



Write your project title

اكتب عنوان المشروع

## EMPLOYEE MANGMENT

STUDENT NAME	STUDENT ID
جوري علي العمري	445002323
لمياء البارقي	445002239
فاطمه الجوير	445002377
رويدا المطري	445002281

اسم الاستاذة

فدى محمد أبو حميد

Table of Contents

Introduction .....3

Part 1 - Conceptual Phase.....4

Part 2 – Logical Phase.....5

Part 3 – Physical Phase.....6

Part 4 – Storage and Security Phase.....9

## Introduction

### Introduce your project

اكتب مقدمة ووصف للمشروع

## Employee Management

The system organizes and tracks employees, departments, divisions, and projects to streamline resource management.

### Key Features:

Departments: Managed by divisions and defined by ID and name.

Employees: Fixed or rovers, with detailed records (ID, name, contact, etc.).

Divisions: Oversee multiple departments, run by one employee, and defined by code and name.

Projects: Employees can work on multiple projects, defined by code and name.

Objective: Simplify resource allocation, track performance, and provide actionable insights for efficient management.

## Part 1- Conceptual Phase:

write a description of the EER model

1. EMPLOYEE (entity): (Emp\_id(PK),Emp\_mobile,Emp\_adress,Emp\_grade)

ROVERS (entity):Emp\_id(FK)

FIXED (entity):Emp\_id(FK)

2. DIVISION (entity):(Div\_code(PK),Div\_Name,Number\_Department)

3. DEPARTMENT (entity):(Dep\_num(PK),Invo\_name)

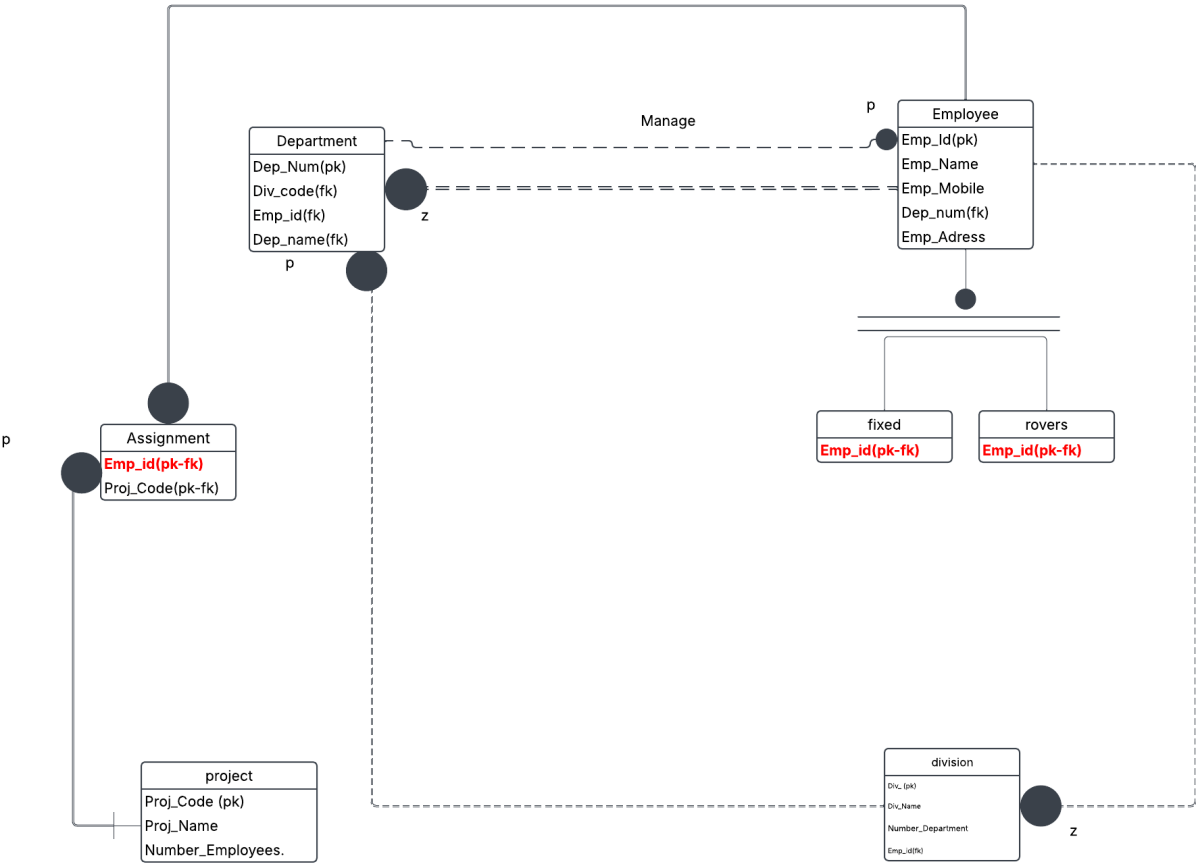
4. EMPLOYEE\_PROJECT(entity):(Emp\_num(PK+FK),Pro\_Code(PK+FK))

