

Eksamen - Databasesystemer våren 2025

Marius Aasgaard, Stian Hetsch

April 22, 2025

1 Normalisering

Vi har hentet inspirasjon fra skisporet.no og data fra swixsport.com, og basert på det vi finner ser vi på det som fornuftig å ha med følgende informasjon i vår database:

Produkt:

- Produktnavn på skismøring
- Produktkategori: (tørrvoks, klister, glider)
- Varenummer
- Produktbeskrivelse
- Brukernivå: (nybegynner, erfaren, konkurranseløper)
- Snøtype: (nyfallen, gammel, våt)
- Temperaturintervall: F.eks. -10°C til -3°C

Løypeforhold:

- Destinasjon
- Tidspunkt for skitur
- Temperatur registrert
- Snøtype registrert
- Løype
- Spor: (klassisk, skøyting)

- Segment: (lengde, stigning, høydetap)

smoring: Produkt, Temperaturintervall, Snobeskrivelse, Brukerniva

Produktnavn: 1, VP45 Pro Blue, Festevoxs TemperaturIntervall: -5 til -1, -8 til -3 Snobeskrivelse: Nyfallen snø, gammel snø Brukernivå: Nybegynner, erfaren og konkurranseløper

løypeforhold: Destinasjon, Tidspunkt, Temperatur, Snobeskrivelse, Loype, Spor, Segment

Destinasjon: (2, 'Sjusjøen', 21.04.25 23:21:00, -4°C, Nyfallen snø, 'Elgåsen', 'Klassisk', 600, 9.5, 2.7)

1.1 1NF - Første normalform

løypeforhold: DestinasjonID, DestinasjonNavn, Tidspunkt, Temperatur, Snobeskrivelse, LoypeNavn, Sporbeskrivelse, SegmentLengde, SegmentStigning, SegmentHoydetap smoring: ProduktID, Produktnavn, Produktkategori, Produktkategoribeskrivelse, Produktnummer, Produktbeskrivelse, Brukerniva, Snotype, TemperaturMinimum, TemperaturMaksimum, Snobeskrivelse

Funksjonelle avhengigheter:

Løypeforhold:

- DestinasjonID \rightarrow DestinasjonNavn
- DestinasjonID, LoypeID \rightarrow Sporbeskrivelse
- DestinasjonID, Tidspunkt \rightarrow Temperatur, Snobeskrivelse
- LoypeID \rightarrow LoypeNavn
- LoypeID, SegmentID \rightarrow SegmentLengde, SegmentStigning, SegmentHoydetap
- SportypeID \rightarrow Sporbeskrivelse

Smøring:

- ProduktID \rightarrow ProduktNavn, ProduktKategori, ProduktNummer, Produktbeskrivelse, Snotype, Brukerniva
- ProduktID, Snotype \rightarrow TemperaturMinimum, TemperaturMaksimum
- Snotype \rightarrow Snobeskrivelse
- Brukerniva \rightarrow Brukernivabeskrivelse

