



KANDIDAT

5552

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 10:06
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)

When you use DROP TABLE, you remove the entire table you select and all its content.

```
DROP TABLE STUDENT;
```

When you use ALTER TABLE, you can change the table in different ways, like adding another column or a constraint.

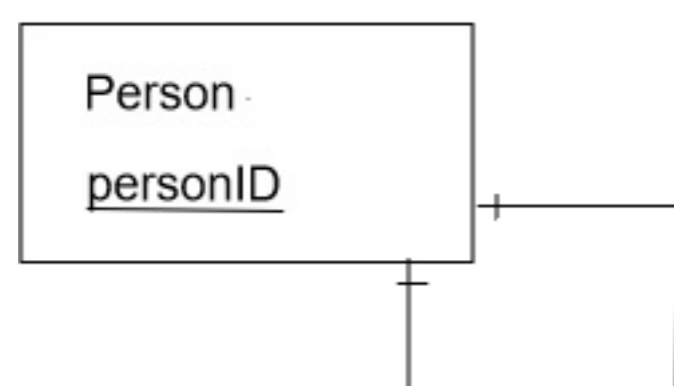
```
ALTER TABLE STUDENT
```

```
ADD COLUMN StudentName varchar(20);
```

When you use DELETE you delete content from tables but not the table itself.

b)

A unary relationship is a relationship with only one type of entity. For example a person being married to another person.



c)

A primary key is a unique identifier. A student can have the same StudentName but not the same StudentID. This helps keeping the integrity of the database.

A foreign key helps maintain relationship integrity. Without foreign keys you risk having the same data stored on multiple locations and you run into problems when the data is altered one place but not the other.

```
Car Owner(PersonID, CarID)
```

```
Car(CarID)
```

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.

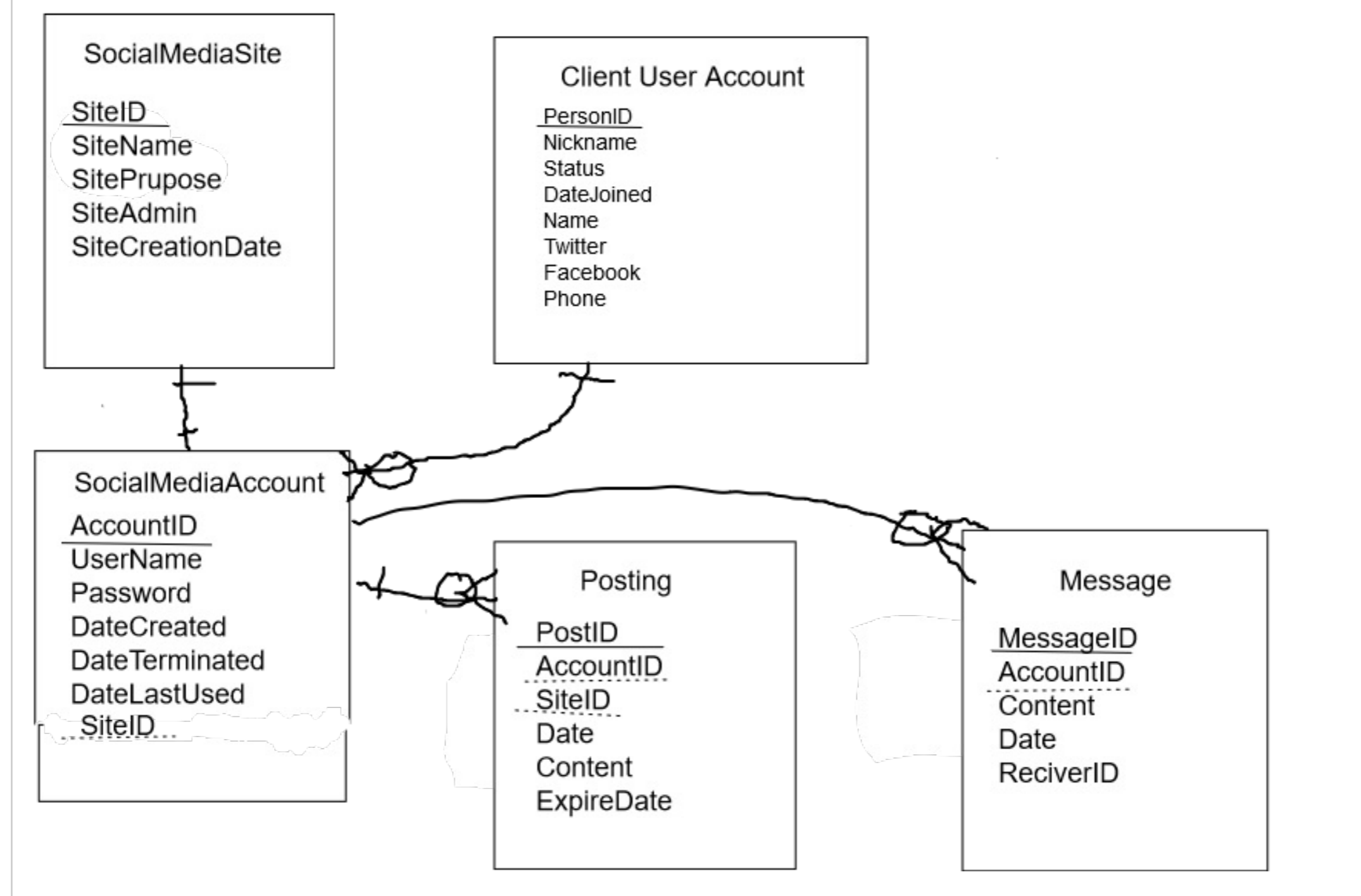
Virtual campus (VC) is a social media firm that specializes in creating virtual meeting places for students, faculty, staff, and others associated with different college campuses. VC was started as a student project in a database class at Cyber University, and online polytechnic college, with headquarters in a research part in Dayton, Ohio. The following parts 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 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.
- 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 Handle, Facebook Page Link, and SMS Contact Number. Anyone may register (no official association with the client is necessary).
- 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.
- 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.
 - A person is permitted to have multiple accounts, each of which is for only one site.
 - A person, over time, may create multiple postings from an account.

