

Normalisering

Introduksjon til spørring mot flere tabeller

- likekobling/indre kobling
- ytre kobling

Skranker

- NULL-merke
- CONSTRAINT for inndata

To viktige begreper

- atomarkravet til et attributt

- 1) attributtet skal ikke kunne deles opp i flere deler
- 2) attributtet skal ikke repeteres

Det viktigste er at "de brukes atomisk"

"fødselsnr er atomisk" selv om det kan deles opp i

fødselsnr = fødselsdato + teller + kontroll
ddmmåå + 123 + 45

- determinant

- et attributt eller en gruppe av attributter som bestemmer/determinerer verdien av et annet attributt
postnr \rightarrow poststed

1. NF - etthvert attributt skal være atomisk

Student

Studentur

Navn \rightarrow blir Fornavn og Etternavn
Adresse \rightarrow blir Gateadresse og Postnr og Poststed
Telefon \rightarrow vi har ikke tlf 1, tlf 2, tlf 3
Kull ("privat", "jobb", "mobil") i samme entitetstype
"flere attributter" og/eller "flere entitetstyper"

2. NF - 1. NF samt unngå

"at et attributt kan avledes av dele av nøkkelen"

dvs: "kun aktuell problemstilling ved sammensatt nøkkel"

Examinens resultat

* Studentur

* Emnekode

Semester

Emne navn \rightarrow ÷

Resultat

"fjerne attributter"

1. NF + vi tillater ikke at dele av identifikatoren determinerer andre attributter

3. NF - 2. NF samt unngå

"at et attributt kan avledes av et attributt som ikke er nøkkel"

Student

Studentnr

Fornavn

Efternavn

Gateadresse

Telefon

Postnr

Poststed

Kull

Postnr
determinerer
Poststed

"splitte opp og bruke
flere entitetstyper"

2. NF + det meste ikke finnes
indirekte determineringer

Hovedregelen for normalisering

Boyce-Codd's normal form - BCNF

En datamodell er godt normalisert
(= "godt tilrettelagt for en relasjonsdatabase") dersom

- 1) alle attributter er atomiske (kravet til 1. NF)
- 2) enhver determinant er en kandidatnøkkel

↙
= "mulig primærnøkkel",
f. eks. e-postadresse

"Ja med determinering fra identifikator
eller mulig identifikator"

```

1  -- Skript for ansattpersonal2022 - utvidet med skranker
2
3  DROP SCHEMA IF EXISTS ansattpersonal2022;
4  CREATE SCHEMA ansattpersonal2022;
5
6  USE ansattpersonal2022;
7
8  -- oppretter tabeller, runde (1)
9  CREATE TABLE Stillingstype
10 (
11     Stillingskode CHAR(4),
12     Stillingsbetegnelse CHAR(20) NOT NULL,
13     CONSTRAINT StillingstypePK PRIMARY KEY(Stillingskode)
14 );
15
16 CREATE TABLE Avdeling
17 (
18     -- Avdelingsnr
19     -- Avdelingsnavn NOT NULL
20     -- CONSTRAINT for PK
21 );
22
23 CREATE TABLE Kurs
24 (
25     -- Kursnr
26     -- Kursnavn NOT NULL
27     -- CONSTRAINT for PK
28 );
29
30 CREATE TABLE Postkatalog
31 (
32     -- Postnr
33     -- Poststed NOT NULL
34     -- CONSTRAINT
35 );
36
37 -- oppretter tabeller, runde (2)
38 CREATE TABLE Ansatt
39 (
40     -- Ansattnr
41     -- Fornavn NOT NULL
42     -- Etternavn NOT NULL
43     -- Gateadresse
44     -- Telefonnr NOT NULL
45     -- Stillingskode NOT NULL???
46     -- Avdelingsnr NOT NULL???
47     -- Postnr NOT NULL???
48     -- CONSTRAINT for PK
49     CONSTRAINT AnsattStillingstypeFK FOREIGN KEY(Stillingskode) REFERENCES
        Stillingstype(Stillingskode),
50     -- CONSTRAINT for FK avdeling
51     -- CONSTRAINT for FK postkatalog
52 );
53
54 -- oppretter tabeller, runde (3)
55 CREATE TABLE Kursdeltagelse
56 (
57     -- Ansattnr
58     -- Kursnr
59     -- Dato
60     -- Vurdering NOT NULL???
61     CONSTRAINT KursdeltagelsePK PRIMARY KEY(Ansattnr,Kursnr,Dato),
62     -- CONSTRAINT for FK ansatt
63     -- CONSTRAINT for FK kurs
64     -- CONSTRAINT Vurderingsregel CHECK(Vurdering IN(oppramsing gyldiger verdier i
        ' separert med ,))
65 );
66

