IN2090 – Databaser og datamodellering

01 – Introduksjon og oversikt over emnet

Leif Harald Karlsen
leifhka@ifi.uio.no



Universitetet i Oslo

Oversikt over kurset

- To hovedkategorier: Modellering og SQL
- Tirsdager: Forhåndsinnspilte videoer (dekker pensum)
- Torsdager: Fysisk forelesning med gjennomgang av eksempler og spørsmål
- Forventes at dere ser/deltar på begge

Oversikt over kurset

Uke	Tema	Dato	Foreleser	Innleveringer
1	Introduksjon og motivasjon	Tors 22.08	Leif Harald	
2	*Modellering:* Relasjonsmodellen	Tors 29.08	Leif Harald	
3	*Modellering:* ER-modellering 1	Tors 05.09	Leif Harald	
4	*Modellering:* ER-modellering 2	Tors 12.09	Leif Harald	
5	*DB:* Grunnleggende SQL	Tors 19.09	Leif Harald	
6	*DB:* Datamanipulering, skranker og views	Tors 26.09	Leif Harald	1. Modellering
7	*Modellering:* Normalformer	Tors 03.09	Arild Waaler	
8	*Modellering:* Tapsfri dekomposisjon	Tors 10.10	Arild Waaler	2. Enkel SQL
9	*DB:* Aggregering og sortering	Tors 17.10	Arild Waaler	
10	*DB:* Ytre joins og mengde operatorer	Tors 24.10	Arild Waaler	3. Normalformer
11	*DB:* Programmering med databaser	Tors 31.10	Arild Waaler	
12	*DB:* Sikkerhet i databaser	Tors 07.11	Arild Waaler	4. Avansert SQL
13	*DB:* Indekser og spørreprosessering	Tors 14.11	Arild Waaler	
14	Repetisjon	Tors 21.11	Arild Waaler	5. Programmering med SQL

Mer om timeplan og obliger:

`https://www.uio.no/studier/emner/matnat/ifi/IN2090/h24/index.html`

Forelesere



Leif Harald Karlsen

leifhka@ifi.uio.no



Arild Waaler

arild@ifi.uio.no

Gruppelærere og rettere

Gruppelærere:

- Jamila Mehmandarova
- Julia Lundeby
- Ole-Sander Halla Johansen
- Lise Katarina Grønland
- Viljar Drevland Hardersen
- William Henrik Behn

Rettere:

- Mehmet Capa
- Nora Elise Haugen
- Sapnil Aditya Ayman

Andre innvolverte



Egor Kostylev

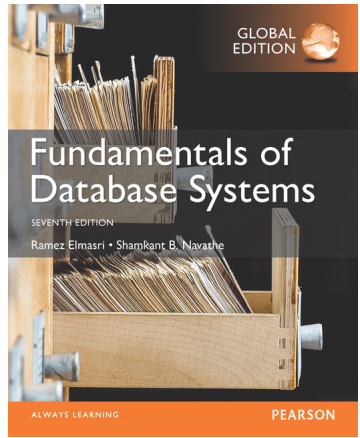
egork@ifi.uio.no

Gruppetimer

- Én-til-én hjelp med ukesoppgaver og innleveringer
- Gruppelærerne velger selv struktur på gruppetimene
- 6 grupper
- Det er lov å bytte gruppe, men spør gruppelæreren først
- Gruppetimenes tema er alltid forelesningstemaet for uken før!
- Første gruppetime er neste uke

Pensum

- Videoene og foilene til kurset
  - Publisert på timeplanen på semestersiden
  - Foiler/screencast fra torsdagsforelesningene blir også publisert
- Ukesoppgavene og innleveringsoppgavene
  - Gjennomgåes på gruppetimene
- Boken *Fundamentals of Database Systems*
  - Tykk bok, men skal bare bruke deler av den
  - Se timeplanen for hva som er pensum
  - Gjenbrukes i andre kurs (IN3020)



Ukesoppgaver

- Publiseres i slutten av uken[1]
- Gjennomgås på gruppetimen uken etter publisering
- Løsningsforslag publiseres i slutten av uken den gjennomgås

