Hva er en database?

* En database er en samling med informasjon. Denne benyttes for å lagre store mengder data på en strukturert måte. Eksempler: Facbook, Imdb, VG etc.

Hvordan lage en database?

Eksempel (personregister)

|  |  |  |  |
| --- | --- | --- | --- |
| Personnummer | Navn | Kjønn | Telefonnummer  (datatype) |
| 11111111111 | Bartosz Piasecki | M | 22222222, 44444444 |
| 22222222222 | Ola Normann | M | 88888888 |
| 33333333333 | Kari Nordame | K | 55555555, 66666666 |

* Alle datatyper må deles opp så mye som mulig. Når en datatype har bare en opplysning sier vi at datatypen er atomære

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Personnummer | Fornavn | Etternavn | Kjønn | Telefon | Mobil |
| 11111111111 | Bartosz | Piasecki | M | 22222222 | 44444444 |
| 22222222222 | Ola | Normann | M | 88888888 |  |
| 33333333333 | Kari | Nordame | K | 55555555 | 66666666 |

Normalisering

* For å få effektive datamodeller må vi benytte normalisering. I dette kurse skal vi lære de tre første normalformene. For å oppfylle første normalform (1NF) må vi sørge for at alle datatyper tilfredsstiller atomærkravet. Atomærkravet betyr å dele opp opplysningene så mye som mulig. I tillegg må vær tabell ha en primærnøkkel.

MySQL Workbench

* Programmet lar oss lage datamodeller. En datamodell er en oversikt over hvilke tabeller en database skal inneholde, og hvordan dette skal kobles sammen.

Datatype

Alle koloner i en datamodell skal ha en tilhørende datatype. Datatyper forteller hva slags informasjon som skal lagres.

* INT (INTEGER) – Heltall
* FLOAT – desimaltall
* VARCHAR(10) – Kort tekst
* DATETIME – dato og tid
* TEXT – lang tekst

**Relasjonsdatabaser**

**Hva er en relasjonsdatabase?**

En relasjonsdatabase består av to eller flere tabeller som henger sammen etter spesielle regler. Vi oppretter da relasjoner (koblinger) mellom tabellene. En relasjon sier noe om sammenhengen mellom to tabeller. Vi kommer til å benytte oss av to varianter: *én-til-mange og mange-til-mange.*

**Eksempel (register over elever og deres kontaktlærer)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ElevID** | **Fornavn** | **Etternavn** | **Klasse** | **Kl\_Fornavn** | **Kl\_Etternavn** |
| Petarn | Petter | Arnesen | 2F | Ingunn | Wettestad |
| Sanarn | Sander | Arntzen | 2F | Ingunn | Wettestad |
| Sigber | Sigurd | Berg | 2C | Lars | Indresand |
| Jenbjo | Jens Jacob | Bjørnslett | 2D | Njål | Engløk |
| Eribjo | Erik | Bjørnstad | 2F | Ingunn | Wettestad |
| Kendan | Kent | Daneel | 2D | Njål | Engløk |

Dobbeltlagring av informasjon! Dataene må bygges opp systematisk slik at vi unngår dobbeltlagring av informasjon (redundans). Det vil si at samme informasjon ikke skal lagres flere steder. Dataene må lagres ryddig for at det skal være enkelt for brukeren å finne informasjon og legge til informasjon.

Vi deler opp i to tabeller. En for elever og en for klassestyrere. Én-til-mange relasjon. Én klassestyrer har mange elever! Én elev har kun klassestyrer.

|  |  |  |  |
| --- | --- | --- | --- |
| **ElevID** | **Fornavn** | **Etternavn** | **Klasse** |
| Petarn | Petter | Arnesen | 2F*\*(fremmednøkkel)* |
| Sanarn | Sander | Arntzen | 2F*\*(fremmednøkkel)* |
| Sigber | Sigurd | Berg | 2C*\*(fremmednøkkel)* |
| Jenbjo | Jens Jacob | Bjørnslett | 2D*\*(fremmednøkkel)* |
| Eribjo | Erik | Bjørnstad | 2F*\*(fremmednøkkel)* |
| Kendan | Kent | Daneel | 2D*\*(fremmednøkkel)* |

|  |  |  |
| --- | --- | --- |
| **Klasse** | **Kl\_Fornavn** | **Kl\_Etternavn** |
| 2C | Lars | Indresand |
| 2D | Njål | Engløk |
| 2F | Ingunn | Wettestad |

**Fremmednøkkel**

Når vi skal koble to tabeller sammen, må vi angi hvordan de skal kobles sammen. En fremmednøkkel er en kolonne som peker på en primærnøkkel i en annen tabell.

