KANDIDAT

5505

PRØVE

# IS-201 1 Datamodellering og databasesystemer

| | |
|---|---|
| Emnekode | IS-201 |
| Vurderingsform | Skriftlig eksamen |
| Starttid | 01.12.2016 09:00 |
| Sluttid | 01.12.2016 13:00 |
| Sensurfrist | 22.12.2016 01:00 |
| PDF opprettet | 28.08.2018 10:05 |
| Opprettet av | Digital Eksamen |

## 1 IS-201, Front Page

**Emnekode**: IS-201
**Emnenavn**: Datamodellering og databasesystemer

**Dato:** 1. desember
**Varighet:** 4 timer

**Tillatte hjelpemidler:** Ordbøker: (engelsk/norsk)

**Merknader:**
Alle oppgaver skal besvares.
Alle hovedspørsmål (1.-4.) må besvares til bestått (karakter E eller bedre) for at eksamen skal regnes som bestått.

----------------------------

Det forekommer av og til spørsmål om bruk av eksamensbesvarelser til undervisnings- og læringsformål. Universitetet trenger kandidatens tillatelse til at besvarelsen kan benyttes til dette. Besvarelsen vil være anonym.

**Tillater du at din eksamensbesvarelse blir brukt til slikt formål?**

[✓ Ja] ✓

Nei

Riktig. 0 av 0 poeng.

## 2 Basic database concepts (20%)

**Problem 1. Basic database concepts**

a) Explain the difference between DROP TABLE, ALTER TABLE, and DELETE statements. Illustrate your answer with an example.
b) What is unary relationship? Illustrate your answer with an example.
c) What is the difference between Primary Key and Foreign Key? Illustrate your answer with an example.

**Skriv ditt svar her...**

a) The `DROP TABLE` statement is used in case you want to remove a table and its contents from a database. For example, you can remove a table called `users` by running the following statement `DROP TABLE users`.

The `ALTER TABLE` is used to edit a table, for example if you want to change its name, or if you want to add a primary key. A specific example could be `ALTER TABLE users ADD PRIMARY KEY (userID).

`DELETE` statements are used to delete a specific value in a table, for example one particular user. An example of a DELETE statement is `DELETE FROM users WHERE userID = 12`

b) A unary relationship is a relationship in a database where for exampel two tables are connected, lets say we have two tables, `Account`, and `Site`. In the `Account` table, we have a foreign key referring to `Site`'s primary key, because this `Account` belongs to this specific site. Also, the `Site` table has a foreign key referring to `Account`'s primary key, because the site has an `Account` as Administrator. These tables have a unary relationship.

c) A primary key is an attribute which is automatically indexed in the database. It is used to identify a specific row in a table. An example of this could be userID. A primary key is always a unique value in its own table. A foreign key, on the other hand, is used as a reference to a specific row, usually in a different table, to create a connection between the two tables (unless the primary and foreign keys are in the same table, that is). It is not indexed automatically. Lets say we have a database for a gym, where a member has a personal trainer. We have the tables `member` and `personalTrainer`. Here, we can say that both tables have a primary key, `memberID` for `member`, and `ptID` for `personalTrainer`. Usually, there are some additional attributes like name, but to keep it simple, we will skip those for now. Well, to create a connection between a member and his/her personal trainer, we have to connect the two tables somehow. This is where foreign keys come to play. If we put an attribute called `personalTrainerID`, with the same datatype and length as the ptID in `personalTrainer`, inside the `member` table, and set this as a foreign key by running the statement `CONSTRAINT fk_member_pt FOREIGN KEY (personalTrainerID) REFERENCES personalTrainer (ptID)`, we have a connection between the two tables. This way, a personal trainer can have multiple connections to members, while a member can only have one specific personal trainer.

Besvart.

