

Vállalat

Adatbázisok kötelező feladat

Egy vállalati nyilvántartó rendszerben tárolják a cég dolgozóinak, osztályainak, részlegeinek és projektjeinek adatait. Az új dolgozóknak regisztrálniuk kell a rendszerbe, majd bejelentkezés után használhatják azt. Az adminok aktualizálhatják az adatokat, a többi dolgozó csak megtekintheti azokat, és beszámolót írhat azon projektekhez, amelyekben részt vesz vagy vett.

Egyed-kapcsolat modell

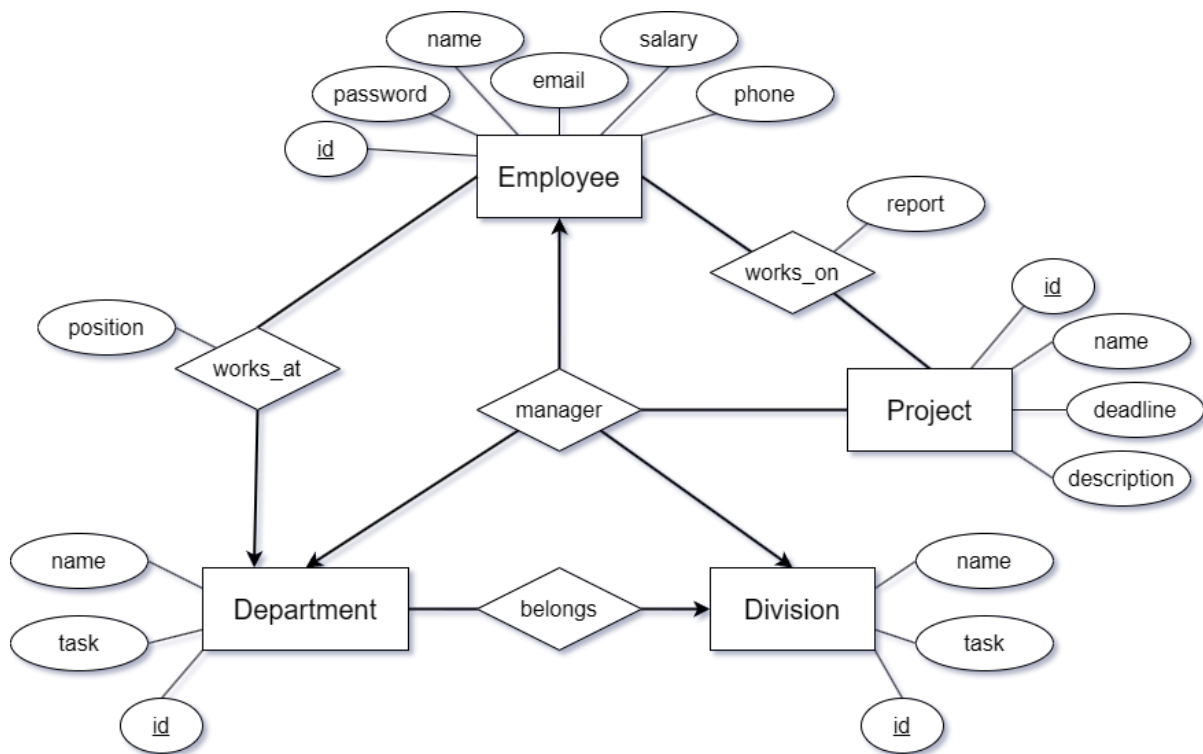


Figure 1: E-K diagram

Az **Employee** egyednek az **id** (céges azonosító) lesz a kulcs attribútuma. A **Department** (részleg), **Division** (osztály) és **Project** egyedeknél a **name** attribútum lehetne kulcs, viszont el szeretném kerülni a string kulcsokat, így bevezetek hozzájuk egy **id** attribútumot, ami kulcs attribútum lesz. Mindegyiküket a **manager** kapcsolat összeköti az **Employee** egyeddel. Mind a **Project**-nek, **Department**-nek és **Division**-nek csak egy darab **manager**-e lehet. egy **Employee** csak egy **Division**-nek, és csak egy **Department**-nek lehet a **manager**-e, míg **Project**-ek közt többnek is. Tehát a **Division** és **Employee** között, illetve **Department** és **Employee** között a **manager** kapcsolat 1:1-hez, a **Project** és **Employee** között a **manager** kapcsolat pedig 1:N-hez.

Az **Employee** továbbá rendelkezik egy **works_on** kapcsolattal a **Project** egyeddel, illetve egy **works_at** kapcsolattal a **Department** egyeddel. A **works_on** egy N:M-hez kapcsolat, mivel egy **Employee** több **Project**-en is dolgozhat, illetve egy **Project**-en több **Employee** is dolgozhat. Emellett rendelkezik egy **report** (beszámoló) attribútummal. A **works_at** egy 1:N-hez kapcsolat, mivel egy **Employee** csak egy **Department**-ben dolgozhat, viszont egy **Department**-ben több **Employee** is dolgozhat. Emellett rendelkezik egy **position** (beosztás) attribútummal.

A **Department** és a **Division** között van egy 1:N hez **belongs** kapcsolat. Egy **Department** csak egy **Division**-be tartozhat, viszont egy **Division**-be több **Department**-is tartozhat.

Relációs adatbázisséma

EMPLOYEES(employee_id, password, name, email, salary, phone)

DEPARTMENTS(dep_id, dep_name, task, manager_id, div_id)

DIVISIONS(div_id, div_name, task, manager_id)

PROJECTS(project_id, project_name, deadline, description, manager_id)

WORKS_AT(employee_id, dep_id, position)

WORKS_ON(employee_id, project_id, report)

Normalizálás

Az 1NF teljesül minden táblára, mivel mindegyik táblában mindegyik attribútum atomi.

