Notes on the lectures 9 and 10

During the class we have normalized the table below (let's call it EmpProj) using the following assumptions: Different people might have the same name, the department and project names are unique.

eid	lname	fname	did	dept	pid	project	hours
1	Alnes	Bernt	1	Planning	1	Alfa	100
2	Fjelldal	Mads	1	Planning	3	Charlie	140
4	Longva	Victor	1	Planning	2	Bravo	80
7	Bakke	Alfred	2	Production A	1	Alfa	80
8	Vie	Tor	2	Production A	1	Alfa	910
8	Vie	Tor	2	Production A	3	Charlie	720
9	Westgaard	Sten	2	Production A	3	Charlie	310
10	Liseth	Rakel	3	Production B	3	Charlie	480
11	Norman	Emil	3	Production B	3	Charlie	460
12	Dyrhaug	Atle	3	Production B	2	Bravo	810
15	Kvien	Amalie	4	Sales	2	Bravo	15
15	Kvien	Amalie	4	Sales	3	Charlie	75
16	Tveten	Thomas	4	Sales	2	Bravo	20
17	Lende	Marita	4	Sales	1	Alfa	40

EmpProj(eid, lname, fname, did, dept, pid, project, hours)
Candidate keys (CKs): {eid, pid} and {eid, project}
Prime attributes (i. e. members of any candidate key): eid, pid, project
Non-prime attributes: lname, fname, did, dept, hours

The completely non-trivial FFDs:

1. {eid, pid} → {lname, fname, did, dept, project, hours}
2. {eid, project} → {lname, fname, did, dept, pid, hours}
3. {eid} → {lname, fname, did, dept}
4. {pid} → {project}
5. {project} → {pid}
6. {did} → {dept}
7. {dept} → {did}

This table is in 1NF but it is not in 2NF since the third FFD has a determinant which is a proper subset of a candidate key ({eid} is a proper subset of both {eid, pid} and {eid, project}). We will create a new table, Employee, which will contain the eid and all attributes which are determined by eid (i. e. $\{\text{eid}\}^+$), using eid as the primary key:

Employee(<u>eid</u>, lname, fname, did, dept)

We can now copy all distinct eid, lname, fname, did, dept to this table and make eid in the original table, EmpProj, a foreign key (FK) referencing eid in the new table, Employee.

After this is done we can remove lname, fname, did and dept from the EmpProj:

EmpProj(*eid*, pid, project, hours)

The table still has the same CKs. The prime attributes are: eid, pid, project and the non-prime attributes are: hours.

What are the completely non-trivial FFDs now?

1. {eid, pid} → {project, hours}
2. {eid, project} → {pid, hours}
3. {pid} → {project}
4. {project} → {pid}

There is only one non-prime attribute, hours, and it depends on the whole candidate keys (not on a proper subset of any candidate keys), and only on the candidate keys! The table is now 2NF as well as in 3NF (because it does not depend on anything other than candidate keys).

Let's have a look at the Employee table. There is only one CK: {eid}. The prime attributes are eid and the non-prime attributes are lname, fname, did and dept. The table has following completely non-trivial FFDs:

1. {eid} → {lname, fname, did, dept}
2. {did} → {dept}
3. {dept} → {did}

The table is in 2NF (candidate keys with only one attribute cannot have any proper subset so the table must be in 2NF), but it is not in 3NF since e. g. dept depends on did which is not a CK. How can we fix this? We will create yet another table, Department, which will contain did and dept, using did as the primary key (we could also choose dept because of the third FFD, but we prefer to have PK which takes as little storage space as possible):

Department(<u>did</u>, dept)

Again, we will copy all distinct did, dept from the Employee table to this table and make did in the original table, Employee, a FK referencing did in the new table. Afterwards we can delete dept from the Employee table:

Employee(<u>eid</u>, lname, fname, *did*)

If we again list the FFDs and analyze them, we will conclude that the table is in 3NF.

To summarize, we now have the following tables (all in 3NF):

EmpProj(*eid*, pid, project, hours)
CK: {eid, pid}, {eid, project}
Prime: eid, pid, project
Non-prime: hours

- {eid, pid} → {project, hours}
- {eid, project} → {pid, hours}
- {pid} → {project}
- {project} → {pid}

Employee(eid, lname, fname, *did*)
CK: {eid}
Prime: eid
Non-prime: lname, fname, did

- {eid} → {lname, fname, did, dept}

Department(did, dept)
CK: {did}, {dept}
Prime: did, dept
Non-prime: none

- {did} → {dept}
- {dept} → {did}

Employee and Department are in BCNF as well, since each determinants (the left sides of their non-trivial FFDs) is a candidate key.