## 3 E-R modeling (20%)

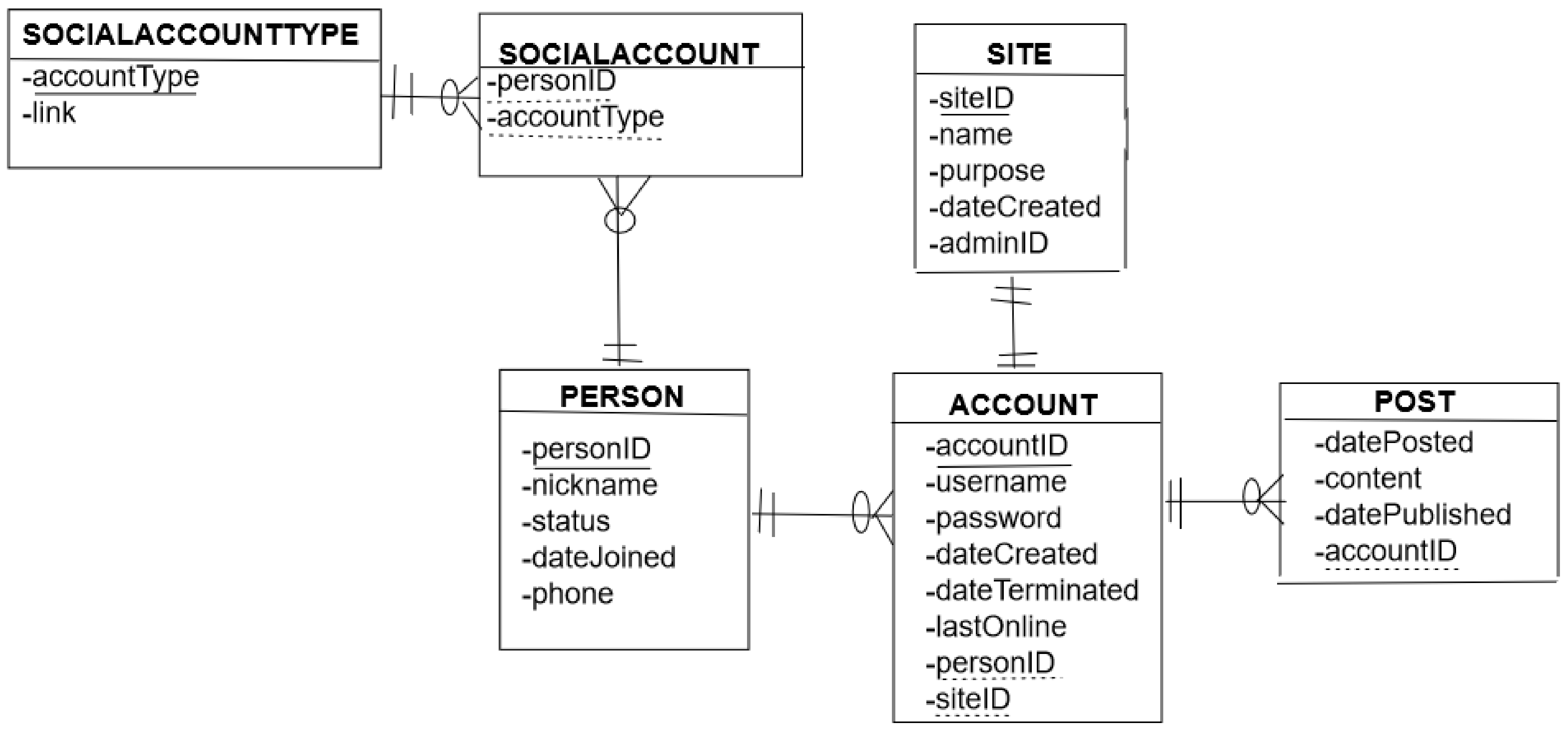
**Problem 2. E-R modeling**

Read the task carefully. Then draw an E-R diagram for the case study.

Virtual campus (VC) is a social media firm that specializes in creating virtual meeting places for students, faculty, staff, and others associated with different college campuses.VC was started as a student project in a database class at Cyber University, and online polytechnic college, with headquarters in a research part in Dayton, Ohio. The following pars of this exercise relate to different phases in the development of the database VC now provides to client institutions to support a threaded discussion application. Your assignment is to draw an E-R diagram to represent the initial phase of the development, described by the following:

