



SCHOOL OF: INFOMATICS
DEPARTEMENT OF: COMPUTER SCIENCES
COURSE TITLE: ADAVACED DATABASE
PROJECT TITLE: EMPLOYEE PAYROLL SYSTEM

GROUP MEMBERS

NUMBER	FNAME	MNAME	ID
1	ASHENAFI	DEGIF	UGR/91463/16
2	DEGINESH	DALGA	UGR/91619/16
3	MAHLET	AMSALU	UGR/91944/16
4	METAGES	OROBA	UGR/92705/16

Create Database table

```
create database Employepayrolls;
```

```
use Employeeparolls;
```

```
create table Departement(
```

```
DepartementID int primary key ,
```

```
Departementname varchar(100) not null unique,
```

```
Budgetlimit int not null);
```

```
create table Employees(
```

```
EmployeeID int primary key ,
```

```
Employeeename varchar(100) not null,
```

```
DepartementID int,
```

```
Jobtitle varchar(100),
```

```
salary decimal(10,2) check (salary>=0),
```

```
Hiredate date,
```

```
foreign key (DepartementID) References Departement(DepartementID)
```

```
);
```

Cont...

```
create table Payroll(  
    PayrollID int primary key,  
    EmployeeID int,  
    Paymentdate date not null,  
    Paymentamount decimal(10,2)  
    foreign key(EmployeeID) References Employees(EmployeeID)  
);
```

cont...

```
insert into Departement values(1,'Computer science',10);
```

```
insert into Departement values(2,'Finance',16);
```

```
insert into Departement values(3,'IS',60);
```

```
insert into Employees values(112,'Adanech', 1,'Manager','20000.00','2019-07-11');
```

```
insert into Employees values(113,'Debebe',2,'Software Engineer','30000.00','2020-01-06');
```

```
insert into Employees values(114,'Abebe',3, 'Accountant','40000.00','2017-04-25');
```

```
insert into Payroll values(123,112,'2025-02-1','15000.00');
```

```
insert into Payroll values(124,113,'2025-02-1','16000.00');
```

```
insert into Payroll values(125,114,'2025-02-1','17000.00');
```

cont...

1 select all employee working in the company

```
select * from Employees;
```



A screenshot of a SQL query results window. The window has a title bar with '100 %' and a back arrow. Below the title bar are two tabs: 'Results' (selected) and 'Messages'. The main area is a grid table with the following data:

	EmployeeID	EmployeeName	DepartmentID	JobTitle	Salary	HireDate
1	112	Adanech	1	Manager	23100.00	2019-07-11
2	113	Debebe	2	Software Engineer	34650.00	2020-01-06
3	114	Abebe	3	Accountant	46200.00	2017-04-25

2 select all employees working in 'finance' department

```
select * from Employees where DepartmentID=(select DepartmentID from Department where Departmentname='Finance')
```

cont...

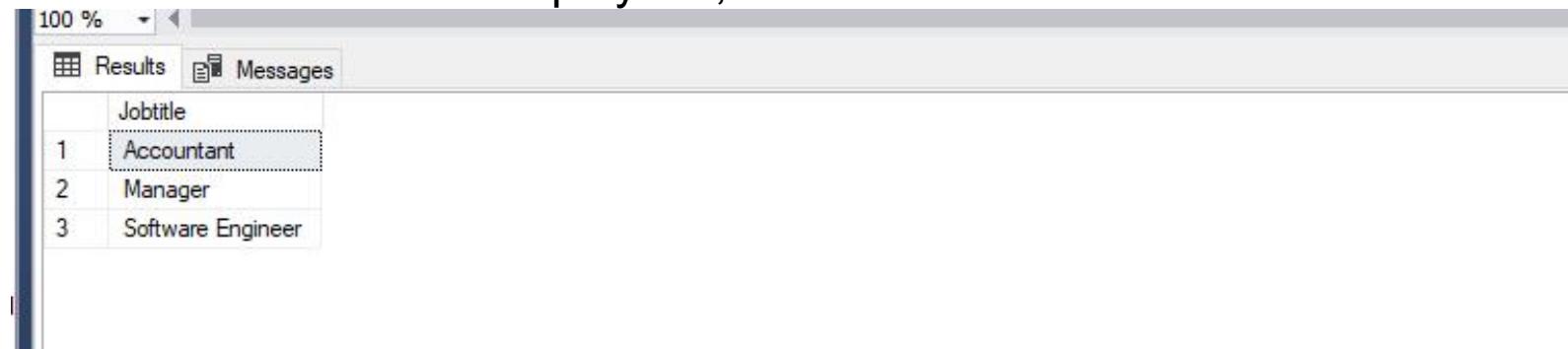


A screenshot of a SQL Server Management Studio (SSMS) results grid. The grid has columns labeled EmployeeID, EmployeeName, DepartmentID, JobTitle, Salary, and HireDate. There is one row of data: EmployeeID 113, EmployeeName Debebe, DepartmentID 2, JobTitle Software Engineer, Salary 34650.00, and HireDate 2020-01-06. The EmployeeID column is highlighted with a red dotted border.

