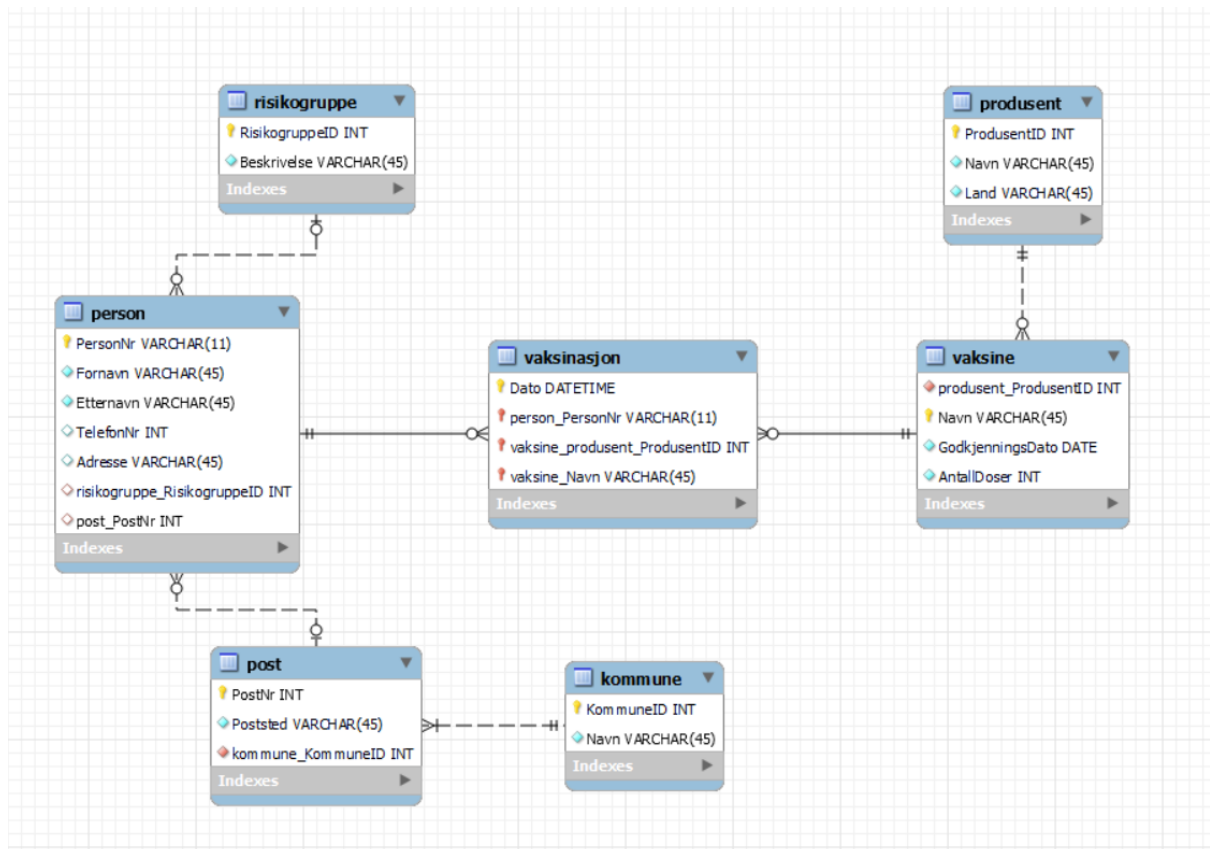


Oppgave 1 – Datamodellering



Oppgave 2 – Fra datamodell til database

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----
-- Schema oblig_3
-- -----

-- -----
-- Schema oblig_3
-- -----

CREATE SCHEMA IF NOT EXISTS `oblig_3` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci ;
USE `oblig_3` ;

-- -----
-- Table `oblig_3`.`kommune`
-- -----

CREATE TABLE IF NOT EXISTS `oblig_3`.`kommune` (
  `KommuneID` INT NOT NULL,
  `Navn` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`KommuneID`))
ENGINE = InnoDB;

