Eksamen 2021

230

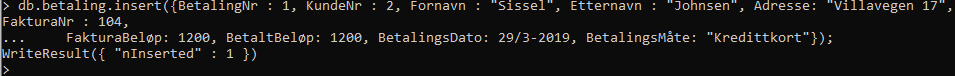
**Oppgave 1.** use Eksamen2021; db.createUser(){user: «root», pwd : «123», roles: [«readWrite», «dbAdmin»]}; db.createCollection(‘betaling’)



**Oppgave 2.**

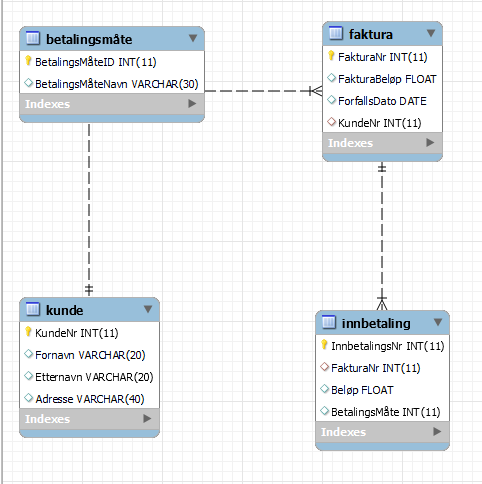
db.betaling.insert({BetalingNr : 1, KundeNr : 2, Fornavn : "Sissel", Etternavn : "Johnsen", Adresse: "Villavegen 17", FakturaNr : 104,

    FakturaBeløp: 1200, BetaltBeløp: {1200}, BetalingsDato: 29/3-2019, BetalingsMåte: "Kredittkort"});



**Oppgave 3.** db.betaling.insert({BetalingNr : 1, KundeNr : 2, Fornavn : "Sissel", Etternavn : "Johnsen", Adresse: "Villavegen 17", FakturaNr : 104,

FakturaBeløp: 1200, Betalinger : [{400, 120, 300}], BetalingsDato: 29/3-2019, BetalingsMåte: "Kredittkort"});

**Oppgave 5.** 

**Oppgave 7.**

create database Eksamen2021;

use Eksamen2021;

CREATE TABLE IF NOT exists Kunde(

KundeNr INT,

Fornavn VARCHAR(20),

Etternavn varchar(20),

Adresse varchar(40),

constraint primary key(KundeNr)

)Engine = INNODB;

CREATE TABLE IF NOT exists Faktura(

FakturaNr INT(11),

FakturaBelop float(20),

BetaltBelop float(20),

ForfallsDato DATE,

BetaltDato DATE,

KundeNr INT,

CONSTRAINT Foreign Key(KundeNr) REFERENCES Kunde(KundeNr),

constraint Primary key(FakturaNr)

)Engine = INNODB;

create table if not exists Innbetaling(

InnbetalingsNr INT,

FakturaNr INT,

Belop float,

BetalingsMåte INT,

constraint Primary key(InnbetalingsNr),

Constraint foreign key(FakturaNr) references Faktura(FakturaNr)

)Engine = INNODB;

create table if not exists BetalingsMåte(

BetalingsMåteID INT,

BetalingsMåteNavn Varchar(30),

constraint primary key (BetalingsMåteID)

)Engine = INNODB;

DELIMITER $$

CREATE TRIGGER Innbetalt

AFTER INSERT

ON Innbetaling For Each row

BEGIN

UPDATE Faktura F

set betaltbelop = betaltbelop + new.belop

where FakturaNr = new.FakturaNr;

END$$

CREATE TRIGGER InnbetaltDato

AFTER INSERT

ON Innbetaling For Each row

BEGIN

UPDATE Faktura F

set betaltdato = curdate()

where FakturaNr = new.FakturaNr;

END$$

DELIMITER ;

insert into betalingsmåte values

(1, "Visa"),

(2, "Paypal"),

(3, "Wire"),

(4, "Cash");

Insert into Kunde values

(1, "Kari", "Andersen", "Krokvegen 11"),

(2, "John", "Smith", "Alleredegate 42"),

(3, "Josef", "Slakk", "Videospillstien 11");

Insert into Faktura values

(1, 120, 0, curdate(), NULL, 1),

(2, 499, 0, curdate(), NULL ,3);

insert into innbetaling values

(1, 2, 499, 2),

(2, 1, 120, 3);

**Oppgave 8.** Select K.Fornavn, K.etternavn, F.FakturaBelop, F.Forfallsdato from kunde K Join Faktura F where f.kundenr = k.kundenr;

**Oppgave 9.** Select K.Fornavn, K.Etternavn, (select count(\*) from faktura F where f.kundenr = k.kundenr) as AntallFakturaer, F.fakturabelop as Totalbelop From Kunde K, Faktura F;

**Oppgave 10.**

select K.Fornavn, K.Etternavn, F.FakturaNr from Kunde K, Faktura F where (F.betaltbelop < f.fakturabelop AND (DATEDIFF(F.FakturaDato, F.BetaltDato) < 7))

**Oppgave 11.** Select K.Fornavn, K.Etternavn, F.FakturaNr from Kunde K, Faktura F where (F.betaltbelop < f.fakturabelop);

**Oppgave 12.** Transaksjoner er funksjoner som kan låse variabler i elementer, så da ikke disse variablene kan endres før transaksjonen er ferdig og låser de opp igjen. Det er kjekt for operasjoner som er avhegig av visse verdier, så de kan da låses så ingen andre brukere på databasen kan endre de.

**Oppgave 13.** index.php:

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<?php

      $Vert="localhost";

      $Bruker="root";

      $Passord="123";

      $Database="Eksamen2021";

      $mysqli = new mysqli($Vert, $Bruker, $Passord, $Database);

      if (mysqli\_connect\_errno()) {

           printf( "Ingen forbindelse : %s", mysqli\_connect\_error() );

           exit();

       }

       $setning = "SELECT \* FROM Kunde";

    //echo $setning;

    $resultat = $mysqli->query($setning);

    $rad = $resultat->fetch\_row();

    if ($rad == null)

    echo '<meta http-equiv="refresh" content="0;URL=index.php">';

    echo '<table>';

    echo '<caption><h3>Kunder i Eksamen2021</h3></caption>';

    echo '<tr><th>KundeNr</th><th>Fornavn</th><th>Etternavn</th><th>Adresse</th>

          do {

                echo '<tr><td>'.$rad[0].'</td>';  //KundeNr

                echo '<td>'.$rad[1].'</td>';         //Fornavn

                echo '<td>'.$rad[2].'</td>';         //Etternavn

                echo '<td>'.$rad[3].'</td></tr>';//Adresse

           }  while ($rad = $resultat->fetch\_row());//tilordning, tester om $rad==0

    echo '</table>';

    $mysqli->close();

?>

</body>

</html>

**Oppgave 14.** Protokoller som kan kommunisere med nettet over flere lag/nivåer i applikasjonslagene. s

**Oppgave 15.** Multicast og anycast er to andre adressetyper.

**Oppgave 16.** a) Det betyr at adresser som skal være gyldige Unique-local må de 7 først bitene matche fc00 sine første 7 bits, derfor /7 på slutten.

b) i og iii er gyldige.

**Oppgave 17**. a) ii og iv.

b) 192.168

**Oppgave 18**. fd10::11:aa00

**Oppgave 19.** fd10:0020:0000:000c:0000:0004:0000:0000

20. 1.Bruke brannmur på PCen. Brannmuren kontrollerer hvem programmer som kan koble seg inn og ut av nettverket på PCen.