DBM 1

Christian Hohlmann Enevoldsen, MRB852

7. marts 2014

1 From ER to Relational

1.
Translation form ER to relational Schema

Stakeholders(CVR, TEL, URL, Text)
Companies(*CVR*)
Owns(*ownerCVR*, *CVR*)
Municipalities(*CVR*, Budget)
Peope(Name, Email, TEL)
WorkIn(*CVR*, *PersonName*)
Project(PID, Budget, URL)
Skills(Name)
NeedSkills(*SkillName*, *PID*, PeopleCount)
HaveSkills(*SkillName*, *PeopleName*)
Sponsors(*PID*, *CVR*)
Participate(*StakeholderCvr*, *PID*, Role)
Allocate(*From*, *To*, *Percentage*, *PersonName*, *PID*)

Den mest væsentlige ændring jeg har lavet i oversættelsen er at jeg har lavet statementet en attribute, da jeg ikke synes at en weak relation var nødvendig. Statementet bliver alligevel fjernet med stakeholderen. På samme måde kunne man bruge mapping oversættelse på det hele for at optimere databasen. Jeg har også valgt at fjerne periods, da allocate alligevel bruger de samme værdier som keys.

2.

```
CREATE TABLE Stakeholders (
CVR CHAR(8) PRIMARY KEY,
TEL CHAR(8) UNIQUE,
URL VARCHAR(255) UNIQUE,
Text VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE Companies (
CVR CHAR(8) PRIMARY KEY REFERENCES Stakeholders(CVR)
);

CREATE TABLE Owns (
OwnerCVR CHAR(8) REFERENCES Stakeholders(CVR),
CVR CHAR(8) REFERENCES Stakeholders(CVR),
PRIMARY KEY (OwnerCVR, CVR)
);

CREATE TABLE Municipalities (
CVR CHAR(8) PRIMARY KEY REFERENCES Stakeholders(CVR),
Budget REAL
);

CREATE TABLE People (
Name VARCHAR(35) PRIMARY KEY,
Email VARCHAR(255) UNIQUE,
TEL VARCHAR(8) UNIQUE
);

CREATE TABLE WorkIn (
CVR CHAR(8) REFERENCES Stakeholders(CVR) NOT NULL,
PersonName VARCHAR(35) REFERENCES People(Name) NOT NULL
);

CREATE TABLE Project (
PID INT PRIMARY KEY,
Budget REAL,
URL VARCHAR(255) UNIQUE
);

CREATE TABLE Skills (
Name VARCHAR(50) PRIMARY KEY
);

CREATE TABLE NeedSkills (
SkillName VARCHAR(50) REFERENCES Skills(Name),
```

```
PID INT REFERENCES Project(PID),
RequiredPeople INT,
PRIMARY KEY (SkillName, PID)
);

CREATE TABLE HaveSkills (
SkillName VARCHAR(50) REFERENCES Skills(Name),
PID INT REFERENCES Project(PID),
PRIMARY KEY (SkillName, PID)
);

CREATE TABLE Sponsors (
PID INT PRIMARY KEY REFERENCES Project(PID),
CVR CHAR(8) REFERENCES Stakeholders(CVR) NOT NULL
);

CREATE TABLE Participate (
CVR CHAR(8) REFERENCES Stakeholders(CVR),
PID INT REFERENCES Project(PID),
Role VARCHAR(255),
PRIMARY KEY (CVR, PID)
);

CREATE TABLE Allocate (
FromDate DATE,
ToDate DATE,
Percentage REAL,
PersonName VARCHAR(35) REFERENCES People(Name),
PID INT REFERENCES Project(PID),
PRIMARY KEY (FromDate, ToDate, Percentage, PersonName, PID)
);
```

3.

```
INSERT INTO Stakeholders (CVR, TEXT) VALUES ('21212121', 'Some text');

INSERT INTO Companies (CVR) VALUES ('21212121');
```

```
INSERT INTO Stakeholders (CVR, TEXT) VALUES ('42424242', 'Another
text');

INSERT INTO Companies (CVR) VALUES ('42424242');

INSERT INTO Owns(OwnerCVR, CVR) VALUES ((SELECT CVR FROM
Companies WHERE Companies.CVR = '42424242'
),
(SELECT CVR FROM Companies WHERE Companies.CVR = '21212121
'));

INSERT INTO Project(PID, URL, Budget) VALUES (42, 'http://42.com',
100000);

INSERT INTO Sponsors(PID, CVR) VALUES ((SELECT PID FROM Pro-
ject WHERE Project.PID = 42),
(SELECT CVR FROM Companies WHERE Companies.CVR = '21212121'
));

INSERT INTO Skills (name) VALUES ('Computer Programming');

INSERT INTO NeedSkills(SkillName, PID, RequiredPeople) VALUES (
(SELECT name FROM Skills WHERE Skills.name = 'Computer Program-
ming'),
(SELECT PID FROM Project WHERE Project.PID = 42),
42
);
```