a. A client may maintain several social media sites (e.g., for intercollegiate sports, academics, local food and beverage outlets, or a specific student organization). Each site has attributes of Site Identifier, Site Name, Site Purpose, Site Administrator, and Site Creation Date.
b. Any person may become a participant in any public site. Persons need to register with the client's social media presence to participate in any site, and when they do the person is assigned a Person Identifier; the person provides his or her Nickname and Status(e.g. student, faculty, staff, or friend, or possibly several such values); the Date Joined the site is automatically generated. A person may also include other information, which is available to other persons on the site; this information includes Name, Twitter Handel, Facebook Page Link, and SMS Contact Number. Anyone may register (no official association with the client is necessary).
c. An account created each time a person registers to use a particular site. An account is described by an Account ID, User Name, Password, Date Created, Date Terminated, and Date/Time the person most recently used that account.
d. Using an account, a person creates a posting, or message, for others to read. A posting has a Posting Date/Time and Content. The person posting the message may also add a Date when the posting should be made invisible to other users.
e. A person is permitted to have multiple accounts, each of which is for only one site.
f. A person, over time, may create multiple postings from an account.

**Skriv ditt svar her...**



Besvart.

## 4 Normalization 20%

**Problem 3. Normalization**

A pet store currently uses a legacy flat file system to store all of its information. The owner of the store, Peter Corona, wants to implement a Web-enable database application. This would enable branch stores to enter data regarding inventory levels, ordering, and so on. Presently, the data for inventory and sales tracking are stored in one file that has the following format:

------------------------------------------------------------------

StoreName, PetName, Pet Description, Price, Cost,
SupplierName, ShippingTime, QuantityOnHand,
DateOfLastDelivery, DateOfLastPurchase,
DeliveryDate1, DeliveryDate2, DeliveryDate3,
DeliveryDate4, PurchaseDate1, PurchaseDate2,
PurchaseDate3, PurchaseDate4, LastCustomerName,
CustomerName1, CustomerName2, CustomerName3,
CustomerName4

------------------------------------------------------------------

Assume that you want to track all purchase and inventory data, such as who bought the fish, the date that it was purchased, the date that it was delivered, and so on. The present file format allows only the tracking of the last purchase and delivery as well as four prior purchases and deliveries. You can assume that a type of fish is supplied by one supplier.

a. Show all functional dependencies.
b. What normal form is this table in?
c. Design a normalized model (RM) for these data. Show that it is in 3NF.

**Skriv ditt svar her...**

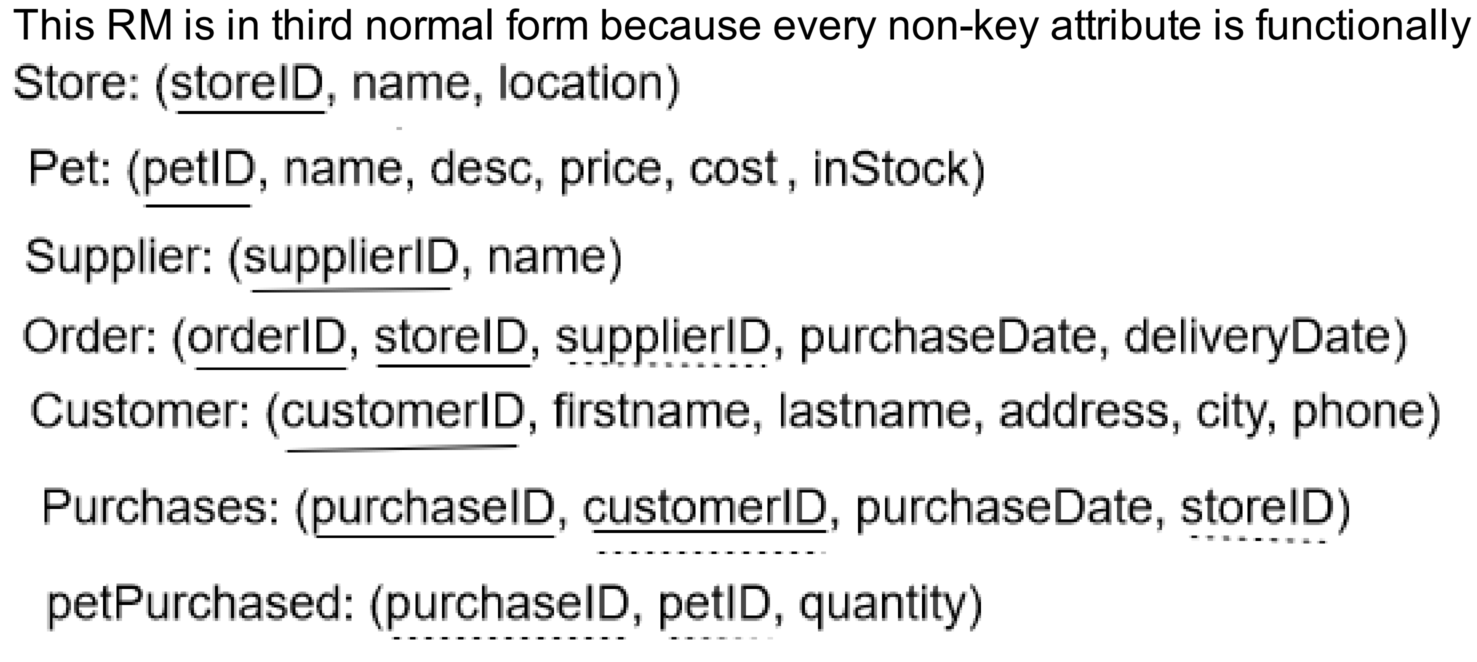
As i understand it, a customer enters the store to buy a fish (pet), which is the purchaseDate. The store then orders a pet from a supplier, which is the dateOfLastPurchase

