

I min databas har jag valt att göra ett HR system där en anställd kan ha en eller flera expenses (kostnads utlägg, exempelvis friskvård eller om man köpt datorutrustning som man ska få ersättning för), d.v.s 1-many relation'

En anställd kan tillhöra flera avdelningar (exempelvis en visstidsanställd som har 25% på avdelning X, 50% på avdelning Y)

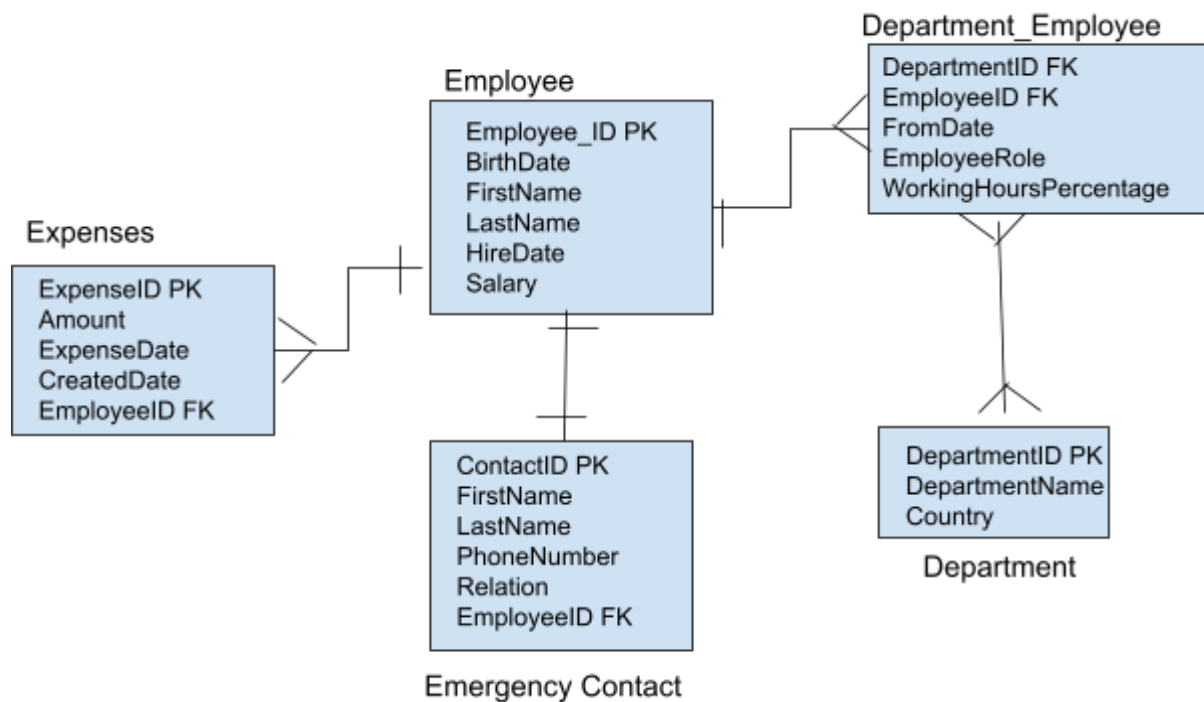
Flera avdelningar kan ha flera anställda. Many to many.

* En anställd kan endast ha en närmast anhörig (emergency contact). 1-1 relationship.

Jag har börjat med att skapa min databas, filnamn "SetupDatabase" där jag skapat mina tabeller och dess innehåll. Nedan tabell representerar min databas och de delar jag har valt att ta med.

I tabellerna har jag definierat vilka som är PrimaryKey och vad som är ForeignKey

Jag har även fått med de olika funktionerna CHAR, INT, DateTime och VARCHAR



Vidare har jag skapat "AddSampleData" där jag har börjat fylla på data i de olika tabellerna för att visa på att de olika funktionerna fungerar och att har data att kunna jobba på med vidare i uppgiften.

Mina Queries med snapshots nedan för varje rad.

1. Hämta **top** 3 anställda med flest expense reports
2. Hämta alla anställda som har varit anställda sedan 1 år tillbaka i tiden (använder funktion för **datum**)
3. Hämta alla anställda vars lön ligger över genomsnittet (**nested query**)
4. Hämta antal anställda per department (**group by**)
5. Hämta anställd och dess emergency contact - **inner join**
6. **Uppdaterat poster** (avdelning bytte namn) gjorts i filen UpdateQuery
7. **Raderat poster** (utläggsrapport var felaktig och ska tas bort) DeleteDataQuery
8. **Deklarerat poster** som **@variabler** i filen AddSampleData
9. **Skapat poster** (lagt till anställda, avdelningar, expenses etc i filen AddSampleData)

1. Top 3 expenses

```
252 -- select top 3 empoloyees with most expenses
253 SELECT * FROM (SELECT TOP 3 * FROM Expenses ORDER BY Amount DESC) AS firstThree;
```