[1] `https://www.uio.no/studier/emner/matnat/ifi/IN2090/h24/ukesoppgaver/`

Obliger og innleveringsoppgaver

- Fem innleveringsoppgaver med tilbakmelding
- Første av disse er obligatorisk
  - Fokus er på læring, ikke på løsning
  - Sterkt anbefalt å gjøre alle!
  - Essensiell del av undervisningen
  - Frihet under ansvar
- Publiseres torsdager, frist på fredager[2]
- Leveres via Devilry
- To forsøk på obligen
- De andre kan man be om tilbakemelding på et andre forsøk
- Dersom du blir syk el., kan man be om utsatt frist

[2] `https://www.uio.no/studier/emner/matnat/ifi/IN2090/h24/innleveringer/index.html`

Eksamen

- Eksamen: 13. desember (4 timer, Silurveien)
- Bokstavkarakter (A-F)
- Alle skriftlige hjelpemidler er tillatt

Programvare

- RDBMS: PostgreSQL 16[2]
  - Avansert relasjonell databasesystem
  - Åpen kildekode og støtter de fleste platformer
  - Kan f.eks. bruke `psql`[4] eller Edbit[5]
- Modelleringsvertøy: Dia[3] eller `app.diagrams.net`
  - Både Dia og `app.diagrams.net` er generelle diagram-verktøy
  - Begge har åpen kildekode og støtter de fleste platformer (Linux, Mac, Windows)
  - Trenger IKKE bruke dette om man ikke ønsker, kan tegne på papir eller bruke andre programmer!







[2] `https://www.postgresql.org/`
[3] `http://dia-installer.de/download/index.html`
[4] `https://www.postgresql.org/docs/16/app-psql.html`
[5] `https://gitlab.com/leifhka/edbit`

Databaser

- IMDB: Internet Movie DataBase
  - Filmdatabasen
  - Alle filmer, serier, episoder, skuespillere, roller, osv. frem til 2011
  - Stor database (2.4G)
  - >1 mil. filmer/serier/episoder, >10 mil. roller, >1.7 mil. personer, osv.
- Northwind
  - Eksempel-database laget av Microsoft Research for opplæring i databaser
  - Database over produkter, bestillinger, kunder, osv.
  - Liten database, men realistisk oppsett
- Personlig database
  - Alle har hver sin egne database
  - Du har full tilgang (kan lage, slette, oppdatere data, osv.)
- Satt opp av USiT for dere
- Mer info om tilkobling osv. kommer senere

UiODiscourse

- Brukes for diskusjon, spørsmål og kommentarer rundt pensum, oppgaver, obliger, osv.
- Bruker UiO-brukernavn og passord for innlogging
- Støtter:
  - Publisering med bruker og anonymt
  - Markering av riktig svar
  - Math-mode og code-blocks (SQL)
  - Kategorier og tags
  - osv.
- Guide ligger her: `https://www.mn.uio.no/english/services/it/discourse-uio/astro-discourse-instructions-(students).pdf`

Vår UiODiscourse

`https://astro-discourse.uio.no/c/in2090-24h`

UiODiscourse: Råd og regler

- Gjerne hjelp hverandre, svar på hverandres spørsmål, osv.
- Vær hyggelig! :)
- Sjekk om spørsmålet du har allerede er postet

Motivasjon

Eksempler på hva dere skal lære

Databaser og SQL

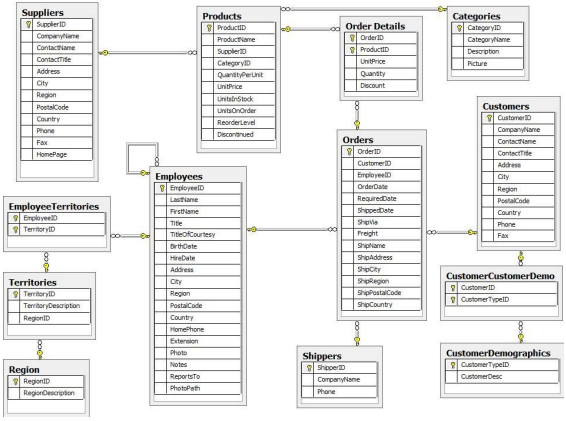
Finn topp 3 rangerte action-filmer med Harrison Ford som skuespiller.

```
SELECT f.title
FROM person AS p
  INNER JOIN filmparticipation AS fp USING (personid)
  INNER JOIN filmgenre AS fg USING (filmid)
  INNER JOIN filmrating AS fr USING (filmid)
  INNER JOIN film AS f USING (filmid)
WHERE p.firstname = 'Harrison' AND
  p.lastname = 'Ford' AND
  fp.parttype = 'cast' AND
  fg.genre = 'Action'
ORDER BY fr.rank DESC
LIMIT 3;
```

