
# Hópverkefni 5

**Eva Sól Pétursdóttir og Halla Margrét Jónsdóttir**
B.Sc. Software Engineering

Gagnasafnsfræði
T-202-GAG1
Anna Sigríður Islind

Spring 2021
Reykjavík University - School of Computer Science

31. mars 2021

**Table of Contents**


**1 Introduction** **3**

**2 Checking for FDs and MVDs** **3**

2.1 Checking for FDs 3

2.2 Checking for MVDs 3

**3 Normalization of Civil services** **4**

3.1 Analysis 4

3.2 Table: CivilServices_CSID_HID_S 5

3.3 Table: CivilServices_CSID_PN 5

3.4 Table: CivilServices_HID_HS_HZ 5

3.5 Table: CivilServices_HZ_HC 5

**4 Normalization of Projects** **5**

4.1 Analysis 5

4.2 Table: Projects_ID_PID_SID 6

4.3 Table: Projects_ID_MID 6

4.4 Table: Projects_MID_MN 6

4.5 Table: Projects_PID_PN 7

4.6 Table: Projects_SID_SN 7

**5 Normalization of Citizens** **7**

5.1 Analysis 7

5.2 Table: Citizens_CID_CN_CS_CNr_CZ_EID 8

5.3 Table: Citizens_CZ_CL 8

**6 Normalization of Coffees** **9**

6.1 Analysis 9

6.2 Table: Coffees_CID_DID_HID 9

6.2.1 Table: Coffees_DID_CID 9

6.2.2 Table: Coffees_DID_HID 10

6.3 Table: Coffees_DID_DN_DS 10

6.4 Table: Coffees_CID_CN_CC 10

**7 Epilogue** **11**

# 1 Introduction

This is the 5th and final project in T-202-GAG1 Databases. Stella, the new president of Iceland needs help to sort out a big mess in the database from the old empire. The database has only 4 tables. Stella wants them on the highest normal form that is possible, and all the existing dependencies need to be preserved.

Two scripts were run on each relation, one to check for FDs and other to check for MVDs. Then we went through the normalization process for each of the relations and decomposed them as needed. This report contains documentation of each step of the normalization process.

# 2 Checking for FDs and MVDs

We used the scripts that were created in dæmatímaverkefni 10 with the following templates and used the results in the process of decomposition.

## 2.1 Checking for FDs

```
SELECT '{R}: {A} --> {B}' AS FD,
   CASE WHEN COUNT(*) = 0 THEN
       'HOLDS'
   ELSE
       'does not hold'
   END AS VALIDITY
FROM (
   SELECT {A}
   FROM {R}
   GROUP BY {A}
   HAVING COUNT(DISTINCT {B}) > 1
) X;
```

The program found all possible combinations of A and B in every relation R.

## 2.2 Checking for MVDs

```
SELECT
   '{A} ->> ({B}, {D}) in {R}' AS Relation,
   CASE WHEN COUNT(*) = 0 THEN
       'MAYBE MVD'
```

```
   ELSE
       'NO MVD'
   END AS MVD
FROM (
   SELECT {A}
   FROM {R}
   GROUP BY {A}
   HAVING COUNT(*) > 1
       AND COUNT(*) <> COUNT(DISTINCT {B}) * COUNT(DISTINCT {D})
) X;
```

The program found all possible combinations of the primary keys A, B and C in every relation R that had a primary key with three or more columns.

# 3 Normalization of Civil services

## 3.1 Analysis

Primary key: (CSID, HID)
Determined FDs:
- CSID → PN
- HID → HS
- HID → HZ
- HID → HC
- HZ → HC

Other keys: None
Minimal cover:
- (CSID, HID) → S
- CSID → PN
- HID → HS
- HID → HZ
- HZ → HC

Normal form: 1NF

Decomposition:
- CivilServices_CSID_HID_S
- CivilServices_CSID_PN
- CivilServices_HID_HS_HZ
- CivilServices_HZ_HC

## 3.2 Table: CivilServices_CSID_HID_S

Columns: CSID, HID, S
Key: (CSID, HID)
FDs: (CSID, HID) → S
Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only two columns and therefore in 4NF

## 3.3 Table: CivilServices_CSID_PN

Columns: CSID, PN
Key: CSID
FDs: CSID → PN
Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

## 3.4 Table: CivilServices_HID_HS_HZ

Columns: HID, HS, HZ
Key: HID
FDs:

HID → HS
HID → HZ

Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

## 3.5 Table: CivilServices_HZ_HC

Columns: HZ, HC
Key: HZ
FDs: HZ → HC
Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