133 %

Results Messages

	ExpenseID	ExpenseDate	CreatedDate	Amount	EmployeeID
1	3	2022-05-15	2022-06-25	3055	1
2	5	2022-03-03	2022-03-28	2500	5
3	2	2022-02-05	2022-02-12	1250	2

2.

```
258
259 SELECT *
260 FROM employees
261 WHERE HireDate <= DATEADD(year, -1, GETDATE())
262
```

132 %

Results Messages

	EmployeeID	LastName	FirstName	BirthDate	HireDate	Salary	SystemUserName
1	1	Doe	John	1980-01-01	2020-05-14	44091	Doj
2	2	Johnson	Joseph	1994-04-01	2021-05-19	23000	Jos
3	4	Svensson	Sara	1999-10-15	2020-03-20	26400	Svs
4	5	Petterson	Carl	1977-12-01	2005-09-05	47500	Pec

3.

```

267 -- Select all employees with salary higher than avg (nested query)
268 SELECT *
269 FROM Employees
270 WHERE salary > (SELECT AVG(salary) FROM Employees)
271
272

```

132 %

Results Messages

	EmployeeID	LastName	FirstName	BirthDate	HireDate	Salary	SystemUserName
1	1	Doe	John	1980-01-01	2020-05-14	44091	Doj
2	3	Anderson	Maria	1991-07-10	2022-01-23	45000	Anm
3	5	Petterson	Carl	1977-12-01	2005-09-05	47500	Pec

4.

```

277 -- Select all employees per department (group by)
278 SELECT COUNT (*) AS num_employees, Country
279 FROM Departments
280 GROUP BY Country
281

```

132 %

Results Messages

	num_employees	Country
1	1	France
2	1	Germany
3	3	Sweden

5.

```

286
287 -- Select all employees and their expenses report - inner join
288 SELECT *
289 FROM Employees
290 INNER JOIN Expenses
291 ON Employees.EmployeeID=Expenses.EmployeeID;
292

```

132 %

Results Messages

	EmployeeID	LastName	FirstName	BirthDate	HireDate	Salary	SystemUserName	ExpenseID	ExpenseDate	CreatedDate	Amount	EmployeeID
1	1	Doe	John	1980-01-01	2020-05-14	44091	Doj	1	2022-01-01	2022-01-14	255	1
2	2	Johnson	Joseph	1994-04-01	2021-05-19	23000	Jos	2	2022-02-05	2022-02-12	1250	2
3	1	Doe	John	1980-01-01	2020-05-14	44091	Doj	3	2022-05-15	2022-06-25	3055	1
4	2	Johnson	Joseph	1994-04-01	2021-05-19	23000	Jos	4	2022-07-10	2022-07-18	150	2
5	5	Petterson	Carl	1977-12-01	2005-09-05	47500	Pec	5	2022-03-03	2022-03-28	2500	5

6.

Before update

	DepartmentID	Departmentname	Country
1	1	IT	Germany
2	2	HR	Sweden
3	3	Finance	France
4	4	Certification	Sweden
5	5	HomeAppliances	Sweden

After update

```
7  -- Department HomeAppliances changed name to HomeAppliances&Tools
8  UPDATE Departments
9  SET Departmentname = 'HomeAppliances&Tools'
10 WHERE DepartmentID = 5;
11
12 SELECT * FROM Departments;
13
14
```

132 %

Results Messages

	DepartmentID	Departmentname	Country
1	1	IT	Germany
2	2	HR	Sweden
3	3	Finance	France
4	4	Certification	Sweden
5	5	HomeAppliances	Sweden

	DepartmentID	Departmentname	Country
1	1	IT	Germany
2	2	HR	Sweden
3	3	Finance	France
4	4	Certification	Sweden
5	5	HomeAppliances&Tools	Sweden

7.

Delete expense with expenseID4

Before

```
3 USE hrssystemdb -- Specifitera att jag vill köra queryn mot min databas
4
5 SELECT * FROM Expenses
6
7
```

132 %

Results Messages

	ExpenseID	ExpenseDate	CreatedDate	Amount	EmployeeID
1	1	2022-01-01	2022-01-14	255	1
2	2	2022-02-05	2022-02-12	1250	2
3	3	2022-05-15	2022-06-25	3055	1
4	4	2022-07-10	2022-07-18	150	2
5	5	2022-03-03	2022-03-28	2500	5

After removal

```
7 -- Delete expense report with ExpenseID 4 that was wrongly added
8 DELETE FROM Expenses
9 WHERE ExpenseID=4;
10
11
```

132 %

Results Messages

	ExpenseID	ExpenseDate	CreatedDate	Amount	EmployeeID
1	1	2022-01-01	2022-01-14	255	1
2	2	2022-02-05	2022-02-12	1250	2
3	3	2022-05-15	2022-06-25	3055	1
4	5	2022-03-03	2022-03-28	2500	5