	EmployeeID	EmployeeName	DepartmentID	JobTitle	salary	Hiredate
1	113	Debebe	2	Software Engineer	34650.00	2020-01-06

3 select all distinct job role in company

select distinct Jobtitle from Employees;



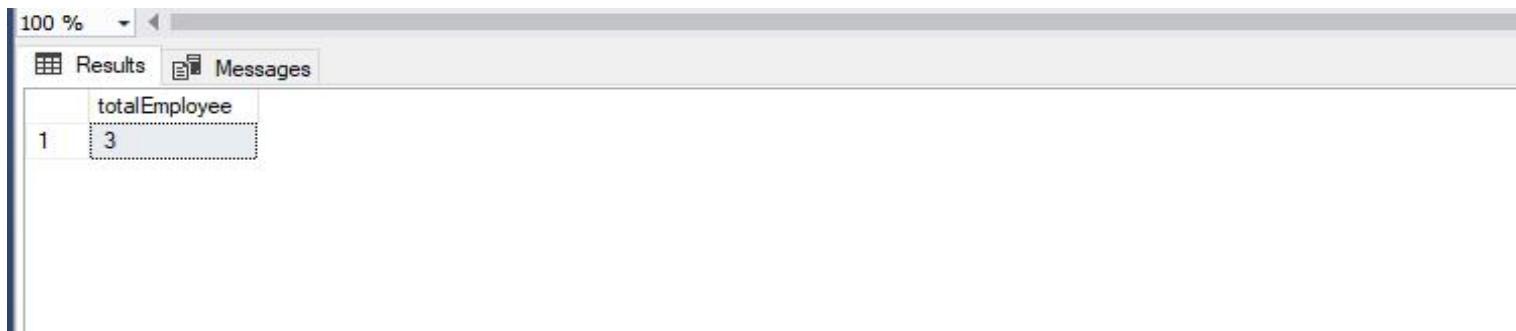
A screenshot of a SQL Server Management Studio (SSMS) results grid. The grid has a single column labeled JobTitle. There are three rows of data: Accountant, Manager, and Software Engineer. The first row, Accountant, is highlighted with a red dotted border.

	Jobtitle
1	Accountant
2	Manager
3	Software Engineer

cont...

4 select the total numbers of employee in the company

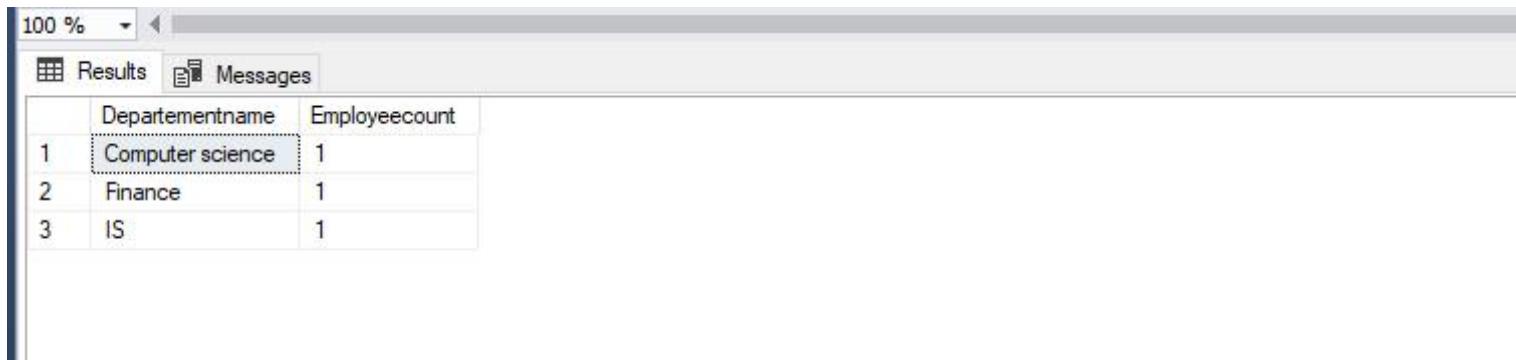
```
select count(*) as totalEmployee from Employees;
```



	totalEmployee
1	3

5 select the departement name and the total number of in each departement

```
select d.Departementname, count(e.EmployeeID) as Employeecount from Departement d  
left join Employees e on d.DepartementID=e.DepartementID group by d.Departementname;
```