-- -----
-- Table `oblig_3`.`post`
-- -----

CREATE TABLE IF NOT EXISTS `oblig_3`.`post` (
  `PostNr` INT NOT NULL,
  `Poststed` VARCHAR(45) NOT NULL,
  `kommune_KommuneID` INT NOT NULL,
  PRIMARY KEY (`PostNr`),
  INDEX `fk_post_kommune_idx` (`kommune_KommuneID` ASC) VISIBLE,
  CONSTRAINT `fk_post_kommune`
    FOREIGN KEY (`kommune_KommuneID`)
      REFERENCES `oblig_3`.`kommune` (`KommuneID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `oblig_3`.`risikogruppe`
-- -----

CREATE TABLE IF NOT EXISTS `oblig_3`.`risikogruppe` (
  `RisikogruppeID` INT NOT NULL,
  `Beskrivelse` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`RisikogruppeID`))
```

```
ENGINE = InnoDB;
```

```
-- -----  
-- Table `oblig_3`.`produsent`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `oblig_3`.`produsent` (  
  `ProdusentID` INT NOT NULL,  
  `Navn` VARCHAR(45) NOT NULL,  
  `Land` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`ProdusentID`))
```

```
ENGINE = InnoDB;
```

```
-- -----  
-- Table `oblig_3`.`person`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `oblig_3`.`person` (  
  `PersonNr` INT NOT NULL,  
  `Fornavn` VARCHAR(45) NOT NULL,  
  `Etternavn` VARCHAR(45) NOT NULL,  
  `TelefonNr` INT NULL,  
  `Adresse` VARCHAR(45) NULL,  
  `risikogruppe_RisikogruppeID` INT NULL,  
  `post_PostNr` INT NULL,  
  PRIMARY KEY (`PersonNr`),  
  INDEX `fk_person_risikogruppel_idx` (`risikogruppe_RisikogruppeID` ASC)  
VISIBLE,  
  INDEX `fk_person_post1_idx` (`post_PostNr` ASC) VISIBLE,  
  CONSTRAINT `fk_person_risikogruppel`  
    FOREIGN KEY (`risikogruppe_RisikogruppeID`)  
    REFERENCES `oblig_3`.`risikogruppe` (`RisikogruppeID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_person_post1`  
    FOREIGN KEY (`post_PostNr`)  
    REFERENCES `oblig_3`.`post` (`PostNr`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;
```

```
-- -----  
-- Table `oblig_3`.`vaksine`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `oblig_3`.`vaksine` (  
  `Produsent` INT NOT NULL,  
  `Navn` VARCHAR(45) NOT NULL,  
  `GodkjenningsDato` DATE NOT NULL,  
  `AntallDoser` INT NOT NULL,  
  `produsent_ProdusentID` INT NOT NULL,  
  PRIMARY KEY (`Produsent`, `Navn`),  
  INDEX `fk_vaksine_produsent1_idx` (`produsent_ProdusentID` ASC) VISIBLE,  
  CONSTRAINT `fk_vaksine_produsent1`  
    FOREIGN KEY (`produsent_ProdusentID`)  
    REFERENCES `oblig_3`.`produsent` (`ProdusentID`)  
    ON DELETE NO ACTION
```

```
        ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `oblig_3`.`vaksinasjon`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `oblig_3`.`vaksinasjon` (
  `Dato` DATETIME NOT NULL,
  `PersonNr` INT NOT NULL,
  `vaksine_Produsent` INT NOT NULL,
  `vaksine_Navn` VARCHAR(45) NOT NULL,
  `person_PersonNr` INT NOT NULL,
  PRIMARY KEY (`Dato`, `vaksine_Produsent`, `vaksine_Navn`,
`person_PersonNr`, `PersonNr`),
  INDEX `fk_vaksinasjon_vaksine1_idx` (`vaksine_Produsent` ASC,
`vaksine_Navn` ASC) VISIBLE,
  INDEX `fk_vaksinasjon_person1_idx` (`person_PersonNr` ASC) VISIBLE,
  CONSTRAINT `fk_vaksinasjon_vaksine1`
    FOREIGN KEY (`vaksine_Produsent`, `vaksine_Navn`)
    REFERENCES `oblig_3`.`vaksine` (`Produsent`, `Navn`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_vaksinasjon_person1`
    FOREIGN KEY (`person_PersonNr`)
    REFERENCES `oblig_3`.`person` (`PersonNr`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Oppgave 3 – SQL

