	7	Chat Service	5000
15	NULL	NULL	NULL	NULL	NULL	12467	Co-Calendar	1000
16	NULL	NULL	NULL	NULL	NULL	13	Co-Weather	1000
17	NULL	NULL	NULL	NULL	NULL	5	Co-Forms	1500
18	NULL	NULL	NULL	NULL	NULL	19203	Co-Docs	2000
19	NULL	NULL	NULL	NULL	NULL	14789	Co-Note	3000
20	NULL	NULL	NULL	NULL	NULL	11	Co-Calendar	5000
21	NULL	NULL	NULL	NULL	NULL	8	Video Call S...	7000

Query 6:

Description : this query retrieves the id , name , email , type of product supplied and industry name of the supplier which supplied products to us specifying the supplier industry type name which is the IT Industry.

Actual code:

```
Select Supplier_ID , Fname + ' ' + Lname AS "Supplier Name", Email , Type, Industry_name
From Supplier
where Supplier_ID in
(
    Select Product_ID
    From Product
    Where Industry_name = 'IT'
);
```

Results		Messages			
	Supplier_ID	Supplier Name	Email	Type	Industry_name
1	2	Mahmoud Mohamed	mahmoud@gmail.com	Software	IT
2	3	Sara Ahmed	sara@gmail.com	Cloud	IT
3	5	Noor Amr	noor@gmail.com	Software	IT
4	6	Abdallah Ali	abdallah@gmail.com	Cloud	IT

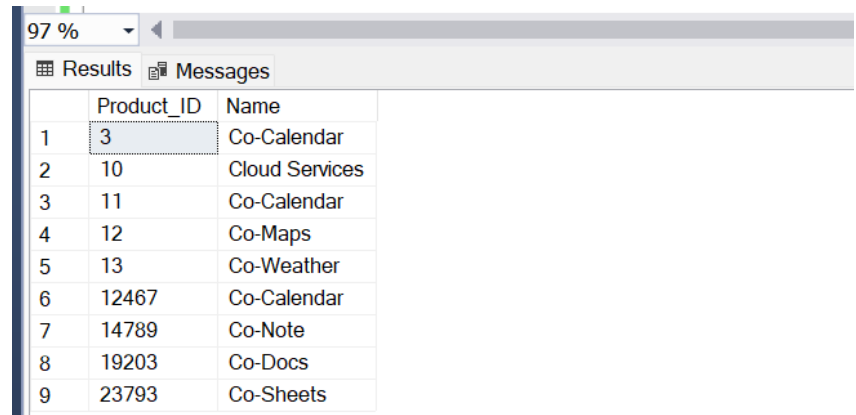
Query 7:

Description :this query retrieves the product name and id which was not bought by a customer.

Actual code:

Create View X AS

```
Select Product_ID , Name
From Product
Where Product_ID not IN
(
    Select Product_ID
    From Interact_with
);
Select * From X ;
```



97 %

Results Messages

	Product_ID	Name
1	3	Co-Calendar
2	10	Cloud Services
3	11	Co-Calendar
4	12	Co-Maps
5	13	Co-Weather
6	12467	Co-Calendar
7	14789	Co-Note
8	19203	Co-Docs
9	23793	Co-Sheets

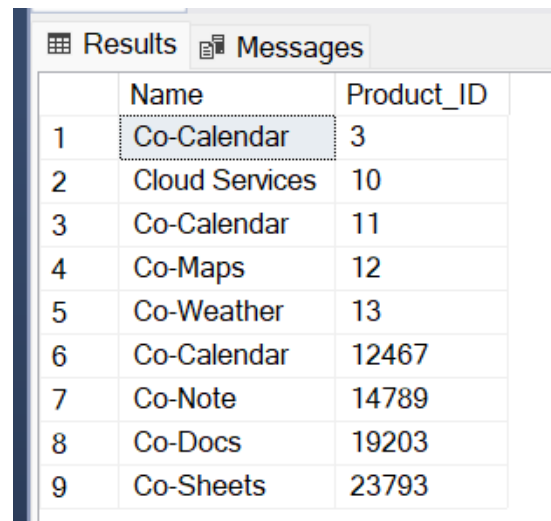
Query 8:

Description :this query retrieves the product name and id which was not bought by a customer but without duplication.