5. PROJECT(entity):(Pro\_Name,Number\_Employees)

Part 2 – Logical Phase:

Draw the EER model

with Microsoft word or Microsoft Visio or any other designing software

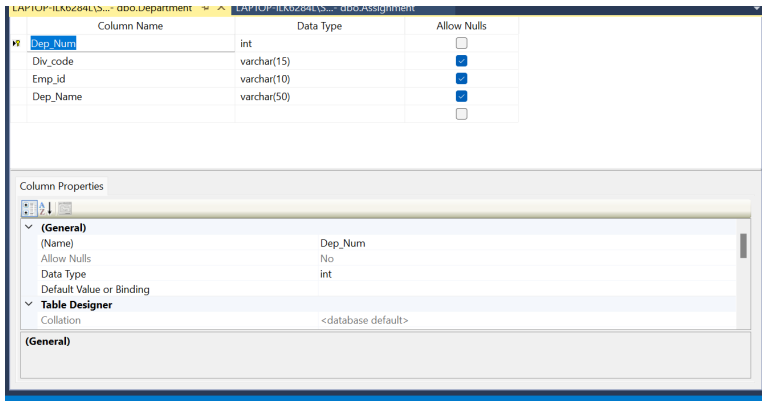
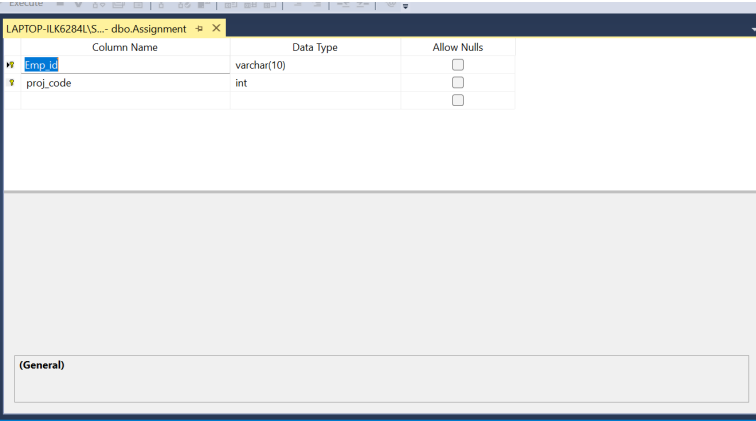


# Part 3 – Physical Phase

- 1. Create the **database** and **Tables** , enter your data and precise types and lengths using Microsoft SQL Server software.

Add a screen shot for each table you created

You should at least create one table by coding



Column Name	Data Type	Allow Nulls
Div	varchar(15)	<input type="checkbox"/>
Div_Name	varchar(50)	<input checked="" type="checkbox"/>
Number_Dempartment	int	<input checked="" type="checkbox"/>
Emp_id	varchar(10)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties

(General)

(Name)

Div\_

Allow Nulls

No

Data Type

varchar

Default Value or Binding

Length

15

Table Designer

(General)

Column Name	Data Type	Allow Nulls
Emp_id	varchar(10)	<input type="checkbox"/>
Emp_name	varchar(50)	<input checked="" type="checkbox"/>
Emp_Mobile	varchar(15)	<input checked="" type="checkbox"/>
Dep_num	int	<input checked="" type="checkbox"/>
Emp_Address	varchar(100)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties

(General)

(Name)

Emp\_id

Allow Nulls

No

Data Type

varchar

Default Value or Binding

Length

10

Table Designer

(General)

Column Name	Data Type	Allow Nulls
Emp_id	varchar(10)	<input type="checkbox"/>
		<input type="checkbox"/>

Column Properties

(General)

(Name)

Emp\_id

Allow Nulls

No

Data Type

varchar

Default Value or Binding

Length

10

Table Designer

(General)

Column Name	Data Type	Allow Nulls
proj_code	int	<input type="checkbox"/>
proj_Name	varchar(50)	<input checked="" type="checkbox"/>
Number_Employee	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties

(General)

(Name)

proj\_code

Allow Nulls

No

Data Type

int

Default Value or Binding

Table Designer

Collation

<database default>

(General)

7

LAPTOP-ILK6284\VS\_loyee - dbo.RoversLAPTOP-ILK6284\VS\_oyee - dbo.projectLAPTOP-ILK6284\VS\_ployee - dbo.Fixed

	Column Name	Data Type	Allow Nulls
Emp_id		varchar(10)	<input type="checkbox"/>
			<input type="checkbox"/>

Column Properties

General

Table Designer

(Name)	Emp_id
Allow Nulls	No
Data Type	varchar
Default Value or Binding	
Length	10



2-Fill your tables with appropriate values with your choice.

LAPTOP-ILK6284\LS... - dbo.Assignment SQLQuery1.sql - LA...K6284\school (63) LAPTOP-ILK6284\LS... - dbo.Rovers

Emp_id	proj_code
1	8
2	7
3	6
NULL	NULL

of 3

LAPTOP-ILK6284\LS... - dbo.Department LAPTOP-ILK6284\LS... - dbo.Assignment SQLQuery1.sql - LA...K6284\school (63)

Dep_Num	Div_code	Emp_id	Dep_Na...
1	8	1	HR
12	6	2	IT
NULL	NULL	NULL	NULL

of 2

Execute LAPTOP-ILK6284\LS... - dbo.Division LAPTOP-ILK6284\LS... - dbo.Department LAPTOP-ILK6284\LS... - dbo.Assignment

Div	Div_Name	Number...	Emp_id
10	Marketing	11	3
3	IT	9	2
4	HR	5	1
NULL	NULL	NULL	NULL

of 3

LAPTOP-ILK6284\LS... - dbo.Employee LAPTOP-ILK6284\LS... - dbo.Division LAPTOP-ILK6284\LS... - dbo.Department

Emp_id	Emp_na...	Emp_Mo...	Dep_num	Emp_Ad...
1	lamia	05566666	11	Riyadh
2	sara	08888809	11	Riyadh
3	remaa	05667889	12	Jeddah
4	lamia	05566666	11	Riyadh
5	Sara	08888889	12	Riyadh
NULL	NULL	NULL	NULL	NULL

of 5

LAPTOP-ILK6284\LS...ployee - dbo.Fixed		LAPTOP-ILK6284\LS...ee - dbo.Employee		LAPTOP-ILK6284\LS...yee - dbo.Division	
Emp_Id					
4					
5					
6					
NULL					

of 3 | Cell is Modified...

LAPTOP-ILK6284\LS...oyee - dbo.project			LAPTOP-ILK6284\LS...ployee - dbo.Fixed			LAPTOP-ILK6284\LS...ee - dbo.Employee		
proj_code	proj_Na...	Number...						
6	Online B...	2						
7	E-Learnin...	3						
8	Website ...	5						
NULL	NULL	NULL						