A - Hvor mange personer har blitt vaksinert(altså mottatt minst en dose) i hver kommune?

```
SELECT kommune.navn, COUNT(distinct person_PersonNr) as vaksinert_mengde
FROM oblig_3.vaksinasjon
INNER JOIN person ON person.PersonNr = vaksinasjon.person_PersonNr
INNER JOIN post ON post.PostNr = person.post_PostNr
INNER JOIN kommune ON kommune.KommuneID = post.kommune_KommuneID
GROUP BY kommune.Navn;
```

B - Hvem(altså fullt navn) har blitt vaksinert?

```
SELECT concat(person.Fornavn, ' ', person.Etternavn) as FulltNavn FROM
oblig_3.vaksinasjon
INNER JOIN person ON person.PersonNr = vaksinasjon.person_PersonNr
GROUP BY PersonNr;
```

C - Hvilken vaksine har flest mottatt, og hvor mange doser av denne er satt?

View:

```
CREATE VIEW oblig_3.vaksiner_per_vaksine AS
SELECT
    vaksine.Navn, COUNT(vaksine_producent_ProducentID) AS
    antall_doser_satt
FROM
    oblig_3.vaksine
INNER JOIN
    vaksinasjon ON (vaksinasjon.vaksine_producent_ProducentID =
    vaksine.producent_ProducentID AND vaksinasjon.vaksine_Navn = vaksine.Navn)
GROUP BY
    vaksine_producent_ProducentID
```

SQL:

```
SELECT * FROM oblig_3.vaksiner_per_vaksine
WHERE antall_doser_satt = (SELECT MAX(antall_doser_satt) FROM
oblig_3.vaksiner_per_vaksine);
```

D - Hvilke kommuner har vi en oversikt over?

```
SELECT Navn as Kommune FROM oblig_3.kommune;
```

E - Hvem har fått vaksine av den typen det er gitt flest doser av?

View:

```
CREATE VIEW oblig_3.vaksiner_per_vaksine AS
    SELECT
        vaksine.Navn, COUNT(vaksine_produzent_ProduzentID) AS
        antall_doser_satt
    FROM
        oblig_3.vaksine
    INNER JOIN
        vaksinasjon ON (vaksinasjon.vaksine_produzent_ProduzentID =
        vaksine.produzent_ProduzentID AND vaksinasjon.vaksine_Navn = vaksine.Navn)
    GROUP BY
        vaksine_produzent_ProduzentID
```

SQL:

```
SELECT person.Fornavn, person.Etternavn, vaksinasjon.vaksine_Navn
FROM oblig_3.person
INNER JOIN vaksinasjon ON vaksinasjon.person_PersonNr = person.PersonNr
WHERE vaksinasjon.vaksine_Navn IN (SELECT Navn FROM
oblig_3.vaksiner_per_vaksine WHERE antall_doser_satt = (SELECT
MAX(antall_doser_satt) FROM oblig_3.vaksiner_per_vaksine));
```

F - Hvilke personer har fått 2 eller flere doser, og når ble disse dosene satt?

```
SELECT vaksinasjon.Dato, concat(person.Fornavn, ' ', person.Etternavn) as
Navn FROM oblig_3.vaksinasjon
INNER JOIN person on person.PersonNr = vaksinasjon.person_PersonNr
WHERE person_PersonNr IN (
    SELECT person_PersonNr FROM oblig_3.vaksinasjon
    GROUP BY person_PersonNr
    HAVING count(person_PersonNr) > 1
);
```

G - Hvor mange personer i hver risikogruppe er fullvaksinert(dvs. har minst to doser?)

```
SELECT Risikogruppe.beskrivelse as Risikogruppe, count(distinct PersonNr)
as Fullvaksinerte FROM oblig_3.vaksinasjon
INNER JOIN person on person.PersonNr = vaksinasjon.person_PersonNr
INNER JOIN risikogruppe on risikogruppe.RisikogruppeID =
person.risikogruppe_RisikogruppeID
WHERE person_PersonNr IN (
    SELECT person_PersonNr FROM oblig_3.vaksinasjon
    GROUP BY person_PersonNr
    HAVING count(person_PersonNr) > 1
)
GROUP BY risikogruppe.RisikogruppeID;
```