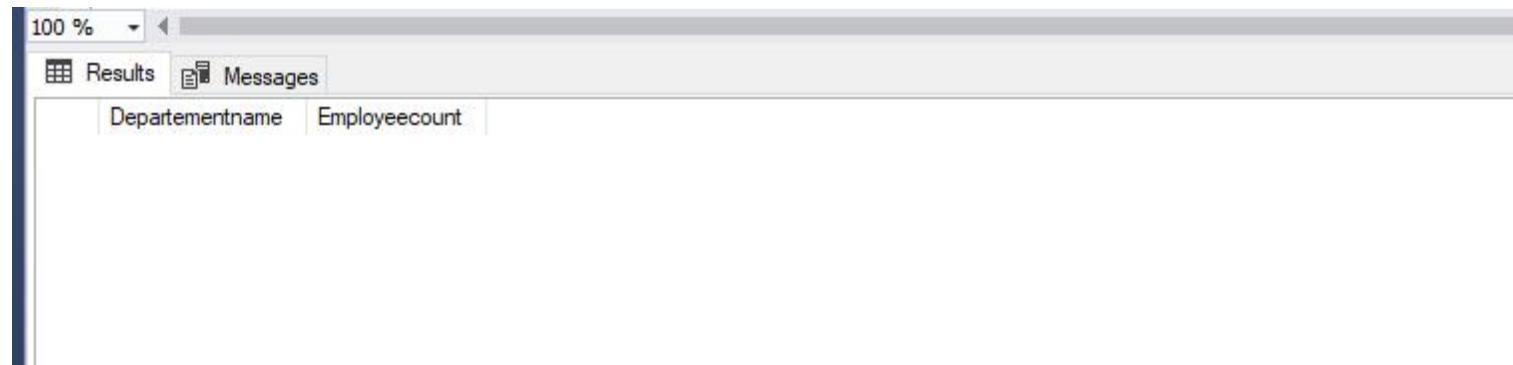


	Departementname	Employeecount
1	Computer science	1
2	Finance	1
3	IS	1

cont...

6 select only departement where the number of employee is greater than 50.

```
select d.Departementname, count(e.EmployeeID) as Employeecount from Departement d  
left join Employees e on d.DepartementID=e.DepartementID group by d.Departementname  
having count(e.EmployeeID) > 50;
```



A screenshot of a SQL query results window. The window has a title bar with '100 %' and a dropdown arrow. Below the title bar are two tabs: 'Results' (selected) and 'Messages'. The main area is a table with two columns: 'Departementname' and 'Employeecount'. There are no rows in the table.

Departementname	Employeecount

7 select each employee's name along with their department and salary

```
select e.EmployeeName, d.Departementname, e.salary from Employees e join Departement d on e.DepartementID=d.DepartementID;
```

cont...

The screenshot shows a results grid from a SQL query. The columns are EmployeeName, DepartmentName, and salary. There are three rows of data:

	EmployeeName	DepartmentName	salary
1	Adanech	Computer science	23100.00
2	Debebe	Finance	34650.00
3	Abebe	IS	46200.00

8 select all department along with the number of employee (including empty department)

```
select d.Departementname, count(e.EmployeeID) as Employeecount  
from Departement d left join Employee e on d.DepartementID=e.DEpartmentID group  
by d.Departementname;
```

The screenshot shows a results grid from a SQL query. The columns are DepartmentName and EmployeeCount. There are three rows of data:

	DepartmentName	EmployeeCount
1	Computer science	1
2	Finance	1
3	IS	1

cont..

