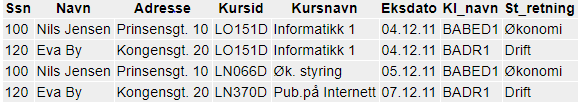


**TABELL TIL OPPGAVEN:**



**Oppgave 1)**

1. **Hvilken normalform (1NF, 2NF, 3NF, BCNF) tilfredsstiller relasjonen (tabellen) under? Begrunn svaret ut fra hvilke forutsetninger du setter.**

* **Tabellen tilfredsstiller 1NF** fordi den har unik ID *(Antar at ‘Ssn’ er primærnøkkel*), den har «single-valued» celler og den har samme datatype på attributtene.
* Tabellen tilfredsstiller ikke **2NF** fordi attributtene er ikke avhengige av ID
* Tabellen tilfredsstiller ikke **3NF** fordi alle kolonnene kan ikke bli fast bestemt av en nøkkel.
* Tabellen tilfredsstiller ikke **BCNF**, fordi det er determinanter som ikke er kandidatnøkler for relasjonen. For eksempel «St\_retning» og «Kl\_navn»
* Tabellen tilfredsstiller ikke kravene til **4NF** fordi det er flere attributt-duplikater.

1. **Forklar kort hvilke (praktiske) konkrete ulemper (problemer) som kan oppstå ved at relasjonen ikke er normalisert.**Det kan oppstå problemer ved at relasjonen ikke er normalisert. Ved å normalisere en database prøver man ofte å oppnå disse målene;

* Arrangere data i logiske grupperinger slik at hver gruppe beskriver en liten del av hele systemet.
* Minimere antall duplikater i systemet.
* Organisere data slik at når du skal modifisere data, må man bare forandre på dataen i

en plass.

* Lage et system som har god flyt og man kan endre og modifisere uten å komprimere datainnholdet på noen måte.

1. **Normaliser relasjonen slik at BCNF er tilfredsstilt. Begrunn svaret ved å forklare prosessen frem til et normalisert resultat (dvs. det holder ikke å bare vise sluttresultatet).**

* **Fjerner unødvendig kolonne «Ssn»**
* **Legger til kolonne Stud ID {PK}**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Stud-ID** | **Navn** | **Adresse** | **Kursid** | **Kursnavn** | **Eksdato** | **Kl\_navn** | **St\_retning** |
| nils\_jensen | Nils Jensen | Prinsensgt.10 | LO151D | Informatikk 1 | 04.12.11 | BABED1 | Økonomi |
| eva\_by | Eva By | Kongensgt.20 | LO151D | Informatikk 1 | 04.12.11 | BADR1 | Drift |
| nils\_jensen | Nils Jensen | Prinsensgt.10 | LN066D | Øk. styring | 05.12.11 | BABED1 | Økonomi |
| eva\_by | Eva By | Kongensgt.20 | LN066D | Informatikk 1 | 07.12.11 | BADR1 | Drift |

* Tabellen tilfredsstiller nå 1NF.
* **Videre fjerner jeg alle attributtene som ikke er direkte avhengig av Primærnøkkelen**
* **Splitter opp i to tabeller der alle attributtene er avhengige av Primærnøkkel.**
* **Legger til en 3. tabell som bare består av Primærnøklene, som binder de to første tabellene. (Stud 🡪 Kurs).**

|  |  |  |
| --- | --- | --- |
| **Kurs-ID (PK)** | **Kursnavn** | **Eksdato** |
| LO151D | Informatikk 1 | 04.12.11 |
| LN066D | Øk. styring | 05.12.11 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stud-ID (PK)** | **Navn** | **Adresse** | **St.retning** | **Kl\_navn** |
| nils\_jensen | Nils Jensen | Prinsensgt.10 | Økonomi | BABED1 |
| eva\_by | Eva By | Kongensgt.20 | Drift | BADR1 |

|  |  |
| --- | --- |
| **Stud-ID (PK)** | **Kurs-ID (PK)** |
| nils\_jensen | LO151D |
| eva\_by | LN066D |