Eksempler på hva dere skal lære

Databaser og SQL

Northwind-databasen:



Eksempel-oppgave:

Finn navn og total regning for hver kunde som har kjøpt fler enn 1000 varer, sortert alfabetisk etter firmanavn.

```sql
SELECT c.company_name,
  sum(d.unit_price * d.quantity) AS bill
FROM customers AS c
  INNER JOIN orders AS o USING (customer_id)
  INNER JOIN order_details AS d USING (order_id)
GROUP BY c.customer_id, c.company_name
HAVING sum(d.quantity) > 1000
ORDER BY c.company_name;
```

Mer avansert eksempel

Vi vil lage en spørring som finner ut:

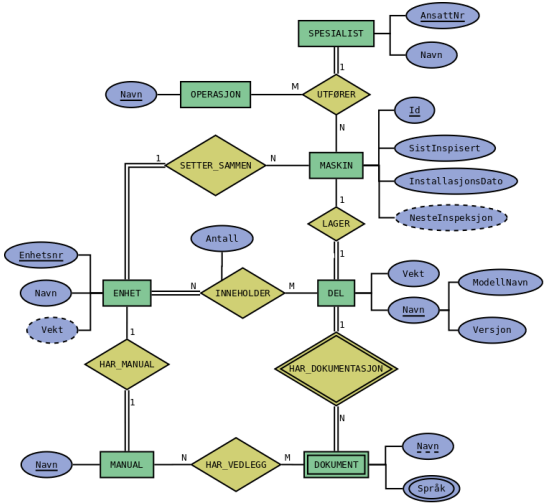
- hvilke produkter vi kan anbefale en kunde å kjøpe,
- basert på hva kunden har kjøpt,
- og hva andre kunder som har kjøpt det samme har kjøpt.

Mer avansert eksempel

```
WITH
  bought AS ( -- Relaterer kunde-IDer til produkt-IDene til det de har kjøpt
    SELECT DISTINCT c.customer_id, d.product_id -- Vil ikke ha duplikater!
    FROM customers AS c
         INNER JOIN orders USING (customer_id)
         INNER JOIN order_details AS d USING (order_id)
  ),
  correspondences AS ( -- Relaterer par av produkter til antallet ganger disse er kjøpt av samme kunde
    SELECT b1.product_id AS prod1, b2.product_id AS prod2, count(*) AS correspondence
    FROM bought AS b1
         INNER JOIN bought b2 USING (customer_id)
    WHERE b1.product_id != b2.product_id -- Fjern par hvor produktene er like
    GROUP BY b1.product_id, b2.product_id -- Gruppér på par av produkter
    HAVING count(*) > 18 -- Antall korrespondanser bør være litt høyt
  ),
  reccomend AS ( -- Relaterer kunde-IDer til anbefalte produkters IDer
    SELECT DISTINCT b.customer_id, c.prod2 AS product_id -- Vil ikke ha duplikater!
    FROM correspondences AS c
         INNER JOIN bought AS b
         ON (b.product_id = c.prod1)
    WHERE NOT c.prod2 IN       -- Fjern produkter som kunden allerede har kjøpt
            (SELECT product_id
             FROM bought AS bi
             WHERE b.customer_id = bi.customer_id)
   )
-- Til slutt finn navn på både kunde og produkt og aggreger produktnavnene
-- for hver kunde i ett array med aggregatfunksjonen array_agg
SELECT c.company_name, array_agg(p.product_name) AS reccomended_products
FROM customers AS c INNER JOIN reccomend AS r USING (customer_id)
     INNER JOIN products AS p USING (product_id)
GROUP BY c.customer_id, c.company_name;
```

Eksempler på hva dere skal lære

Modellering og ER



Eksempel ER-modell av informasjon om fabrikk som produserer enheter.