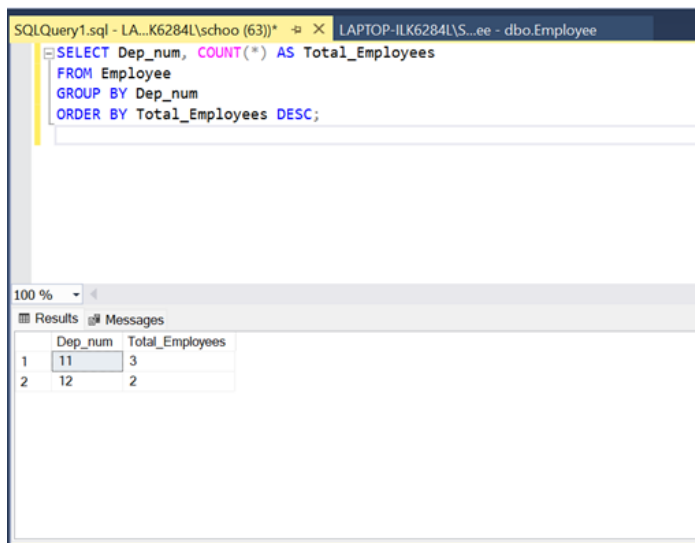
of 3 |

LAPTOP-ILK6284\LS...loyee - dbo.Rovers		LAPTOP-ILK6284\LS...oyee - dbo.project		LAPTOP-ILK6284\LS...ployee - dbo.Fixed	
Emp_Id					
2					
3					
NULL					

of 3 |

2. Write SQL statements(codes) corresponding to the following:

SQL query statement 6 statements



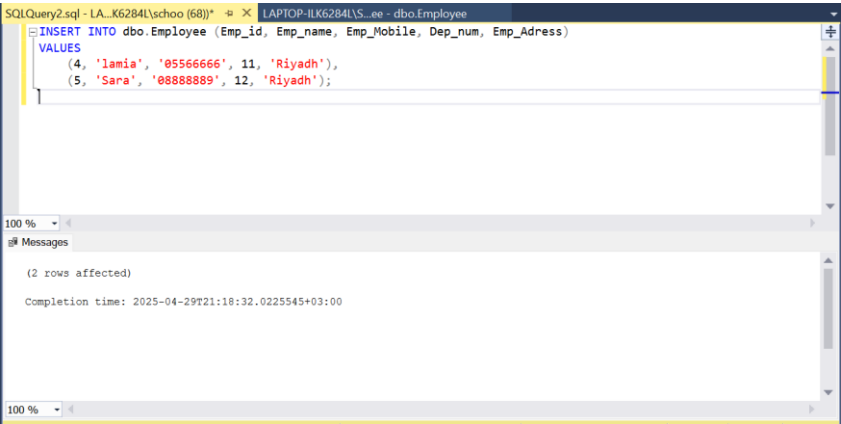
The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor contains the following SQL statement:

```
SELECT Dep_num, COUNT(*) AS Total_Employees
FROM Employee
GROUP BY Dep_num
ORDER BY Total_Employees DESC;
```

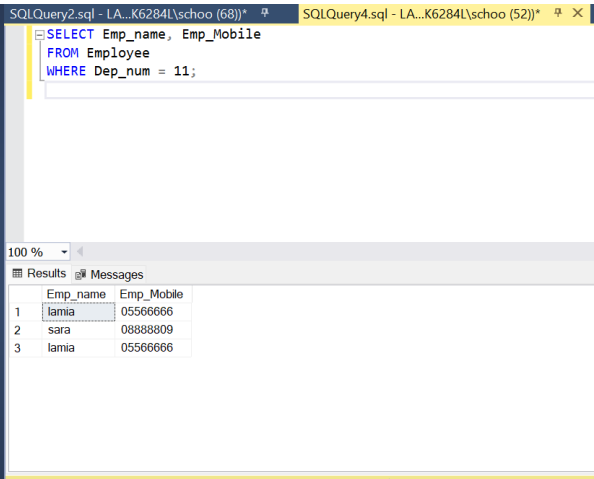
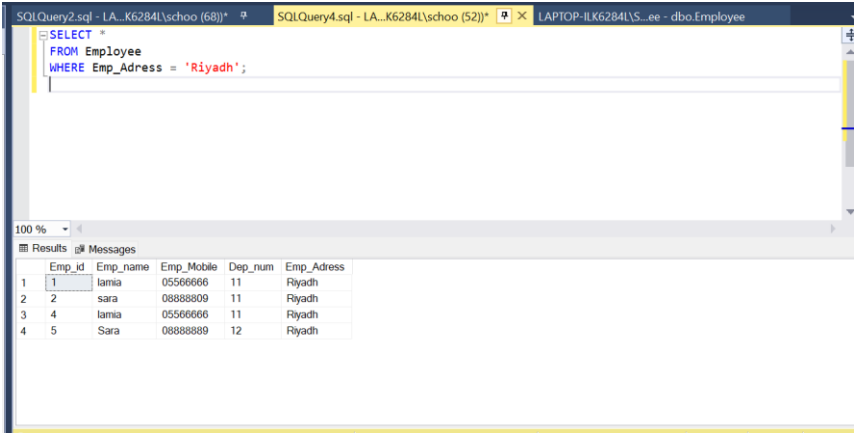
The results pane displays the output of the query as a table with two columns: Dep\_num and Total\_Employees. The results are ordered by Total\_Employees in descending order.