Actual code:

Create View I AS

```
Select Distinct Name , Product_ID
From Product
Where Product_ID not IN
(
    Select Product_ID
    From Interact_with
);
Select * From I ;
```



Results Messages

	Name	Product_ID
1	Co-Calendar	3
2	Cloud Services	10
3	Co-Calendar	11
4	Co-Maps	12
5	Co-Weather	13
6	Co-Calendar	12467
7	Co-Note	14789
8	Co-Docs	19203
9	Co-Sheets	23793

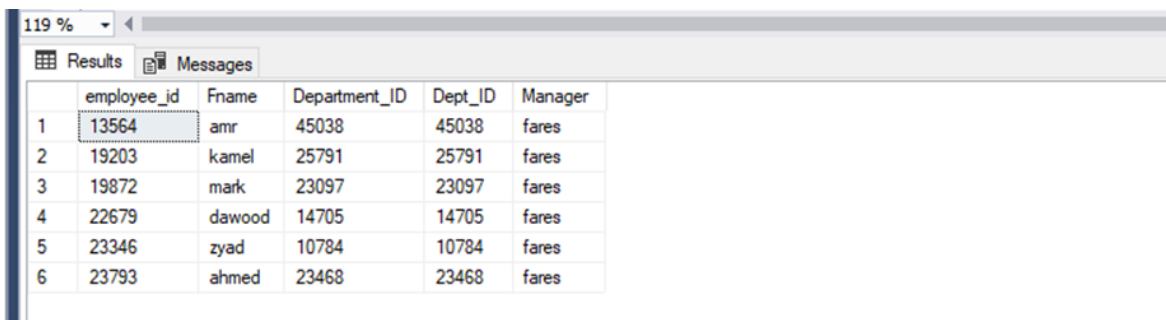
Nadine (230223)

Query 1:

Description : This query retrieves all the employees their manager's name is fares

Actual code :

```
select e.employee_id,e.Fname,e.Department_ID, d.Dept_ID,d.Manager
from Employee e left join Department d
on e.Department_ID=d.Dept_ID
where d.Manager='fares';
```



The screenshot shows a database query results window. The window has a title bar with a zoom level of 119%. Below the title bar, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with the following columns: employee_id, Fname, Department_ID, Dept_ID, and Manager. The table contains 6 rows of data, all of which have 'fares' as the manager. The first row is highlighted with a dashed border.

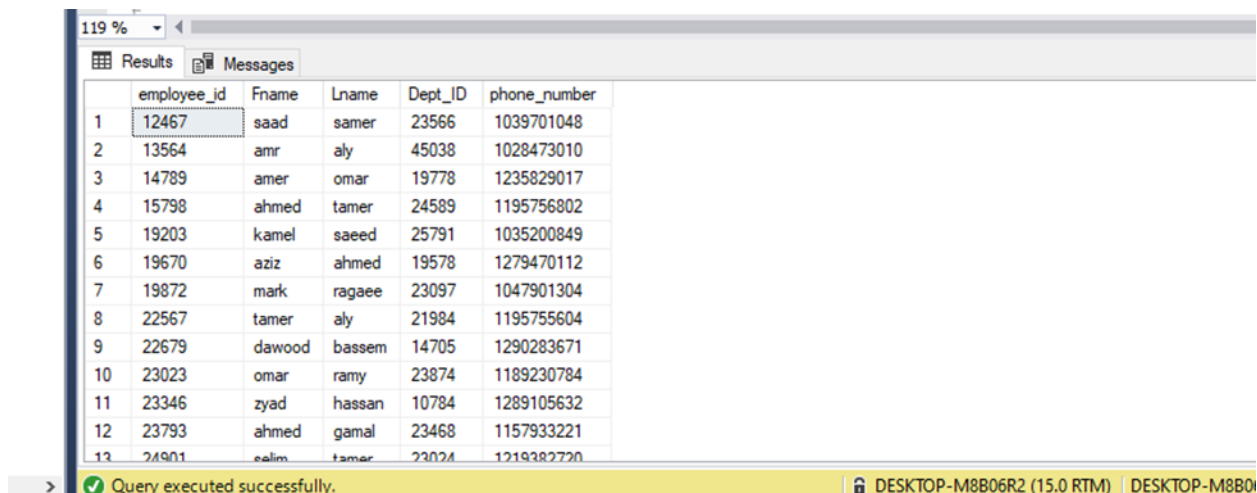
	employee_id	Fname	Department_ID	Dept_ID	Manager
1	13564	amr	45038	45038	fares
2	19203	kamel	25791	25791	fares
3	19872	mark	23097	23097	fares
4	22679	dawood	14705	14705	fares
5	23346	zyad	10784	10784	fares
6	23793	ahmed	23468	23468	fares

Query 2 :

Description : This query retrieves the ID, names and the ID of the department that they are working on, and the phone numbers of all the employees.