1.2 2NF

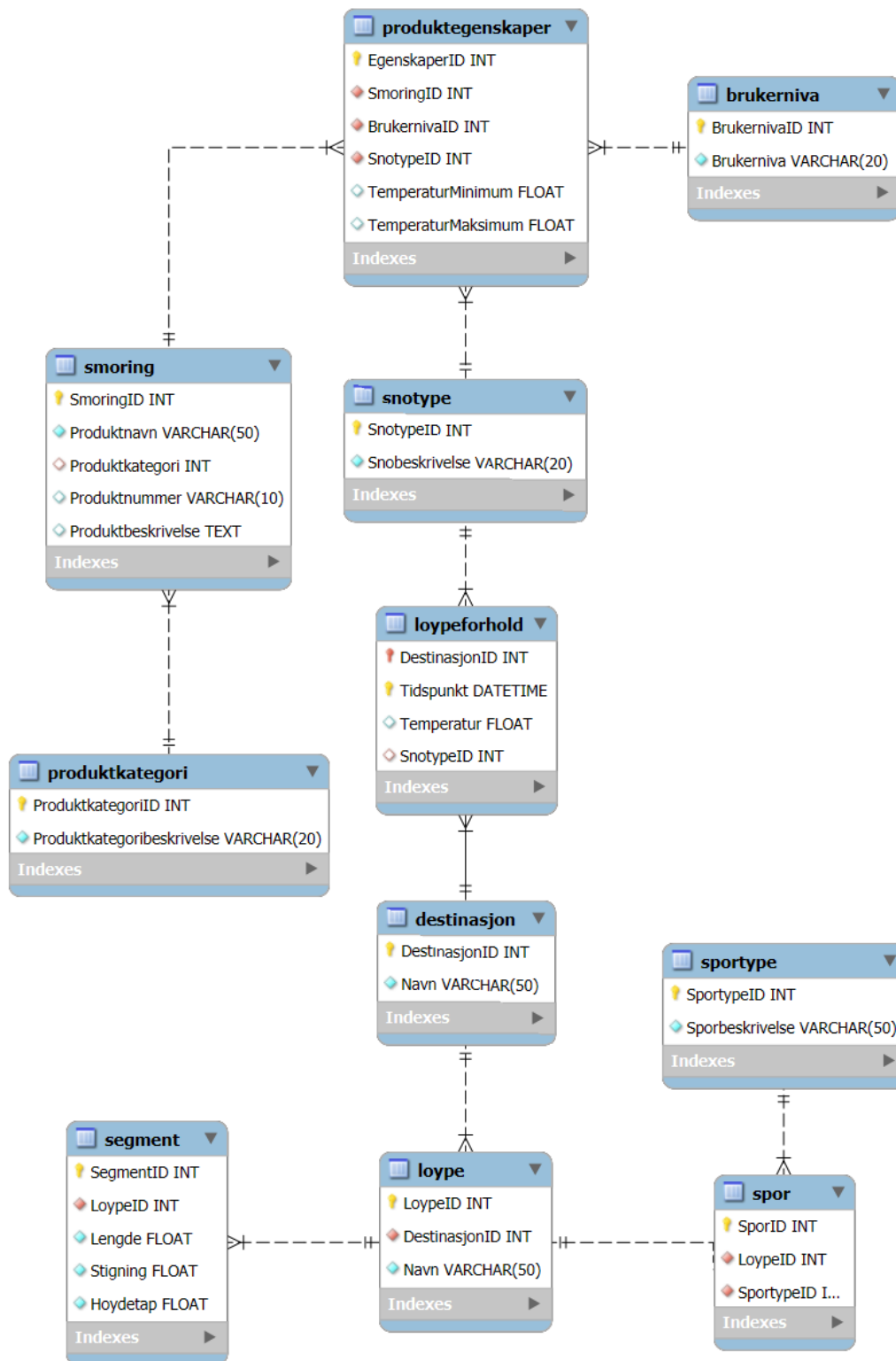
- Destinasjon(DestinasjonID, DestinasjonNavn)
- Loype(LoypeID, DestinasjonID*, LoypeNavn)
- Segment(SegmentID, LoypeID*, Lengde, Stigning, Hoydetap)
- Sportype(SportypeID, Sporbeskrivelse)
- Spor(SporID, LoypeID*, SportypeID*)
- Snotype(SnotypeID, Snobeskrivelse)
- Loypeforhold(DestinasjonsID*, Tidspunkt, Temperatur, SnotypeID*)
– Forutsetter at det ikke er mulig å registrere 2 forskjellige temperaturer / snøtyper på samme tidspunkt, på samme sted
- Produktkategori(ProduktkategoriID, Produktkategoribeskrivelse)
- Smoring(ProduktID, BrukernivaID*, SnotypeID*, ProduktNavn, Produktkategori*, Produktnummer, Produktbeskrivelse, , TemperaturMinimum, TemperaturMaksimum)