**Tredje normalform (3NF)**

For å oppfylle tredje normalform (3NF) må vi sørge for at første og andre normalform er oppfylt, og at ingen av kolonnene bestemmes av noe annet enn primærnøkkelen.

Klasse bestemmer Kl\_Fornavn og Kl\_Etternavn. Vi splitter i to taballer og oppretter en én-til-mange relasjon mellom tabellene.

**Mange til mange relasjoner**

Eksempel-karakteroverskrift:

En elev har mange fag, mens ett fag har mange elever

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ElevID | Fornavn | Etternavn | Fagkode | Fagnavn | Karakter |
| 1 | Bartosz | Piasecki | REA3014 | Informasjonsteknologi 1 | 2 |
| 1 | Bartosz | Piasecki | REA3026 | Matematikk S1 | 6 |
| 2 | Tommy | Lund | REA3026 | Matematikk S1 | 3 |

|  |  |  |
| --- | --- | --- |
| ElevID | Fornavn | Etternavn |
| 1 | Bartosz | Piasecki |
| 2 | Tommy | Lund |

|  |  |
| --- | --- |
| Fagkode | Fagnavn |
| REA3014 | Informasjonsteknologi 1 |
| REA3026 | Matematikk S1 |

Mange til mange relasjoner lag en sammenføyingstabell.

Alternativ 1: lage en sammensatt primærnøkkel

|  |  |  |
| --- | --- | --- |
| ElevID | Fagkode | Karakter |
| 1 | REA3014 | 2 |
| 1 | REA3026 | 6 |
| 2 | REA3026 | 3 |
| 1 | REA3014 | 3\* |

\*Ikke mulig å registrere denne kombinasjonen av elev og fag kode er allerede registrert fra før av.

Alternativ 2: lag en egen primærnøkkel for sammenføyingstabell

|  |  |  |  |
| --- | --- | --- | --- |
| KarakterID | ElevID | Fagkode | Karakter |
| 1 | 1 | REA3014 | 2 |
| 2 | 1 | REA3026 | 6 |
| 3 | 2 | REA3026 | 3 |
| 4 | 1 | REA3014 | 4 |

Her vil det være mulig å registrere flere karakterer til samme person i samme fag.

Eksempel (fjelloversikt)

En person kan bestige mange fjell, mens ett fjell kan bestiges av mange personer

|  |  |  |
| --- | --- | --- |
| PersonID | Fornavn | Etternavn |
| 1 | Bartosz | Piasecki |
| 2 | Tommy | Lund |

|  |  |  |
| --- | --- | --- |
| FjellID | Fjellnavn | Moh. |
| 1 | Galdhøpiggen | 2469 |
| 2 | Kilimanjaro | 5895 |

Dette er en mange til mange relasjoner! Lag en sammenføyingstabell!

Alternativ 1: Lag en sammensatt primærnøkkel

|  |  |  |
| --- | --- | --- |
| PersonID | FjellID | Dato |
| 1 | 1 | 10.01.2019 |
| 1 | 2 | 11.01.2019 |
| 2 | 1 | 05.05.1995 |
| 1 | 1 | 17.01.2019\* |

\*Ikke mulig å registrere samme person på samme fjell

Alternativ 2: Lag en egen primærnøkkel for sammensføyingstabellen

|  |  |  |  |
| --- | --- | --- | --- |
| TurID | PersonID | FjellID | Dato |
| 1 | 1 | 1 | 10.01.2019 |
| 2 | 1 | 2 | 11.01.2019 |
| 3 | 2 | 1 | 05.05.2019 |
| 4 | 1 | 1 | 17.01.2019 |

Vil være mulig å registrere samme person på samme fjell flere ganger!!!