Actual code:

```
SELECT e.employee_id, e.Fname, e.Lname, d.Dept_ID,  
(  
    SELECT p.phone_number  
    FROM Employee_Phone p  
    WHERE e.employee_id = p.phone_ID  
)as phone_number  
FROM Employee e  
full JOIN Department d  
ON e.Department_ID = d.Dept_ID;
```



	employee_id	Fname	Lname	Dept_ID	phone_number
1	12467	saad	samer	23566	1039701048
2	13564	amr	aly	45038	1028473010
3	14789	amer	omar	19778	1235829017
4	15798	ahmed	tamer	24589	1195756802
5	19203	kamel	saeed	25791	1035200849
6	19670	aziz	ahmed	19578	1279470112
7	19872	mark	ragae	23097	1047901304
8	22567	tamer	aly	21984	1195755604
9	22679	dawood	bassem	14705	1290283671
10	23023	omar	ramy	23874	1189230784
11	23346	zyad	hassan	10784	1289105632
12	23793	ahmed	gamal	23468	1157933221
13	24901	salim	tamer	23024	1219382720

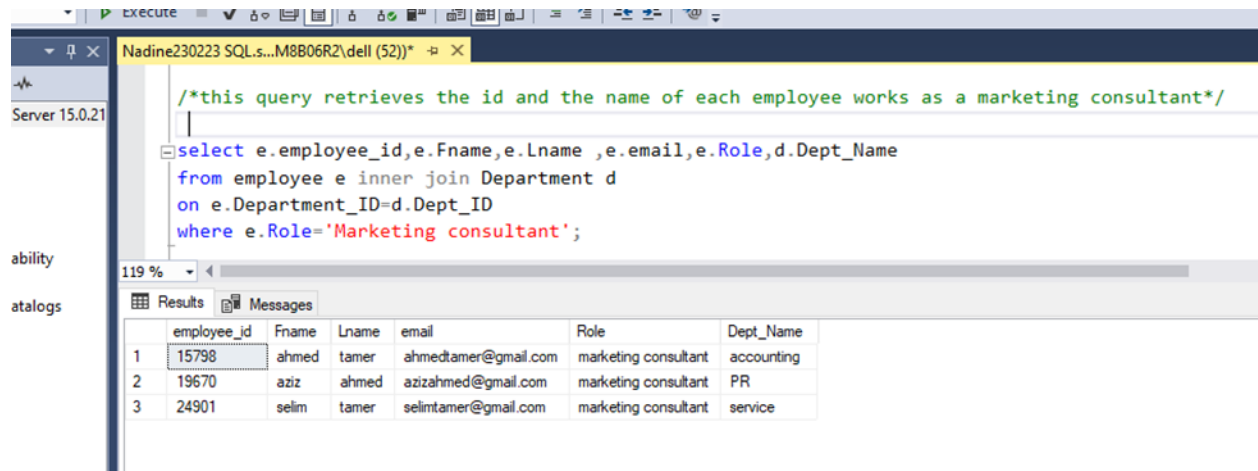
Query executed successfully.

Query 3:

Description : This query retrieves the id and the name of each employee works as a marketing consultant

Actual code:

```
select e.employee_id,e.Fname,e.Lname ,e.email,e.Role,d.Dept_Name
from employee e inner join Department d
on e.Department_ID=d.Dept_ID
where e.Role='Marketing consultant';
```



The screenshot shows the SQL Developer interface. The top pane displays a SQL query with a comment: `/*this query retrieves the id and the name of each employee works as a marketing consultant*/`. The query is: `select e.employee_id,e.Fname,e.Lname ,e.email,e.Role,d.Dept_Name from employee e inner join Department d on e.Department_ID=d.Dept_ID where e.Role='Marketing consultant';`. The bottom pane shows the 'Results' tab with a table containing 3 rows and 7 columns: employee_id, Fname, Lname, email, Role, and Dept_Name. The first row is highlighted.