9 select all department where the number of employee exceeds the department budgetd limit

```
select d.Departementname,count(e.EmployeeID) as Employeecount,d.Budgetlimit  
from Departement d join Employee e on d.DepartementID=e.DEpartmentID group  
by d.Departementname,d.Budgetlimit having count(e.EmployeeID)>d.Budgetlimit;
```

Departementname	Employeecount	Budgetlimit

10 select all employee who have not received a salary payment in last month

```
select e.Employeeename from Employees e left join Payroll p on e.EmployeeID=p.EmployeeID  
and p.Paymentdate >= DATEADD(MONTH, -1,getdate()) where p.Paymentdate is null
```

cont...

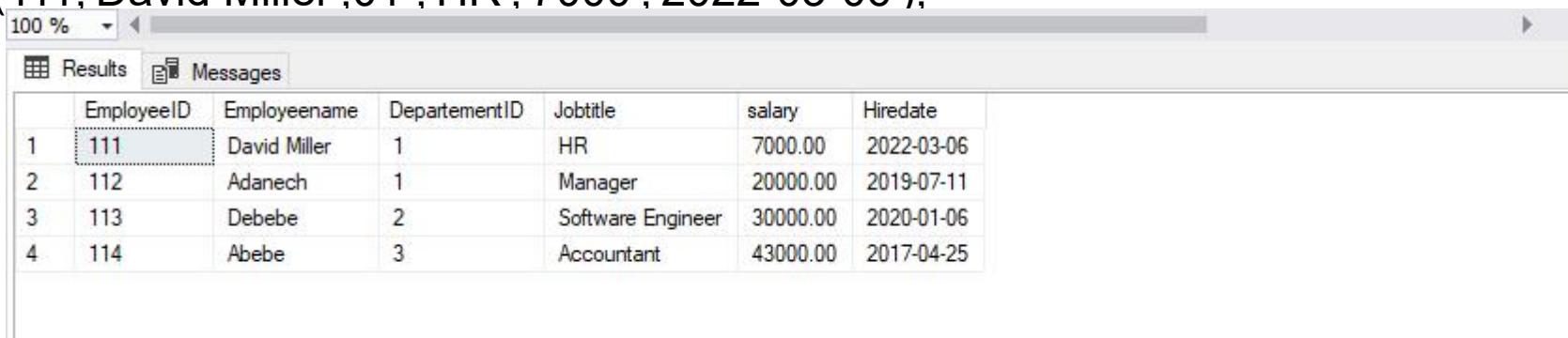


A screenshot of a SQL query results window. The title bar shows "100 %". Below it are two tabs: "Results" (selected) and "Messages". The results table has one column labeled "EmployeeName" and three rows with values "Adanech", "Debebe", and "Abebe".

	EmployeeName
1	Adanech
2	Debebe
3	Abebe

11. Add a new employee named "David Miller" to the "HR" department

```
insert into Employees(EmployeeID,EmployeeName,DepartementID,Jobtitle,salary, Hiredate )
values (111,'David Miller',01 , 'HR','7000','2022-03-06');
```



A screenshot of a SQL query results window showing the updated employee data. The title bar shows "100 %". Below it are two tabs: "Results" (selected) and "Messages". The results table has seven columns: EmployeeID, EmployeeName, DepartementID, Jobtitle, salary, and Hiredate. There are four rows of data:

	EmployeeID	EmployeeName	DepartementID	Jobtitle	salary	Hiredate
1	111	David Miller	1	HR	7000.00	2022-03-06
2	112	Adanech	1	Manager	20000.00	2019-07-11
3	113	Debebe	2	Software Engineer	30000.00	2020-01-06
4	114	Abebe	3	Accountant	43000.00	2017-04-25

cont...

12 add a salary payment of \$3000 for “Sarah Johson”

update Employees set salary=salary + 3000

where EmployeeName='Abebe';

A screenshot of a SQL query results window. The title bar shows "100 %". Below it are two tabs: "Results" (selected) and "Messages". The results table has columns: EmployeeID, EmployeeName, DepartmentID, JobTitle, salary, and Hiredate. There is one row of data: EmployeeID 114, EmployeeName Abebe, DepartmentID 3, JobTitle Accountant, salary 43000.00, and Hiredate 2017-04-25.

	EmployeeID	EmployeeName	DepartmentID	JobTitle	salary	Hiredate
1	114	Abebe	3	Accountant	43000.00	2017-04-25

13 Increase the salary of all employees by 5%.

Update Employee set salary=salary*1.05;

A screenshot of a SQL query results window. The title bar shows "100 %". Below it are two tabs: "Results" (selected) and "Messages". The results table has columns: EmployeeID, EmployeeName, DepartmentID, JobTitle, salary, and Hiredate. There are three rows of data: EmployeeID 112, EmployeeName Adanech, DepartmentID 1, JobTitle Manager, salary 21000.00, and Hiredate 2019-07-11; EmployeeID 113, EmployeeName Debebe, DepartmentID 2, JobTitle Software Engineer, salary 31500.00, and Hiredate 2020-01-06; and EmployeeID 114, EmployeeName Abebe, DepartmentID 3, JobTitle Accountant, salary 42000.00, and Hiredate 2017-04-25.