Litt mer motivasjon

Tall fra Stack Overflows *Developer Syrvey 2024*:

`https://survey.stackoverflow.co/2024/technology`

- SQL nest mest brukte språket blant profesjonelle utviklere!
- (PostgreSQL mest brukte databasesystemet)

Et par ord om læring

- Delta på undervisningen, gjør alle oppgaver, les tilbakemeldinger, osv.
- Husk: dere er er for å lære, ikke for å løse problemer!
- En av grunnene til endret oblig-opplegg
- Sjekk egen forståelse, og vær ærlige med dere selv
- Denne muligheten til å virkelig lære noe dypt får dere sjeldent igjen!

Forelesning vs. video/opptak

Hvorfor ha/møte til fysiske forelesninger?

- Gir noe ekstra å se noe i levende live (f.eks. teater)
- Tilskuere gir energi
- Tilskuere gir indirekte tilbakemelding (f.eks. blanke ansikter)
- Dere kan også gi direkte tilbakemelding (spørsmål, kommentarer)
- Dere kan diskutere med hverandre, snakke sammen
- Bygger et godt læringsmiljø

Så møt opp! :)

Teknologi, språkmodeller og læring

- Mye ny og gammel teknologi som kan løse problemer
- Store språkmodeller (som ChatGPT, Bing, osv.) er flotte verktøy!
- Vil nok klare mange av oppgavene i dette kurset
- Derfor to viktige spørsmål:
  - Er det noen grunn til å lære seg pensum?
  - Hvordan bruke store språkmodeller i studiene?

Er det noen grunn til å lære seg pensum? (1)

JA! :D

- Språkmodeller er langt flinkere på problemene i dette kurset enn pensum generelt!
  - Problemene dere får er “enkle” og vanlige
  - Modellene er derfor godt trent på disse, men ikke “ekte” problemer
  - Modellene sliter med å sette sammen mindre program-snutter til større løsninger
- Ikke alltid mulig å bruke språkmodeller
  - Sensitive domener
  - For store eller komplekse problemer
  - Dette kan endre seg, men vanskelig å spå fremtiden

Er det noen grunn til å lære seg pensum? (2)

JA! :D

- Må gi riktig input/prompt
  - Krever ofte dyp forståelse av domenet
  - Må vite hva som er mulig og ikke
- Språkmodellene gjør feil
  - Feil i selve løsningen
  - Løser feil problem
  - Svarene må derfor sjekkes
- Språkmodellene kan kun det den kan
  - Nye teknologier kommer og går
  - Kan ikke forskning og utvikling
- Det er gøy!

Hvordan bruke store språkmodeller i studiene?

- Ikke bruk språkmodeller til å løse oppgaver!
  - Læring krever at man faktisk tenker
  - Lærer lite av å bare se en løsning
- Bruk dem heller for samtaler om pensum!
  - Still generelle spørsmål og oppfølgingsspørsmål
  - Husk: De gjør fortsatt feil! Test ut svaret den gir
- Læring er noe du må gjøre selv og kan ikke automatiseres!

Nyttige lenker (Ikke pensum)

- Tidligere års forelesninger (opptak, slides, videoer, osv.)
  - https://www.uio.no/studier/emner/matnat/ifi/IN2090/h19/timeplan/index.html
  - https://www.uio.no/studier/emner/matnat/ifi/IN2090/h20/timeplan/index.html
  - https://www.uio.no/studier/emner/matnat/ifi/IN2090/h21/timeplan/index.html
  - https://www.uio.no/studier/emner/matnat/ifi/IN2090/h22/timeplan/index.html
  - https://www.uio.no/studier/emner/matnat/ifi/IN2090/h23/timeplan/index.html
- Mer om databaser:
  - Engelsk: `https://en.wikipedia.org/wiki/Database`
  - Norsk: `https://no.wikipedia.org/wiki/Database`
- Mer om relasjons-databaser:
  - Engelsk: `https://en.wikipedia.org/wiki/Relational_database`
  - Norsk: `https://no.wikipedia.org/wiki/Relasjonsdatabase`
- Mer om datamodellering:
  - Engelsk: `https://en.wikipedia.org/wiki/Data_modeling`
  - Engelsk: `https://en.wikipedia.org/wiki/Data_model`
- *What is the benefit of learning SQL?* (engelsk):
  `https://www.quora.com/What-is-the-benefit-of-learning-SQL`