a)
Due to the fact that the "table" does not have a primary key, it shouldn't be possible for it to be functionally dependent.
PetDescription, Price, and Cost is dependent on PetName.
QuantityOnHand, ShippingTime is dependent on SupplierName
DateOfLastDelivery is dependent on DateOfLastPurchase,
DeliveryDateX (1, 2, 3, 4) is depedent on PurchaseDateX (1, 2, 3, 4).

b)
This table is in UNF, or Unnormalized Form, due to the fact that all the data lies within the same `table`. In other words, since the data is not in seperate tables, for exampling having purchases in one table, and customer names in another, it is in UNF.

c)
According to the description, this shop is part of or might be a part of a chain of stores in the future. Therefore, i took some liberties like adding a primary key `storeID` to the table for Store.
`The key, the whole key, and nothing but the key, so help me Codd`. 3NF, or third normal form, says that all non-key attributes must be dependent on the whole key.

This RM is in third normal form because every non-key attribute is functionally dependent on the primary key.



Store: (storeID, name, location)
Pet: (petID, name, desc, price, cost, inStock)
Supplier: (supplierID, name)
Order: (orderID, storeID, supplierID, purchaseDate, deliveryDate)
Customer: (customerID, firstname, lastname, address, city, phone)
Purchases: (purchaseID, customerID, purchaseDate, storeID)
petPurchased: (purchaseID, petID, quantity)

Besvart.

5 ## Construction (40%)

**Problem 4. Read the following Relational Model (RM).**

STUDENT(StudentID, StudentName);
QUALIFIED(FacultyID, CourseID, DateQualified);
FACULTY(FacultyID, FacultyName);
SECTION(SectionNo, Semester, CourseID);
COURSE(CourseID, CourseName);
REGISTRATION(StudentID, SectionNo);

**Exercise:**

a. Write SQL queries to create all the tables listed in the RM.

b. Write SQL queries to insert three values in each table.

c. Create a VIEW to see FacultyName, StudentName, CourseName and DateQualified. Use INNER JOIN to link tables.

d. List all students enrolled in course in 2016 (hint. Use DateQualified) in alphabetical order by StudentName.

e. Delete a student record who's StudentID='X'. (Assume any studentID that have been inserted in step 'b').