	Dep_num	Total_Employees
1	11	3
2	12	2

1.Insert rows to tables with your choice.



1. Read data from tables with WHERE clause.



## 2. Read data from tables using **GROUP BY** and **ORDER BY**.

The screenshot shows a SQL query window with the following code:

```
SELECT Dep_num, COUNT(*) AS Total_Employees
FROM Employee
GROUP BY Dep_num
ORDER BY Dep_num;
```

Below the query window, the 'Results' tab displays the following data:

	Dep_num	Total_Employees
1	11	3
2	12	2

A status bar at the bottom indicates: 'Query executed successfully. LAPTOP-ILK6284L\SQLEXPRESS ... LAPTOP-ILK6284L\schoo ... Employee 00:00:00 2 rows'.

## 3. Add **CONSTRAINT** for **FOREIGN KEY**.

The screenshot shows a SQL query window with the following code:

```
ALTER TABLE Employee
ADD CONSTRAINT FK_Department_New
FOREIGN KEY (Dep_num)
REFERENCES Department (Dep_num);
```

Below the query window, the 'Messages' tab displays the following text:

Commands completed successfully.  
Completion time: 2025-04-29T21:48:16.6763614+03:00

## 4. Read data from 2 table using **JOIN**.

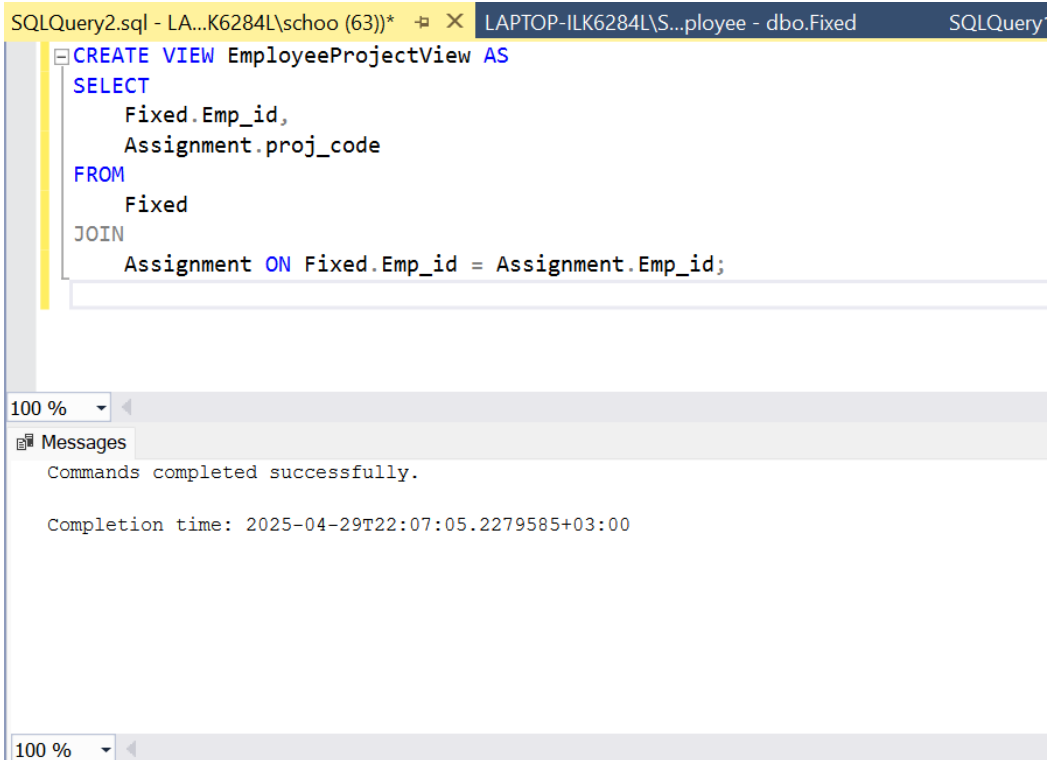
The screenshot shows a SQL query window with the following code:

```
SELECT
    Assignment.Emp_id,
    Employee.Emp_name,
    Assignment.proj_code
FROM
    Assignment
JOIN
    Employee ON Assignment.Emp_id = Employee.Emp_id;
```