The EmpProj is however not, since neither {pid} nor {project} is a candidate key. We can fix this by creating a table for projects with columns pid and project, using e. g. pid as the primary key:

Project(pid, project)

As before we copy distinct pid, proj to this table, add a FK, and remove the project column from the EmpProj table. We will also fix the PK for this table:

EmpProj(*eid*, *pid*, hours)

It is easy to check that both this table and Project are now in BCNF.

The final decomposition

The final decomposition consists of four tables (all in BCNF):

EmpProj(*eid*, *pid*, hours)
CK: {eid, pid}
Prime: eid, pid
Non-prime: hours

- {eid, pid} → {hours}

Employee(eid, lname, fname, *did*)
CK: {eid}
Prime: eid
Non-prime: lname, fname, did

- {eid} → {lname, fname, did, dept}

Department(did, dept)
CK: {did}, {dept}
Prime: did, dept
Non-prime: none

- {did} → {dept}
- {dept} → {did}

Project(pid, project)
CK: {pid}, {project}
Prime: pid, project
Non-prime: none

- {pid} → {project}
- {project} → {pid}

Using SQL to normalize tables

```
select * from empproj;
```

```
+-----+-----------+--------+-----+--------------+-----+---------+-------+
| eid | lname     | fname  | did | dept         | pid | project | hours |
+-----+-----------+--------+-----+--------------+-----+---------+-------+
|   1 | Alnes     | Bernt  |   1 | Planning     |   1 | Alfa    |   100 |
|   2 | Fjelldal  | Mads   |   1 | Planning     |   3 | Charlie |   140 |
|   4 | Longva    | Victor |   1 | Planning     |   2 | Bravo   |    80 |
|   7 | Bakke     | Alfred |   2 | Production A |   1 | Alfa    |    80 |
|   8 | Vie       | Tor    |   2 | Production A |   1 | Alfa    |   910 |
|   8 | Vie       | Tor    |   2 | Production A |   3 | Charlie |   720 |
|   9 | Westgaard | Sten   |   2 | Production A |   3 | Charlie |   310 |
|  10 | Liseth    | Rakel  |   3 | Production B |   3 | Charlie |   480 |
|  11 | Norman    | Emil   |   3 | Production B |   3 | Charlie |   460 |
|  12 | Dyrhaug   | Atle   |   3 | Production B |   2 | Bravo   |   810 |
|  15 | Kvien     | Amalie |   4 | Sales        |   2 | Bravo   |    15 |
|  15 | Kvien     | Amalie |   4 | Sales        |   3 | Charlie |    75 |
|  16 | Tveten    | Thomas |   4 | Sales        |   2 | Bravo   |    20 |
|  17 | Lende     | Marita |   4 | Sales        |   1 | Alfa    |    40 |
+-----+-----------+--------+-----+--------------+-----+---------+-------+
```

```
create table employee as select distinct eid, lname, fname, did, dept from empproj;
alter table employee add primary key (eid);
alter table empproj add foreign key (eid) references employee(eid);
alter table empproj drop column lname;
alter table empproj drop column fname;
alter table empproj drop column did;
alter table empproj drop column dept;
select * from empproj;
```

```
+-----+-----+---------+-------+
| eid | pid | project | hours |
+-----+-----+---------+-------+
|   1 |   1 | Alfa    |   100 |
|   2 |   3 | Charlie |   140 |
|   4 |   2 | Bravo   |    80 |
|   7 |   1 | Alfa    |    80 |
|   8 |   1 | Alfa    |   910 |
|   8 |   3 | Charlie |   720 |
|   9 |   3 | Charlie |   310 |
|  10 |   3 | Charlie |   480 |
|  11 |   3 | Charlie |   460 |
|  12 |   2 | Bravo   |   810 |
|  15 |   2 | Bravo   |    15 |
|  15 |   3 | Charlie |    75 |
|  16 |   2 | Bravo   |    20 |
|  17 |   1 | Alfa    |    40 |
+-----+-----+---------+-------+
select * from employee;
```

```
+-----+-----------+--------+-----+--------------+
| eid | lname     | fname  | did | dept         |
+-----+-----------+--------+-----+--------------+
|   1 | Alnes     | Bernt  |   1 | Planning     |
|   2 | Fjelldal  | Mads   |   1 | Planning     |
|   4 | Longva    | Victor |   1 | Planning     |
|   7 | Bakke     | Alfred |   2 | Production A |
|   8 | Vie       | Tor    |   2 | Production A |
|   9 | Westgaard | Sten   |   2 | Production A |
|  10 | Liseth    | Rakel  |   3 | Production B |
|  11 | Norman    | Emil   |   3 | Production B |
|  12 | Dyrhaug   | Atle   |   3 | Production B |
|  15 | Kvien     | Amalie |   4 | Sales        |
|  16 | Tveten    | Thomas |   4 | Sales        |
|  17 | Lende     | Marita |   4 | Sales        |
+-----+-----------+--------+-----+--------------+
```

```
create table department as select distinct did, dept from employee;
alter table department add primary key (did);
alter table employee add foreign key (did) references department(did);
alter table employee drop column dept;
select * from employee;
```

```
+-----+-----------+--------+-----+
| eid | lname     | fname  | did |
+-----+-----------+--------+-----+
|   1 | Alnes     | Bernt  |   1 |
|   2 | Fjelldal  | Mads   |   1 |
|   4 | Longva    | Victor |   1 |
|   7 | Bakke     | Alfred |   2 |
|   8 | Vie       | Tor    |   2 |
|   9 | Westgaard | Sten   |   2 |
|  10 | Liseth    | Rakel  |   3 |
|  11 | Norman    | Emil   |   3 |
|  12 | Dyrhaug   | Atle   |   3 |
|  15 | Kvien     | Amalie |   4 |
|  16 | Tveten    | Thomas |   4 |
|  17 | Lende     | Marita |   4 |
+-----+-----------+--------+-----+
```

```
select * from department;
```

```
+-----+--------------+
| did | dept         |
+-----+--------------+
|   1 | Planning     |
|   2 | Production A |
|   3 | Production B |
|   4 | Sales        |
+-----+--------------+
```

```
create table project as select distinct pid, project from empproj;
alter table project add primary key (pid);
alter table empproj add foreign key (pid) references project(pid);
alter table empproj drop column project;
alter table empproj add primary key(eid, pid);
select * from empproj;
```

```
+-----+-----+-------+
| eid | pid | hours |
+-----+-----+-------+
|   1 |   1 |   100 |
|   2 |   3 |   140 |
|   4 |   2 |    80 |
|   7 |   1 |    80 |
|   8 |   1 |   910 |
|   8 |   3 |   720 |
|   9 |   3 |   310 |
|  10 |   3 |   480 |
|  11 |   3 |   460 |
|  12 |   2 |   810 |
|  15 |   2 |    15 |
|  15 |   3 |    75 |
|  16 |   2 |    20 |
|  17 |   1 |    40 |
+-----+-----+-------+
```

```
select * from project;
```

```
+-----+---------+
| pid | project |
+-----+---------+
|   1 | Alfa    |
|   2 | Bravo   |
|   3 | Charlie |
+-----+---------+
```