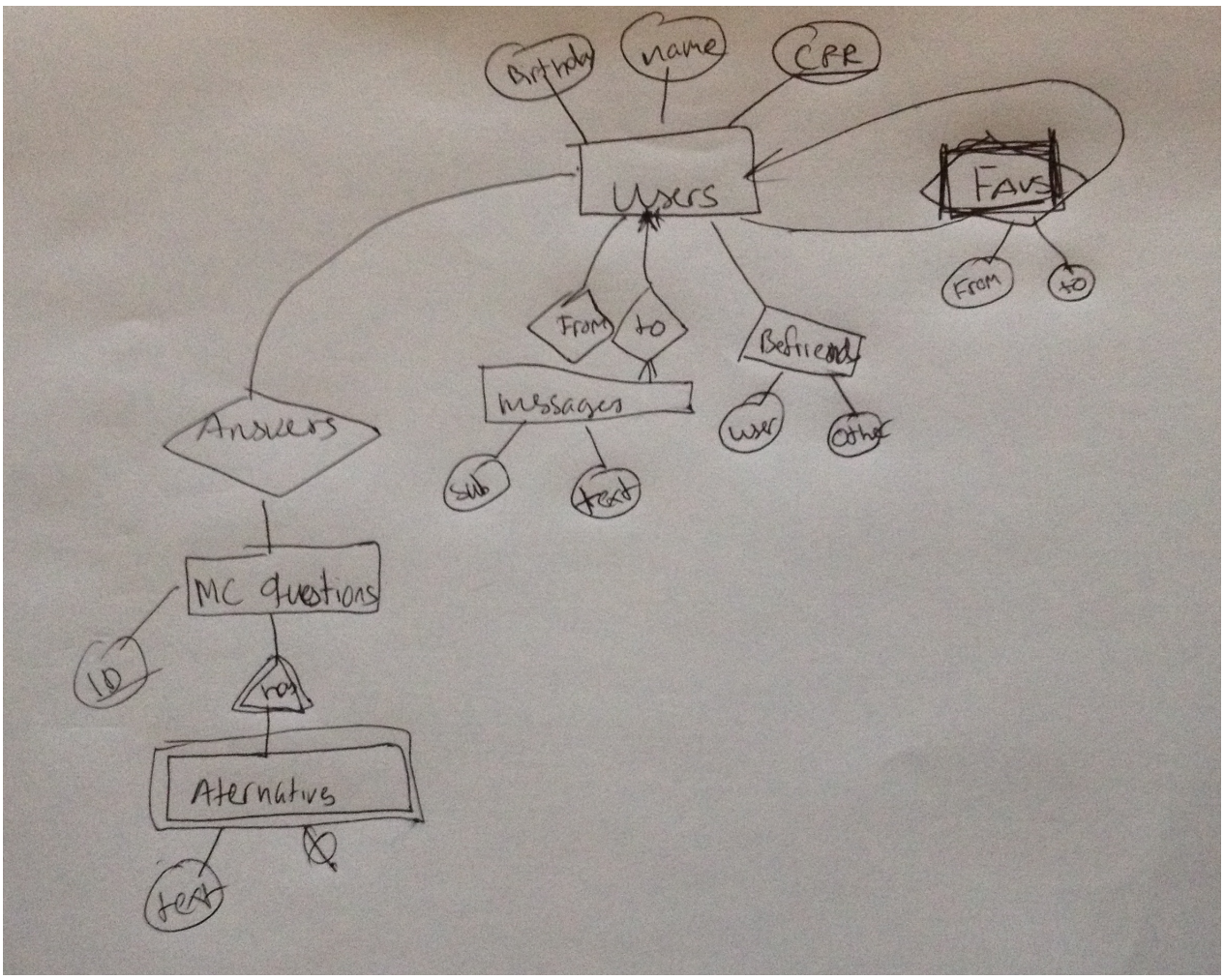
4.

```
UPDATE Project
SET Budget=420000
WHERE Project.Budget = 100000;
```

2 Relational Algebra

1. $\pi_{names}(\sigma_{type=enzymes})(Proteins)$
2. $\pi_{Proteins.name}(\sigma_{name=TP53}(Gene) \times (Influences \times Protein))$
3. $\pi_{((Proteins.id,type=enzyme)\times(Infuences\times Gene\times(\sigma_{type=membrane}(Protein))))}$

3 ER Modelling



Jeg har valgt at lave alternatives weak da der ikke er nogen primary key og uden questionarries vil det ikke give nogen mening at have dem.

4 Functional Dependencies, Schema Design, and Normal Forms

5 SQL

1.
SELECT names FROM proteins WHERE type=fibrous
2.
SELECT g.name
FROM Genes AS g
LEFT JOIN Influences AS i ON g.id = i.gid
RIGHT JOIN Protein AS p ON p.id = i.pid
WHERE p.type = 'membrane' AND p.type 'fibrous'

3.
SELECT *
FROM Proteins AS p
LEFT JOIN Influences AS i ON
p.id = i.pid
RIGHT JOIN Genes AS g ON g.id = i.gid
WHERE g.name IS 'TP53' AND g.name IS NOT 'BCO2'

4.
SELECT *
FROM proteins AS p
LEFT JOIN influences AS i ON p.id = i.pid
RIGHT JOIN genes AS g ON g.id = i.gid
HAVING COUNT(i.pid) > 42;