

Innhold

1 Oppgaven	1
1.1 a) Opprette tabeller	1
1.1.1 Eier	1
1.1.2 Kjøretøy	1
1.1.3 Kjøretøy eier	2
1.1.4 Bompasering	2
1.1.5 Bomstasjon	2
1.2 b) Legge inn test data	2
1.3 c) Dropp telefonnummer	2
1.4 d) Håndtere passering med manglende skilt	3
1.5 e) List all informasjon	3
1.6 f) List all informasjon der det er registrert registreringsnummer	3
1.7 g) Nå med relasjonsalgebra	3
1.8 h) Vis antall passeringer på ulike registreringsnummer	3
1.9 i) Vis siste passering med registreringsnummer AA100000	3
1.10 j) Vis antall passeringer uten registreringsnummer	3
1.11 k) Kan vi håndtere utenlandske registreringsnummer?	3
1.12 l) Oppgaven ønsker egentlig 2 tabeller, hvilke andre ønsker vi?	3

1 Oppgaven

1.1 a) Opprette tabeller

For å oppfylle kravene oppgitt i oppgaven om å lagre informasjon om bilene og eierne, ble det vurdert å lage 5 tabeller. Nedenfor er SQL-koden for å opprette tabellene med forklaring.

1.1.1 Eier

Vi oppretter tabellen **eier** for å lagre informasjon om eierne, inkludert navn, adresse, telefonnummer og e-postadresse. Hver eier får en unik ID som primær-nøkkel.

```
CREATE TABLE IF NOT EXISTS eier (  
    id SERIAL PRIMARY KEY,  
    navn VARCHAR(255) NOT NULL,  
    adresse VARCHAR(255) NOT NULL,  
    telefon VARCHAR(20),  
    epost VARCHAR(255)  
);
```

1.1.2 Kjøretøy

Deretter oppretter vi tabellen **kjøretøy** for å lagre informasjon om kjøretøyene, som for eksempel registreringsnummer. Hver kjøretøy får en unik ID som pri-

mærknøkkel. Tanken med denne tabellen er å kunne lagre flere kjøretøy for hver eier, og dermed håndtere situasjoner der en eier har flere biler.

```
CREATE TABLE IF NOT EXISTS kjoretoy (  
    id SERIAL PRIMARY KEY,  
    registreringsnummer VARCHAR(20) UNIQUE NOT NULL  
);
```

1.1.3 Kjøretøy eier

kjoretoy_eier tabellen bruker vi som en mellomtabell for å håndtere forholdet mellom kjøretøy og eiere, vi tenkte at en eier kan ha flere kjøretøy.

```
CREATE TABLE IF NOT EXISTS kjoretoy_eier (  
    kjoretoyid INTEGER,  
    eierid INTEGER,  
    FOREIGN KEY (kjoretoyid) REFERENCES kjoretoy(id),  
    FOREIGN KEY (eierid) REFERENCES eier(id)  
);
```

1.1.4 Bompassering

I **bompassering** tabellen lagrer vi informasjon om hver passering, inkludert tidspunktet for passeringen, kjøretøyets ID og bomstasjonens ID. Vi bruker fremmednøkler for å referere til kjøretøy og bomstasjoner.

```
CREATE TABLE IF NOT EXISTS bompassering (  
    id SERIAL PRIMARY KEY,  
    passeringstid TIMESTAMP NOT NULL  
  
    kjoretoyid INTEGER,  
    bomstasjonid INTEGER,  
    FOREIGN KEY (kjoretoyid) REFERENCES kjoretoy(id),  
    FOREIGN KEY (bomstasjonid) REFERENCES bomstasjon(id),  
);
```

1.1.5 Bomstasjon

bomstasjon tabellen lagrer informasjon om bomstasjonene, inkludert navn og plassering.

```
CREATE TABLE IF NOT EXISTS bomstasjon (  
    id SERIAL PRIMARY KEY,  
    navn VARCHAR(255) NOT NULL,  
    plassering VARCHAR(255) NOT NULL  
);
```

1.2 b) Legge inn test data

1.3 c) Dropp telefonnummer

Hvis det ikke er behov å lagre telefonnummeret i eier tabellen, så er det mulig å fjerne det på denne måten.

- 1.4 d) Håndtere passering med manglende skilt
- 1.5 e) List all informasjon
- 1.6 f) List all informasjon der det er registrert registreringsnummer
- 1.7 g) Nå med relasjonsalgebra
- 1.8 h) Vis antall passeringer på ulike registreringsnummer
- 1.9 i) Vis siste passering med registreringsnummer AA100000
- 1.10 j) Vis antall passeringer uten registreringsnummer
- 1.11 k) Kan vi håndtere utenlandske registreringsnummer?
- 1.12 l) Oppgaven ønsker egentlig 2 tabeller, hvilke andre ønsker vi?

```
ALTER TABLE eier DROP COLUMN telefon;
```