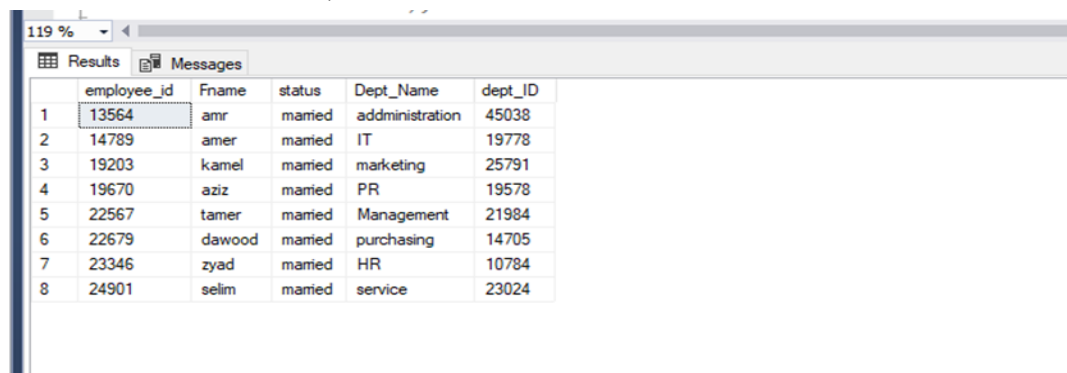
	employee_id	Fname	Lname	email	Role	Dept_Name
1	15798	ahmed	tamer	ahmedtamer@gmail.com	marketing consultant	accounting
2	19670	aziz	ahmed	azizahmed@gmail.com	marketing consultant	PR
3	24901	selim	tamer	selimtamer@gmail.com	marketing consultant	service

Query 4:

Description : This query retrieves the name, id, department and the status to get the relationship that their status =married

Actual code:

```
select e.employee_id, e.Fname, e.status, d.Dept_Name,d.dept_ID
FROM Employee e inner join Department d
on e.Department_ID=d.Dept_ID
where e.employee_id in(select e2.employee_id
from employee e2
where e2.status='married');
```



	employee_id	Fname	status	Dept_Name	dept_ID
1	13564	amr	married	administration	45038
2	14789	amer	married	IT	19778
3	19203	kamel	married	marketing	25791
4	19670	aziz	married	PR	19578
5	22567	tamer	married	Management	21984
6	22679	dawood	married	purchasing	14705
7	23346	zyad	married	HR	10784
8	24901	selim	married	service	23024

Query 5:

Description : This query retrieves the salary of each employee who has a salary more than 1900

Actual code:

```
select e.salary, e.Fname,e.employee_id,e.department_id
from Employee e
where Salary>(select avg(1900)
from Employee
where Department_ID= Employee.Department_ID)
```

er 15.0.21

119 %

Results Messages

	salary	Fname	employee_id	department_id
1	2450	saad	12467	23566
2	2300	amr	13564	45038
3	2300	amer	14789	19778
4	2300	ahmed	15798	24589
5	2300	aziz	19670	19578
6	5000	mark	19872	23097
7	2400	tamer	22567	21984
8	2500	dawood	22679	14705
9	2400	omar	23023	23874
10	2300	zyad	23346	10784
11	2300	ahmed	23793	23468
12	2300	selim	24901	23024

Query 6:

Description : This query retrieves the names and the phone numbers of the employees that their salaries are more than 3000

Actual code:

```
select e.Fname,p.phone_number,e.salary
from employee e right join employee_phone p
on e.employee_id=p.phone_ID
where e.salary>3000;
```



The screenshot shows a database query result window. At the top, there is a zoom level of 119%. Below the zoom level, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with three columns: 'Fname', 'phone_number', and 'salary'. The table contains one row with the following data: '1' in the first column, 'mark' in the second column, and '5000' in the third column.

	Fname	phone_number	salary
1	mark	1047901304	5000

Nouran (235185)

Query 1 :

Description :the query gets the campaign sorted from lower budget to the highest budget campaigns including the campaign ID, budget formatted with the currency, and the type of the campaign.