A 2NF teljesül az EMPLOYEES, DEPARTMENTS, DIVISIONS, PROJECTS és WORKS_AT táblákra, mivel egy elemű a kulcsuk. A 2NF teljesül a WORKS_ON táblára, mert az egyedüli másodlagos attribútum (**report**) teljesen függ mindkét kulcstól.

A 3NF teljesül az EMPLOYEES táblára, mivel az egyedüli attribútum, amin keresztül tranzitív függőség léphet fel az az **email**, viszont az **email** feltételezhetően egyedi, így az kulcs is lehetne. Ebből következik, hogy az **email** → **employee_id** függés fenn áll. A 3NF teljesül a DEPARTMENTS, DIVISIONS és PROJECTS táblára. Hasonlóan az EMPLOYEE táblához, ezeknél csak a ***_name** attribútumon keresztül vezethet tranzitív függés, viszont a **dep_name** → **department_id**, **div_name** → **division_id** és **project_name** → **project_id** függések fennállnak, mivel a ***_name** attribútum egyedi mindegyik esetben. A WORKS_AT tábla is 3NF-ben van, mivel mindkét másodlagos attribútuma csak az elsődleges kulcstól függ. A WORKS_ON tábla is 3NF-ben van, mivel csak egy darab másodlagos attribútuma van, így nem jöhet létre tranzitív függés másodlagos attribútumok között.

Táblatervek

EMPLOYEES

employee_id	INT	A dolgozó céges azonosítója, kulcs
password	VARCHAR(128)	A dolgozó belépési jelszava (hashelve, és szóva)
name	VARCHAR(100)	A dolgozó teljes neve
email	VARCHAR(100)	A dolgozó munkahelyi email címe
salary	INT	A dolgozó bére
phone	BIGINT	A dolgozó telefonszáma

DEPARTMENTS

dep_id	INT	A részleg azonosítója, kulcs
--------	-----	-------------------------------------

DEPARTMENTS

dep_name	VARCHAR(100)	A részleg neve
task	TEXT	A részleg feladata
manager_id	INT	A részleg vezetője. Külső kulcs az EMPLOYEES tábla kulcsára
div_id	INT	A részleg osztálya. Külső kulcs a DIVISIONS tábla kulcsára

DIVISIONS

div_id	INT	Az osztály azonosítója, kulcs
div_name	VARCHAR(100)	Az osztály neve
task	TEXT	Az osztály feladata
manager_id	INT	Az osztály vezetője. Külső kulcs az EMPLOYEES tábla kulcsára

PROJECTS

project_id	INT	A projekt azonosítója, kulcs
project_name	VARCHAR(100)	A projekt neve
deadline	DATETIME	A projekt határideje, napra és órára pontosan
description	TEXT	A projekt leírása
manager_id	INT	A projekt vezetője. Külső kulcs az EMPLOYEES tábla kulcsára

WORKS_AT

employee_id	INT	Külső kulcs az EMPLOYEES tábla kulcsára, kulcs
dep_id	INT	Külső kulcs a DEPARTMENTS tábla kulcsára
position	VARCHAR(40)	A dolgozó beosztása

WORKS_ON

employee_id	INT	Külső kulcs az EMPLOYEES tábla kulcsára, kulcs
project_id	INT	Külső kulcs a PROJECTS tábla kulcsára, kulcs
report	TEXT	A dolgozó beszámolója a projektről

Összetett lekérdezés

```
SELECT name, project_name, report
FROM EMPLOYEES, PROJECTS, WORKS_ON
WHERE EMPLOYEES.employee_id = WORKS_ON.employee_id AND
      PROJECTS.project_id = WORKS_ON.project_id
ORDER BY name ASC;
```

```
SELECT name, div_name, dep_name, salary
FROM EMPLOYEES, DIVISIONS, DEPARTMENTS, WORKS_AT
WHERE EMPLOYEES.employee_id = WORKS_AT.employee_id AND
      WORKS_AT.dep_id = DEPARTMENTS.dep_id AND
      DIVISIONS.div_id = DEPARTMENTS.div_id AND
      EMPLOYEES.salary <
      (SELECT AVG(salary)
       FROM EMPLOYEES, DEPARTMENTS, DIVISIONS, WORKS_AT
```

```
WHERE EMPLOYEES.employee_id = WORKS_AT.employee_id AND
      WORKS_AT.dep_id = DEPARTMENTS.dep_id AND
      DEPARTMENTS.div_id = 1)
ORDER BY salary ASC;
```

```
SELECT EMPLOYEES.name AS manager,
       CONCAT("Division: " ,DIVISIONS.div_name) AS manages
FROM EMPLOYEES, DIVISIONS
WHERE EMPLOYEES.employee_id = DIVISIONS.manager_id
UNION
SELECT EMPLOYEES.name AS managers,
       CONCAT("Department: " ,DEPARTMENTS.dep_name) AS manages
FROM EMPLOYEES, DEPARTMENTS
WHERE EMPLOYEES.employee_id = DEPARTMENTS.manager_id
UNION
SELECT EMPLOYEES.name AS managers,
       CONCAT("Project: " ,PROJECTS.project_name) AS manages
FROM EMPLOYEES, PROJECTS
WHERE EMPLOYEES.employee_id = PROJECTS.manager_id
ORDER BY manager, manages;
```

Megvalósítás, funkciók

Könnyített teljesítés esetén ilyen nincs.