* Tabellen tilfredsstiller nå 2NF.
* Nå er det ingen duplikat-kolonner eller noen data som viser til flere kolonner. Videre fjernes alle kolonner som ikke er HELT avhengig av primærnøkkel.

|  |  |  |
| --- | --- | --- |
| **Kurs-ID (PK)** | **Kursnavn** | **Eksdato** |
| LO151D | Informatikk 1 | 04.12.11 |
| LN066D | Øk. styring | 05.12.11 |

|  |  |
| --- | --- |
| **Stud-ID (PK)** | **Kurs-ID (PK)** |
| nils\_jensen | LO151D |
| eva\_by | LN066D |

|  |  |  |  |
| --- | --- | --- | --- |
| **Stud-ID (PK)** | **Navn** | **Adresse** | **Kl\_navn (FK)** |
| nils\_jensen | Nils Jensen | Prinsensgt.10 | BABED1 |
| eva\_by | Eva By | Kongensgt.20 | BADR1 |

|  |  |
| --- | --- |
| **Kl\_navn (PK)** | **St.retning** |
| BABED1 | Økonomi |
| BADR1 | Drift |

* Nå er 3NF tilfredsstilt.
* Fjerner nå relasjoner hvor determinanten ikke er kandidatnøkkel, og putter dem i en egen tabell.

|  |  |  |
| --- | --- | --- |
| **Kurs-ID (PK)** | **Kursnavn** | **Eksdato** |
| LO151D | Informatikk 1 | 04.12.11 |
| LN066D | Øk. styring | 05.12.11 |

|  |  |
| --- | --- |
| **Stud-ID (PK)** | **Kurs-ID (PK)** |
| nils\_jensen | LO151D |
| eva\_by | LN066D |

|  |  |
| --- | --- |
| **Stud-ID (PK)** | **Navn** |
| nils\_jensen | Nils Jensen |
| eva\_by | Eva By |

|  |  |
| --- | --- |
| **Kl\_navn (PK)** | **St.retning** |
| BABED1 | Økonomi |
| BADR1 | Drift |

|  |  |
| --- | --- |
| **Kurs\_ID (PK)** | **Kl\_navn (PK)** |
| LO151D | BABED1 |
| LN066D | BADR1 |

|  |  |
| --- | --- |
| **Stud-ID (PK)** | **Adresse** |
| nils\_jensen | Prinsensgt.10 |
| eva\_by | Kongensgt.20 |

* Nå er BCNF tilfredsstilt.

**Oppgave 2)**

Kandidatnøkler:

* kunde\_id (INT)
* kunde\_navn (VAR)
* kunde\_adresse (VAR)
* kunde\_tfl (INT)
* eiendoms\_id (INT)
* eiendoms\_adresse (VAR)
* eier\_id (INT)
* eier\_navn (VAR)
* eier\_adresse (VAR)
* eier\_tlf (INT)
* leie\_pris (INT)
* leie\_fra\_uke (INT)
* leie\_til\_uke (INT)

***Tabellen er ikke problemfri mht registrering og sletting av data***🡪 Dette er fordi:

* I denne tabellen vil det blir mye dobbeltlagring av data. Hvis vi for eksempel skal oppdatere ‘eier\_adresse’ til en eier, vil det kun bli oppdatert for en rad. Det vil si at hvis man ikke manuelt har kontrollen på alle stedene en bestemt data er lagret vil det oppstå logiske inkonsekvenser.
* Vi kan ikke sette inn nye kunder hvis dem ikke har leid noe enda, hvis for eksmpel leie\_fra\_uke og leie\_til\_uke ikke kan være null.
* Hvis vi vil slette for eksempel leie\_fra og leie\_til fordi denne personen ikke lenger leier hytten, må vi slette hele kundedataen, med mindre vi setter leie\_til og leie\_fra til null, derfor er dette en «deletion anomaly».