Actual code:

```
Select ID,FORMAT(Budget_Amount, 'C',Budget_Currency) as Budget ,Type_  
from campaign  
order by Budget_Amount asc ;
```

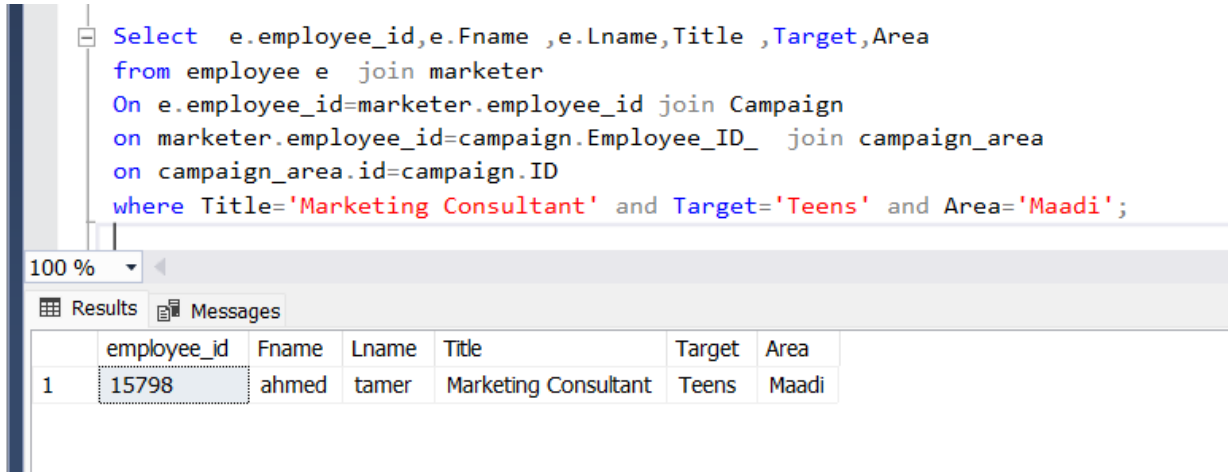
Results Messages			
	ID	Budget	Type_
1	99	20,000,000.00.م.ع	PR
2	125	20 000 000,00 €	PR
3	120	\$50,000,000.00	Social Media
4	130	100,000,000.00.م.ع	Social Media
5	122	\$200,000,000.00	Paid advertisements
6	100	\$200,000,000.00	Email Marketing
7	127	230 000 000,00 €	Social Media

Query 2 :

Description : The query retrieves all the marketing consultants employees that run a campaign and targets teens in the Maadi area. (ID ,First name ,last name ,The title of the marketer ,The target of the Campaign ,and the targeted Area).

Actual code:

```
Select e.employee_id,e.Fname ,e.Lname,Title ,Target,Area
from employee e join marketer
On e.employee_id=marketer.employee_id join Campaign
on marketer.employee_id=campaign.Employee_ID_ join campaign_area
on campaign_area.id=campaign.ID
where Title='Marketing Consultant' and Target='Teens' and Area='Maadi';
```



The screenshot shows a SQL query editor with a query window and a results window. The query window displays the SQL code for Query 2. The results window shows a single row of data for the employee with ID 15798.

	employee_id	Fname	Lname	Title	Target	Area
1	15798	ahmed	tamer	Marketing Consultant	Teens	Maadi

Query 3 :

Description : The query retrieves all campaigns along with their IDs ,Area,the first and the last name of the marketer that manages this campaign and the type of the campaign .

Actual code:

```
Select c.ID as 'Campaign id',Area,Fname,Lname,Type_  
from Campaign c inner join Campaign_Area  
on c.ID=Campaign_Area.ID;
```

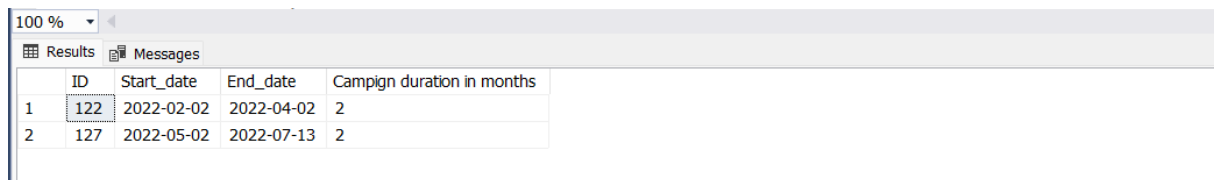
100 %					
Results Messages					
	Campaign id	Area	Fname	Lname	Type_
1	99	ElNozha	mohamed	ahmed	PR
2	100	Sheraton	tamer	ahmed	Email Marketing
3	120	Madinaty	samer	saad	Social Media
4	122	ElNozha	samer	saad	Paid advertisements
5	125	Nasr City	tamer	ahmed	PR
6	127	Maadi	tamer	ahmed	Social Media
7	130	Nasr City	samer	saad	Social Media

Query 4 :

Description : The query retrieves all the campaigns(ID) whose duration is longer than one month along with the Start , End date and duration in months of this campaign .