1.3 3NF

- Destinasjon(DestinasjonID, DestinasjonNavn)
- Loype(LoypeID, DestinasjonID*, LoypeNavn)
- Segment(SegmentID, LoypeID*, Lengde, Stigning, Hoydetap)
- Sportype(SportypeID, Sporbeskrivelse)
- Spor(SporID, LoypeID*, SportypeID*)
- Snotype(SnotypeID, Snobeskrivelse)
- Loypeforhold(DestinasjonsID*, Tidspunkt, Temperatur, SnotypeID*)
- Produktkategori(ProduktkategoriID, Produktkategoribeskrivelse)
- Smoring(ProduktID, BrukernivaID*, SnotypeID*, ProduktNavn, Produktkategori*, Produktnummer, Produktbeskrivelse, , TemperaturMinimum, TemperaturMaksimum)
- Produktegenskaper(EgenskaperID, ProduktID*, BrukernivaID*, SnotypeID*, TemperaturMinimum, TemperaturMaksimum)

1.4 BCNF

2 ER-modellering



3 Database implementasjon

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Navigator' pane shows the 'eksamendbsvår25' database with a tree view of tables and views. The 'brukerniva' table is selected. The main pane shows the SQL script for creating the 'brukerniva' table. Below the script, the 'Result Grid' shows the table's structure and data.

SQL Script for brukerniva:

```
1 • SELECT * FROM eksamendbsvår25.brukerniva;
2
3 • CREATE TABLE brukerniva(
4     BrukernivaID INT PRIMARY KEY,
5     Brukerniva VARCHAR (20) NOT NULL
6 );
```

Result Grid for brukerniva:

BrukernivaID	Brukerniva
1	Nybegynner
2	Erfaren
3	Konkurranseløper
*	NULL

The screenshot also shows the 'destinasjon' table being created. The SQL script for creating the 'destinasjon' table is shown below.

SQL Script for destinasjon:

```
1 • SELECT * FROM eksamendbsvår25.destinasjon;
2
3 • CREATE TABLE destinasjon(
4     DestinasjonID INT AUTO_INCREMENT PRIMARY KEY,
5     Navn VARCHAR(50) NOT NULL
6 );
```

Result Grid for destinasjon:

DestinasjonID	Navn
1	Hurdal Hele Året
2	Sjusjøen
*	NULL

Navigator

SCHEMAS

Filter objects

- eksamendbsvår25
 - Tables
 - brukerniva
 - destinasjon
 - loype
 - loypeforhold
 - produkttegnaskaper
 - produktkategori
 - segment
 - smoring
 - snotype
 - spor
 - sportype
 - Views
 - alternativsmoring
 - alternativsmoring_loy
 - anbefaltsmoring_dest
 - destinasjonoversikt
 - loypeoversikt
 - smoringoversikt
 - Stored Procedures
 - Functions
- eksamenvår25
- sakila
- sys
- world

Opprettelse av alle tabeller

DummyDataALT

Opprettelse av alle viewsa

produktkategori

segment

smoring

snotype

Limit to 1000 rows

```

1 • SELECT * FROM eksamendbsvår25.snotype;
2
3 • CREATE TABLE snotype(
4     SnotypeID INT PRIMARY KEY,
5     Snobeskrivelse VARCHAR(20) NOT NULL
6 );

```

Result Grid

SnotypeID	Snobeskrivelse
1	Nyfallen snø
2	Gammel snø
3	Våt snø
NULL	NULL

Navigator

SCHEMAS

Filter objects

- eksamendbsvår25
 - Tables
 - brukerniva
 - destinasjon
 - loype
 - loypeforhold
 - produkttegnaskaper
 - produktkategori
 - segment
 - smoring
 - snotype
 - spor
 - sportype
 - Views
 - alternativsmoring
 - alternativsmoring_loy
 - anbefaltsmoring_dest
 - destinasjonoversikt
 - loypeoversikt
 - smoringoversikt
 - Stored Procedures
 - Functions
- eksamenvår25
- sakila
- sys
- world