	EmployeeID	EmployeeName	DepartmentID	JobTitle	salary	Hiredate
1	112	Adanech	1	Manager	21000.00	2019-07-11
2	113	Debebe	2	Software Engineer	31500.00	2020-01-06
3	114	Abebe	3	Accountant	42000.00	2017-04-25

cont...

14 Apply a 10% salary increase to employee who have worked for more than five years.

```
update Employee set salary=salary*1.10 where Hiredate < dateadd(year,-5,getdate())
```

	EmployeeID	Employeeename	DepartementID	Jobtitle	salary	Hiredate
1	112	Adanech	1	Manager	23100.00	2019-07-11
2	113	Debebe	2	Software Engineer	34650.00	2020-01-06
3	114	Abebe	3	Accountant	46200.00	2017-04-25

15 Remove all employee earning below \$1500.

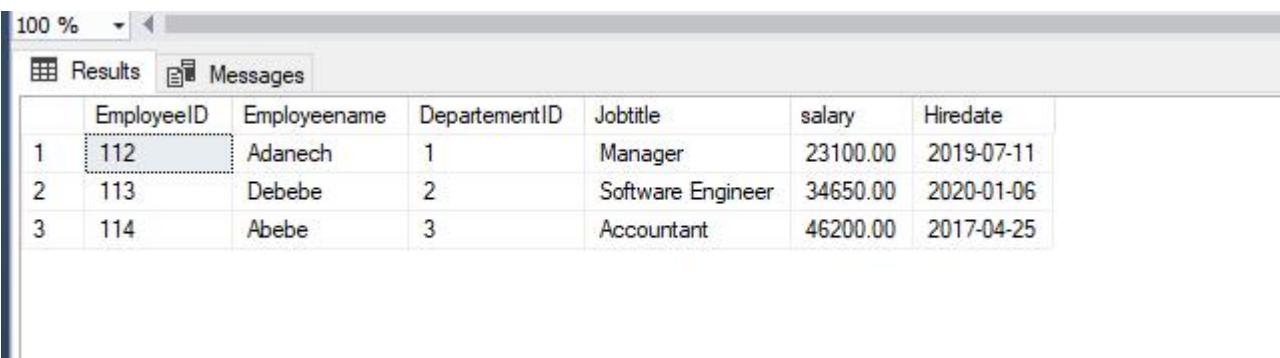
```
delete from Employees where salary < 1500;
```

	EmployeeID	Employeeename	DepartementID	Jobtitle	salary	Hiredate
1	112	Adanech	1	Manager	23100.00	2019-07-11
2	113	Debebe	2	Software Engineer	34650.00	2020-01-06
3	114	Abebe	3	Accountant	46200.00	2017-04-25

cont...

16 Remove all employee from department that have exceed their budgeted employee limit.

```
delete from Employees where DepartementID in ( select d.DepartementID from Departement  
d join Employees e on d.DepartementID = e.DepartementID  
group by d.DepartementID, d.Budgetlimit having count(e.EmployeeID) > d.Budgetlimit );
```



The screenshot shows a Microsoft SQL Server Management Studio (SSMS) window with the title bar "100 %". Below the title bar are two tabs: "Results" (selected) and "Messages". The main area displays a table with the following data:

	EmployeeID	EmployeeName	DepartementID	Jobtitle	salary	Hiredate
1	112	Adanech	1	Manager	23100.00	2019-07-11
2	113	Debebe	2	Software Engineer	34650.00	2020-01-06
3	114	Abebe	3	Accountant	46200.00	2017-04-25

Relational Schema

- Relational Schema for Employee Payroll System
- A relational schema defines the structure of the database, including tables, primary keys (PK), and foreign keys (FK).

Departement table

DepartmentID → Primary Key (PK)

DepartmentName → Unique constraint

BudgetLimit → Budget allocation

Employee table

EmployeeID → Primary Key (PK)

DepartmentID → Foreign Key (FK) referencing Departments(DepartmentID)

Salary → Employee salary

HireDate → Date of hiring

cont...