**Skriv ditt svar her...**

a)

```
CREATE TABLE IF NOT EXISTS STUDENT (
   StudentID INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
   StudentName VARCHAR(45) NOT NULL,
   PRIMARY KEY (StudentID)
);
```

(I cant get semicolon to work due to my keyboard layout switching to english when using SafeExamBrowser)

```
CREATE TABLE IF NOT EXISTS FACULTY (
   FacultyID INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
   FacultyName VARCHAR(45) NOT NULL,
   PRIMARY KEY (FacultyID)
);

CREATE TABLE IF NOT EXISTS COURSE (
   CourseID INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
   CourseName VARCHAR(45) NOT NULL,
   PRIMARY KEY (CourseID)
);

CREATE TABLE IF NOT EXISTS QUALIFIED (
   FacultyID INT(11) UNSIGNED NOT NULL,
   CourseID INT(11) UNSIGNED NOT NULL,
   DateQualified DATE NOT NULL,
   PRIMARY KEY (FacultyID, CourseID),
   CONSTRAINT fk_qualified_facultyID FOREIGN KEY (FacultyID) REFERENCES FACULTY (FacultyID),
   CONSTRAINT fk_qualified_courseID FOREIGN KEY (CourseID) REFERENCES COURSE (CourseID)
);

CREATE TABLE IF NOT EXISTS SECTION (
   SectionNo INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
   Semester VARCHAR(45) NOT NULL,
   CourseID INT(11) UNSIGNED NOT NULL,
   PRIMARY KEY (SectionNo, CourseID),
   CONSTRAINT fk_section_courseID FOREIGN KEY (CourseID) REFERENCES COURSE (CourseID)
);

CREATE TABLE IF NOT EXISTS REGISTRATION (
   StudentID INT(11) UNSIGNED NOT NULL,
   SectionNo INT(11) UNSIGNED NOT NULL,
   PRIMARY KEY (StudentID, SectionNo),
   CONSTRAINT fk_registration_studentID FOREIGN KEY (StudentID) REFERENCES STUDENT
(StudentID),
   CONSTRAINT fk_registration_sectionNo FOREIGN KEY (SectionNo) REFERENCES SECTION
(SectionNo)
);
```

b)

I am not adding the PRIMARY KEYS for each table in the inserts because of the fact that they are AUTO_INCREMENTED, and will therefore be automatically increased for each input. If necessary, or wanted, one can easily modify the queries to manually enter the Identifier/primary key.

```
INSERT INTO STUDENT (StudentName)
VALUES ('Harald Hund'), ('Bertil Larsen'), ('Frida Kalamander');

INSERT INTO FACULTY (FacultyName)
VALUES ('Fakultet for informasjonssystemer'), ('Fakultet for kunst'), ('Fakultet for Musikk');

INSERT INTO COURSE (CourseName)
VALUES ('IS-200'), ('IS-201'), ('IS-202');

INSERT INTO QUALIFIED (FacultyID, CourseID, DateQualified)
VALUES ('1', '1', 'current_date()'), ('2', '2', '2016-07-12'), ('3', '3', '2016-07-13');

INSERT INTO SECTION (Semester, CourseID)
VALUES ('H16', '1'), ('H16', '2'), ('V17', '3');

INSERT INTO REGISTRATION (StudentID, SectionNo)
VALUES ('1', '1'), ('2', '2'), ('2', '3');
```

c)

```
CREATE VIEW taskC AS
SELECT
   f.FacultyName AS Faculty,
   s.StudentName AS Student,
   c.CourseName AS Course,
   q.DateQualified AS 'Date Qualified'
FROM FACULTY f
INNER JOIN QUALIFIED q ON f.FacultyID = q.FacultyID
INNER JOIN COURSE c ON q.CourseID = c.CourseID
INNER JOIN SECTION se ON c.courseID = se.CourseID
INNER JOIN REGISTRATION r ON se.SectionID = r.SectionID
INNER JOIN STUDENT s ON r.StudentID = s.StudentID;
```

d)

```
SELECT * FROM STUDENT
WHERE
   (SELECT * FROM QUALIFIED
   INNER JOIN REGISTRATION r ON s
   INNER JOIN REGISTRATION r ON r.SectionNo = SECTION.SectionNo
   INNER JOIN SECTION se ON se.CourseID = COURSE.CourseID
   INNER JOIN QUALIFIED q ON COURSE.CourseID = q.CourseID
   WHERE DateQualified LIKE '%2016%' )
GROUP BY StudentID
ORDER BY StudentName DESC;
```

e)

```
DELETE FROM REGISTRATION WHERE StudentID='2';
```

Besvart.