Actual code:

```
Select ID,Start_date,End_date ,datediff(MONTH,Start_date,End_date) as 'Campaign duration in months'
from Campaign
where datediff(MONTH,Start_date,End_date) >1;
```



The screenshot shows a database query results window. At the top, there is a zoom level dropdown set to '100 %'. Below it, there are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a table with the following data:

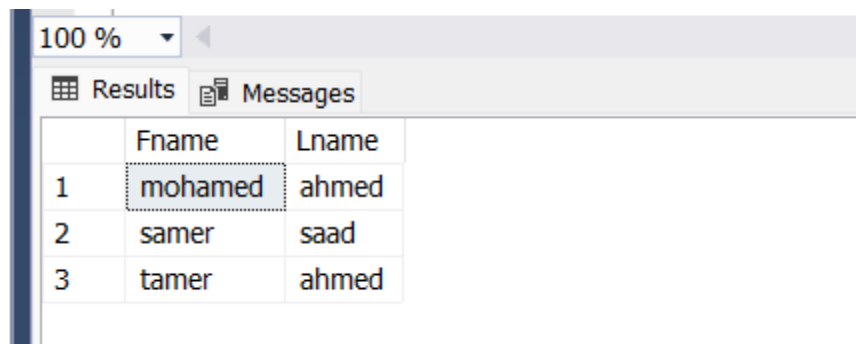
	ID	Start_date	End_date	Campaign duration in months
1	122	2022-02-02	2022-04-02	2
2	127	2022-05-02	2022-07-13	2

Query 5 :

Description : The query retrieves the marketer who is responsible for each campaign that manages one or more customers (Fname ,Lname).

Actual code:

```
SELECT Distinct Fname, Lname  
FROM Campaign  
WHERE EXISTS (SELECT *  
FROM Potential_customer  
WHERE ID = Campaign_ID);
```



	Fname	Lname
1	mohamed	ahmed
2	samer	saad
3	tamer	ahmed

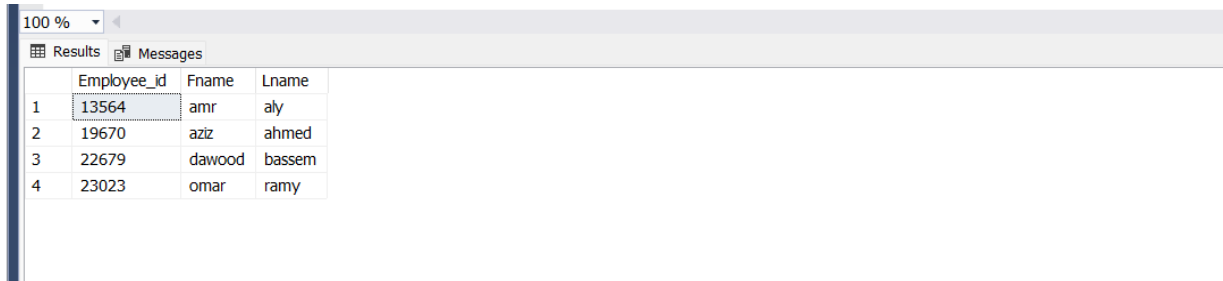
Query 6 :

Description : The query get all Marketers that are not managing any campaigns with their IDs.

-Actual code:

```
select employee.Employee_id,Fname,Lname
```


from Marketer,employee
where employee.Employee_id= Marketer.Employee_id and marketer.Employee_id not IN
(Select Employee_ID_
from campaign) ;



The screenshot shows a database query results window. At the top, there is a zoom level dropdown set to '100 %'. Below it are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a table with the following data:

	Employee_id	Fname	Lname
1	13564	amr	aly
2	19670	aziz	ahmed
3	22679	dawood	bassem
4	23023	omar	ramy