Below the query window, the 'Results' tab displays the following data:

	Emp_id	Emp_name	proj_code
1	1	lamia	8
2	2	sara	7
3	3	remaa	6

5. Create a **VIEW** with your choice.



The screenshot displays the SQL Server Enterprise Manager interface. At the top, a tab bar shows 'SQLQuery2.sql - LA...K6284L\schoo (63))' and 'LAPTOP-ILK6284L\S...ployee - dbo.Fixed'. The main editor window contains the following SQL code:

```
CREATE VIEW EmployeeProjectView AS
SELECT
    Fixed.Emp_id,
    Assignment.proj_code
FROM
    Fixed
JOIN
    Assignment ON Fixed.Emp_id = Assignment.Emp_id;
```

Below the editor, a 'Messages' pane shows the execution results:

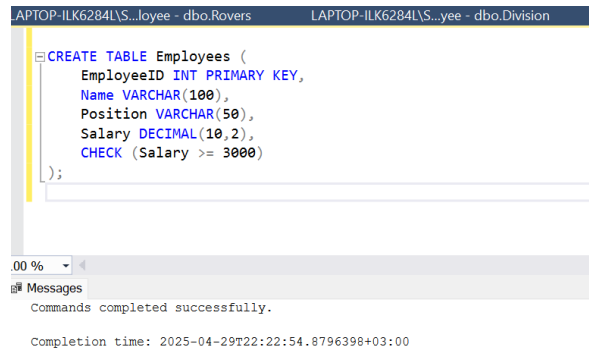
```
Commands completed successfully.

Completion time: 2025-04-29T22:07:05.2279585+03:00
```

Both the editor and the messages pane have a zoom level set to '100 %'.

## Part 4 – Storage and Security Phase :

### ▪ CHECK CONSTRAINT code



The screenshot displays the SQL Server Enterprise Manager interface. At the top, two server names are visible: 'LAPTOP-ILK62841\S...loyee - dbo.Rovers' and 'LAPTOP-ILK62841\S...yee - dbo.Division'. The main pane shows a SQL script with the following code:

```
CREATE TABLE Employees (  
    EmployeeID INT PRIMARY KEY,  
    Name VARCHAR(100),  
    Position VARCHAR(50),  
    Salary DECIMAL(10,2),  
    CHECK (Salary >= 3000)  
);
```

Below the script, a 'Messages' pane shows the following output:

Commands completed successfully.  
Completion time: 2025-04-29T22:22:54.8796398+03:00

## ▪TRIGGER code

```
CREATE TRIGGER PreventManagerDeletion
ON Employees
AFTER DELETE
AS
BEGIN
    IF EXISTS (
        SELECT * FROM deleted WHERE Position = 'Manager'
    )
    BEGIN
        RAISERROR ('Cannot delete a Manager!', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-29T22:25:55.0887277+03:00

## ▪ give **OR** take away (permissions) code.

```
CREATE LOGIN user_lulu WITH PASSWORD = 'StrongPass123!';
CREATE USER user_lulu FOR LOGIN user_lulu;
GRANT SELECT, INSERT, UPDATE ON Employees TO user_lulu;
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-29T22:31:12.5161766+03:00

100 %

Query executed successfully. LAPTOP-ILK6284\SQLEXPRESS ... LAPTOP-ILK6284\school ... master 00:00:00 0 rows