The final state

```
select * from empproj;
```

```
+-----+-----+-------+
| eid | pid | hours |
+-----+-----+-------+
|   1 |   1 |   100 |
|   2 |   3 |   140 |
|   4 |   2 |    80 |
|   7 |   1 |    80 |
|   8 |   1 |   910 |
|   8 |   3 |   720 |
|   9 |   3 |   310 |
|  10 |   3 |   480 |
|  11 |   3 |   460 |
|  12 |   2 |   810 |
|  15 |   2 |    15 |
|  15 |   3 |    75 |
|  16 |   2 |    20 |
|  17 |   1 |    40 |
+-----+-----+-------+
```

```
select * from employee;
```

```
+-----+-----------+--------+-----+
| eid | lname     | fname  | did |
+-----+-----------+--------+-----+
|   1 | Alnes     | Bernt  |   1 |
|   2 | Fjelldal  | Mads   |   1 |
|   4 | Longva    | Victor |   1 |
|   7 | Bakke     | Alfred |   2 |
|   8 | Vie       | Tor    |   2 |
|   9 | Westgaard | Sten   |   2 |
|  10 | Liseth    | Rakel  |   3 |
|  11 | Norman    | Emil   |   3 |
|  12 | Dyrhaug   | Atle   |   3 |
|  15 | Kvien     | Amalie |   4 |
|  16 | Tveten    | Thomas |   4 |
|  17 | Lende     | Marita |   4 |
+-----+-----------+--------+-----+
```

```sql
select * from department;
```

```
+-----+--------------+
| did | dept         |
+-----+--------------+
|   1 | Planning     |
|   2 | Production A |
|   3 | Production B |
|   4 | Sales        |
+-----+--------------+
```

```sql
select * from project;
```

```
+-----+---------+
| pid | project |
+-----+---------+
|   1 | Alfa    |
|   2 | Bravo   |
|   3 | Charlie |
+-----+---------+
```