Opprettelse av alle tabeller

DummyDataALT

Opprettelse av alle viewsa

smoring

snotype

spor

Limit to 1000 rows

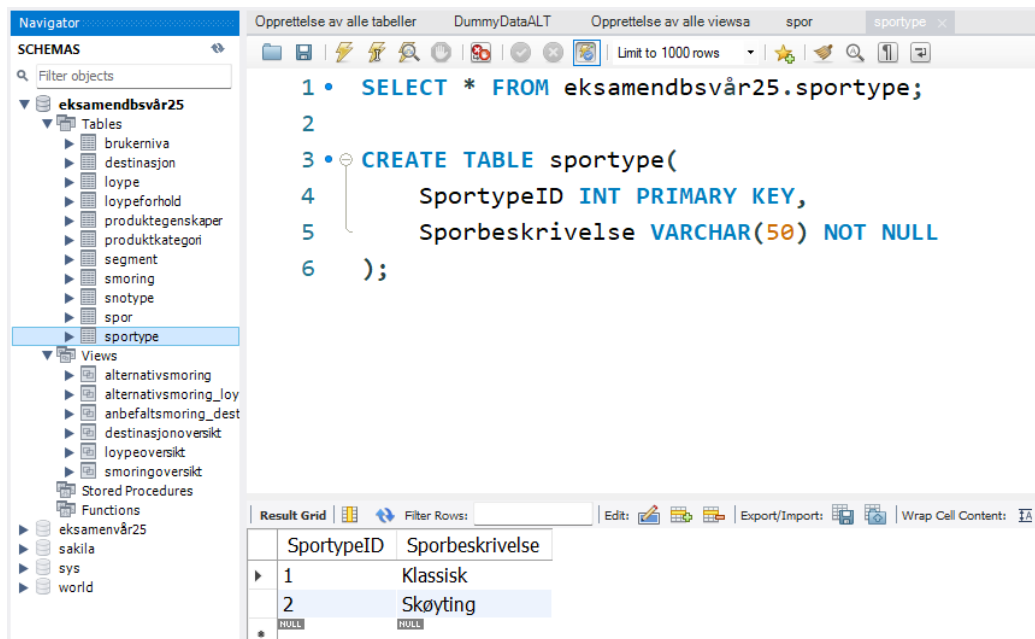
```

1 • SELECT * FROM eksamendbsvår25.spor;
2
3 • CREATE TABLE spor(
4     SporID INT AUTO_INCREMENT PRIMARY KEY,
5     LoypeID INT NOT NULL,
6     SportypeID INT NOT NULL,
7     FOREIGN KEY (LoypeID) REFERENCES loype(LoypeID)
8     ON DELETE CASCADE ON UPDATE CASCADE,
9     FOREIGN KEY (SportypeID) REFERENCES sportype(SportypeID)
10    ON DELETE RESTRICT ON UPDATE CASCADE
11 );

```

Result Grid

SporID	LoypeID	SportypeID
1	1	1
2	2	1
3	2	2
NULL	NULL	NULL



4 Dummy data

Hovedformål med dummydata: Forklare hvorfor dummydata er nødvendig og hva vi tenker at den skal demonstrere. Altså systemets funksjonelle krav og generelle spesifikasjoner. Vi har derfor valgt å ta med realistiske eksempler på ... smøring, løyper, løypeforhold og brukernivå.

Hvordan har vi tenkt?: Forklare hvilke tabeller vi har fylt med dummydata og hva tanken bak tabellen er. Hvilke varianter har vi med av smøringer? Produktkategorier, Temperaturintervall osv...

Begrunnelse: Her forklarer vi hvorfor nettopp dataen vi har brukt ble valgt og forklarer at de dekker et bredt spekter av skismøring med ulike kombinasjoner. Alle smøringer er realistiske fra Swix sine produkter og det sørger videre for at alle views og spørringer gir meningsfulle resultater

5 SQL-spørringer

Navn på spørringen: Anbefaltsmøringalternativ f.eks

Hvilket systemkrav prøver vi å oppfylle med denne spørringen - hva skal spørringen gjøre?: Forklarer da hvilket resultat spørringen fører til og hvilket systemkrav den oppfyller.

Kommentere SQL-spørringen: Her kommenter vi SQL-koden under hver linje og forklarer nøyaktig hva som skjer

Eksempel på resultat: Her tar vi med skjermbilde av resultatet, f.eks to eller tre rader (hvis ikke bare ett resultat)

Begrunnelse: Forklarer her hvorfor spørringen er logisk og hvordan f.eks temperaturinput henger sammen med at den faller mellom et gitt temperaturintervall....