# 4 Normalization of Projects

## 4.1 Analysis

Primary key: (ID, PID, SID)

Determined FDs:

ID → MID
ID → MN
PID → PN
SID → SN
MID → MN

Other keys: None
Minimal cover:

(ID, PID, SID) → (ID, PID, SID)
ID → MID
MID → MN
PID → PN
SID → SN

Normal form: 1NF
Decomposition:

Projects_ID_PID_SID
Projects_ID_MID
Projects_MID_MN
Projects_PID_PN
Projects_SID_SN

## 4.2 Table: Projects_ID_PID_SID

Columns: ID, PID, SID
Key: (ID, PID, SID)
Other FDs: None
Normal forms:

- All the columns are part of the primary key and therefore the table is in BCNF
- When we ran the script to check for MVDs the result said “NO MVD” so the table is on 4NF.

## 4.3 Table: Projects_ID_MID

Columns: ID, MID
Key: ID
FDs: ID → MID
Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

## 4.4 Table: Projects_MID_MN

Columns: MID, MN

Key: MID
FDs: MID → MN
Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

## 4.5 Table: Projects_PID_PN

Columns: PID, PN
Key: PID
FDs: PID → PN
Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

## 4.6 Table: Projects_SID_SN

Columns: SID, SN
Key: SID
FDs: SID → SN
Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

# 5 Normalization of Citizens

## 5.1 Analysis

Primary key: CID
Determined FDs:

CID → CN
CID → CS
CID → CNr

CID → CZ
CID → CL
CID → EID
CZ → CL

Other keys: None

Minimal cover:

CID → CN
CID → CS
CID → CNr
CID → CZ
CID → EID
CZ → CL

Normal form: 2NF

Decomposition:

Citizens_CID_CN_CS_CNr_CZ_EID
Citizens_CZ_CL

## 5.2 Table: Citizens_CID_CN_CS_CNr_CZ_EID

Columns: CID, CN, CS, CNr, CZ, EID

Key: CID

FDs:

CID → CN
CID → CS
CID → CNr
CID → CZ
CID → EID

Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

## 5.3 Table: Citizens_CZ_CL

Columns: CZ, CL

Key: CZ

FDs: CZ → CL

Normal forms:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

# 6 Normalization of Coffees

## 6.1 Analysis

Primary key: (CID, DID, HID)
Determined FDs:
    DID → DN
    DID → DS
    CID → CN
    CID → CC
Other keys: None
Minimal cover:
    (CID, DID, HID) → (CID, DID, HID)
    DID → DN
    DID → DS
    CID → CN
    CID → CC
Normal form: 1NF
Decomposition:
    Coffees_CID_DID_HID
    Coffees_DID_DN_DS
    Coffees_CID_CN_CC

## 6.2 Table: Coffees_CID_DID_HID

Columns: CID, DID, HID
Key: (CID, DID, HID)
Other FDs: None
Normal form:

- All FDs are key FDs and therefore the table is in BCNF
- The key has three columns and when we ran the script it resulted in the MVD: DID → (HID, CID). Therefore, the table is not in 4NF and must be decomposed.

Decomposition:
    Coffees_DID_CID

    Coffees_DID_HID

### 6.2.1 Table: Coffees_DID_CID

Columns: DID, CID

Key: (DID, CID)

Other FDs: None

Normal forms:

- All the columns are part of the primary key and therefore the table is in BCNF
- The key has only two columns and is therefore in 4NF

### 6.2.2 Table: Coffees_DID_HID

Columns: DID, HID

Key: (DID, HID)

Other FDs: None

Normal forms:

- All the columns are part of the primary key and therefore the table is in BCNF
- The key has only two columns and is therefore in 4NF

## 6.3 Table: Coffees_DID_DN_DS

Columns: DID, DN, DS
Key: DID
FDs:

DID → DN
DID → DS

Normal form:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

## 6.4 Table: Coffees_CID_CN_CC

Columns: CID, CN, CC
Key: CID
FDs:

CID → CN
CID → CC

Normal form:

- All FDs are key FDs and therefore the table is in BCNF
- The key has only one column and is therefore in 4NF

# 7 Epilogue

After the normalization process all the tables should be on the highest normal form possible. Stella can now start using the database while enjoying her cup of coffee.

Two files were handed in with this report:

- DECOMPOSE.sql containing SQL commands to create the resulting database tables
- POPULATE.sql containing SQL commands to fill the resulting database tables from the tables in the original database