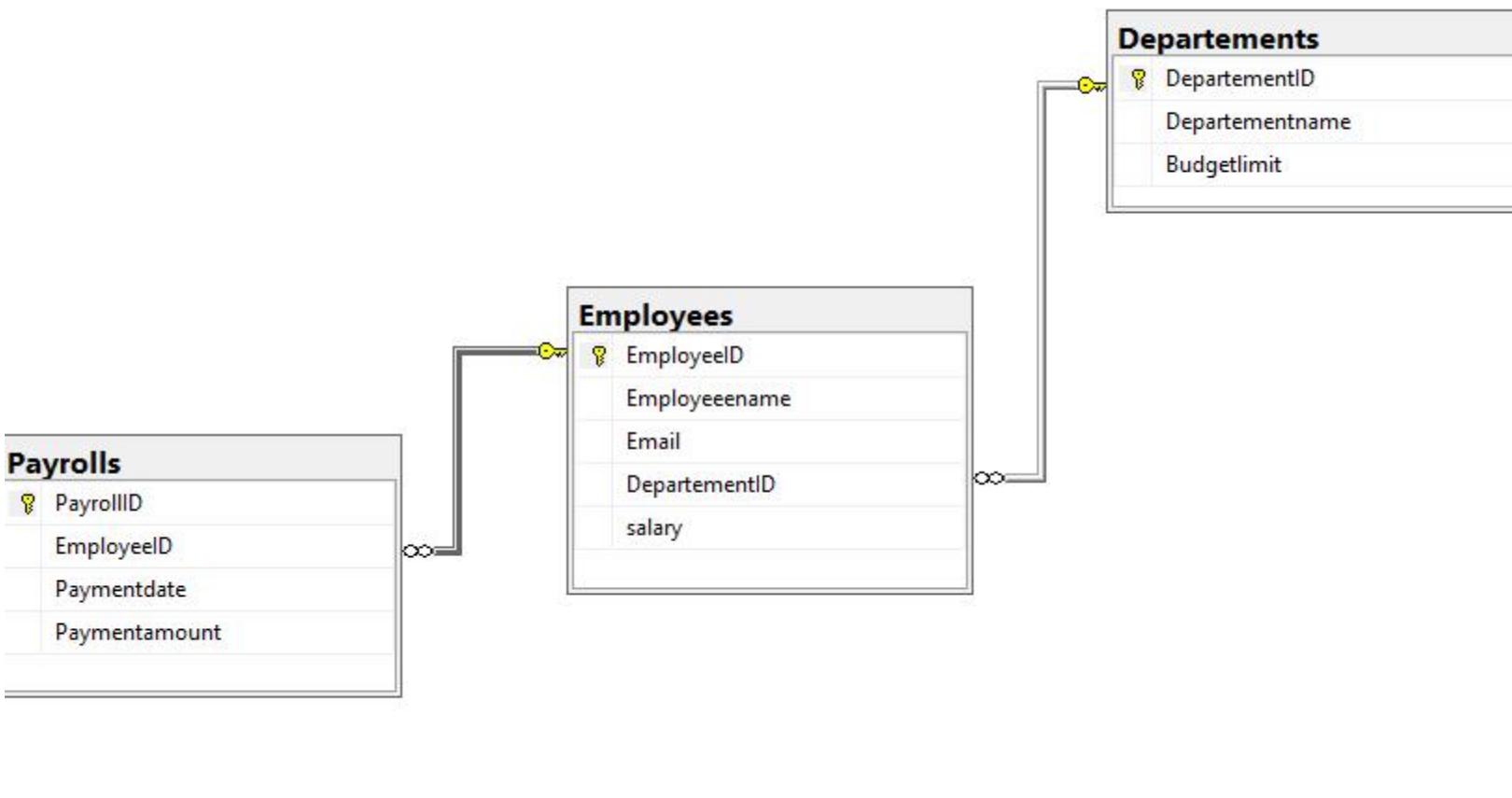
Payrolls Table

PayrollID → Primary Key (PK)

EmployeeID → Foreign Key (FK) referencing Employees(EmployeeID)

PaymentAmount → Salary payment

cont...



Key Constraints

Primary Keys (PK):

DepartmentID in Departments

EmployeeID in Employees

PayrollID in Payrolls

Foreign Keys (FK):

Employees.DepartmentID → Departments.DepartmentID

Payrolls.EmployeeID → Employees.EmployeeID