```



```

1  USE ansattpersonal2022;
2
3  -- Data i tabellen Ansatt
4  SELECT *
5  FROM Ansatt;
6
7  -- Data i tabellen Postkatalog
8  SELECT *
9  FROM Postkatalog;
10
11 -- Kryssproduktet av Postkatalog og Ansatt
12 SELECT *
13 FROM Postkatalog, Ansatt;
14
15 -- Likekobling, liste over ansatte med postadresser, med WHERE-betingelse
16 SELECT *
17 FROM Ansatt, Postkatalog
18 WHERE Ansatt.Postnr=Postkatalog.Postnr;
19
20 -- med et kolonneutvalg
21 SELECT Ansattnr, Fornavn, Etternavn, Gateadresse, Ansatt.Postnr, Poststed
22 FROM Ansatt, Postkatalog
23 WHERE Ansatt.Postnr=Postkatalog.Postnr;
24
25 -- Likekobling, liste over ansatte med postadresser, med INNER JOIN
26 SELECT Ansattnr, Fornavn, Etternavn, Gateadresse, Ansatt.Postnr, Poststed
27 FROM Ansatt INNER JOIN Postkatalog
28     ON Ansatt.Postnr=Postkatalog.Postnr;
29
30 -- Likekobling 3 tabeller, liste over ansatte med stilling og avdeling, med
    WHERE-betingelser
31 SELECT Etternavn, Fornavn, Stillingsbetegnelse, Avdelingsnavn
32 FROM Ansatt, Stillingstype, Avdeling
33 WHERE Ansatt.Stillingskode=Stillingstype.Stillingskode
34     AND Ansatt.Avdelingsnr=Avdeling.Avdelingsnr;
35
36 -- koble tre eller flere tabeller, først 2, "steg for steg" tankegang
37 SELECT Etternavn, Fornavn, Stillingsbetegnelse
38 -- , Avdelingsnavn
39 FROM Ansatt, Stillingstype
40 -- , Avdeling
41 WHERE Ansatt.Stillingskode=Stillingstype.Stillingskode
42 -- AND Ansatt.Avdelingsnr=Avdeling.Avdelingsnr
43 ;
44
45 -- ved bruk av INNER JOIN kan det være enda viktigere å bygge opp "steg for
    steg"
46 -- først Ansatt mot avdeling (innerste JOIN) før
47 -- Stillingstype kobles mot mellomresultatet av innerste JOIN
48 SELECT Etternavn, Fornavn, Stillingsbetegnelse, Avdelingsnavn
49 FROM Stillingstype INNER JOIN
50     (Ansatt INNER JOIN Avdeling
51         ON Ansatt.Avdelingsnr=Avdeling.Avdelingsnr)
52     ON Stillingstype.Stillingskode=Ansatt.Stillingskode;
53
54 -- alternativ med INNER JOIN
55 -- først Ansatt mot Stillingstype (innerste JOIN) før
56 -- Avdeling kobles mot mellomresultatet av innerste JOIN
57 SELECT Etternavn, Fornavn, Stillingsbetegnelse, Avdelingsnavn
58 FROM Avdeling INNER JOIN
59     (Ansatt INNER JOIN Stillingstype
60         ON Ansatt.Stillingskode=Stillingstype.Stillingskode)
61     ON Avdeling.Avdelingsnr=Ansatt.Avdelingsnr;
62
63 -- View/utsnitt
64 -- oppretting av View Ansattliste
65 -- Etternavn, Fornavn, Stilling, Avdeling med info fra de 3 tabellene
66 -- Ansatt, Stillingstype, Avdeling
67 DROP VIEW IF EXISTS Ansattliste;
68
69 CREATE VIEW Ansattliste (Etternavn, Fornavn, Stilling, Avdeling) AS

```

```

70  (SELECT Etternavn, Fornavn, Stillingsbetegnelse, Avdelingsnavn
71  FROM Ansatt, Stillingstype, Avdeling
72  WHERE Ansatt.Stillingskode=Stillingstype.Stillingskode
73  AND Ansatt.Avdelingsnr=Avdeling.Avdelingsnr);
74
75  -- kan så kjøre spørringer mot view'et
76  SELECT *
77  FROM Ansattliste
78  ORDER BY Etternavn;
79
80  -- ytre koblinger
81  -- ønsker også stillingsbetegnelser som ikke er i bruk (ingen har pt)
82  SELECT *
83  FROM Stillingstype LEFT OUTER JOIN Ansatt
84  ON Stillingstype.Stillingskode=Ansatt.Stillingskode;
85
86  -- ekvivalent med (samme spørreresultat, annen presentasjon)
87  SELECT *
88  FROM Ansatt RIGHT OUTER JOIN Stillingstype
89  ON Ansatt.Stillingskode=Stillingstype.Stillingskode;
90
91  -- ønsker også avdelinger som ingen er tilknyttet pt
92  SELECT *
93  FROM Avdeling LEFT OUTER JOIN Ansatt
94  ON Avdeling.Avdelingsnr=Ansatt.Avdelingsnr;
95
96  -- ekvivalent med (samme spørreresultat, annen presentasjon)
97  SELECT *
98  FROM Ansatt RIGHT OUTER JOIN Avdeling
99  ON Ansatt.Avdelingsnr=Avdeling.Avdelingsnr;
100

```