Skriv ditt svar her...

A person need to create a user account with the virtual campus and and through the VC creates diffrent accounts for different social media sites.



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
-----

```

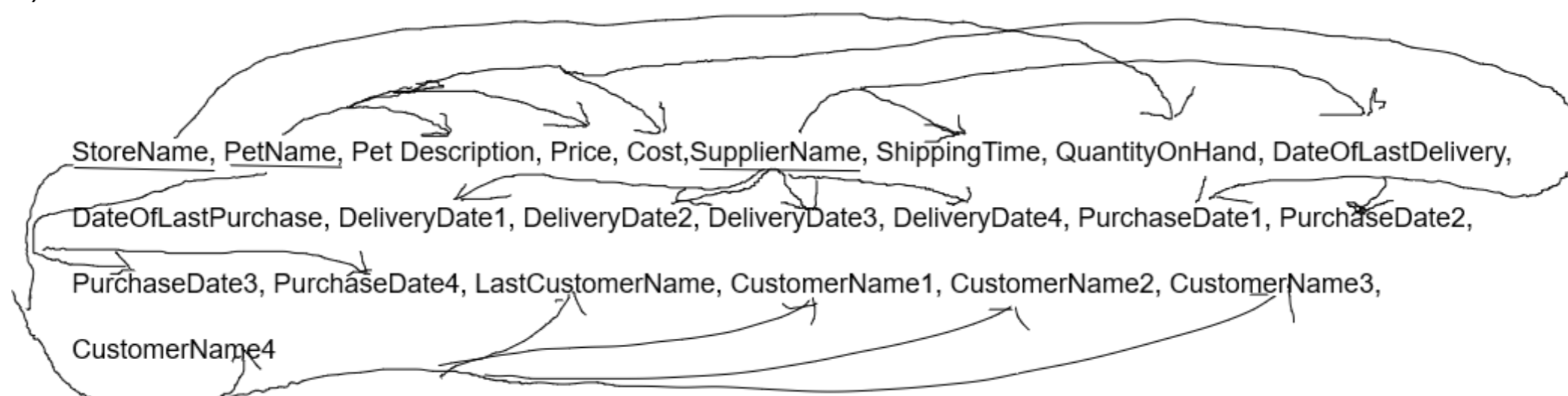
Assume that you want to track all purchase and inventory data, such as who bought the fish, the date that it was

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.

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

Skriv ditt svar her...

a)



b)

This is not in any normal form since we will have multivalued attributes in pet and price etc unless the store only sells one kind of pet.

c)

RM

1NF

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

2NF

PetStore(StoreName, StoreOwner)
 Stock(StoreName, PetName, quantity on hand)
 Pet(PetName, PetDescription, Price)
 Supplier(SupplierName, *PetName*, cost)
 Customer(CustomerName)
 Purchase(CustomerName, PurchaseDate, Price, *PetName*)
 Delivery(SupplierName, DeliveryDate, *Petname*)

3NF

PetStore(StoreName, StoreOwner)
 Stock(StoreName, PetName, quantity on hand)
 Pet(PetName, PetDescription, Price)
 Supplier(SupplierName, *PetName*, cost)
 Customer(CustomerName)
 Purchase(CustomerName, PurchaseDate, *PetName*)
 Delivery(SupplierName, DeliveryDate, *Petname*)

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:

- Write SQL queries to create all the tables listed in the RM.
- Write SQL queries to insert three values in each table.
- Create a VIEW to see FacultyName, StudentName, CourseName and DateQualified. Use INNER JOIN to link tables.
- List all students enrolled in course in 2016 (hint. Use DateQualified) in alphabetical order by StudentName.
- Delete a student record who's StudentID='X'. (Assume any studentID that have been inserted in step 'b').

Skriv ditt svar her...

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

a)

```
create table Student (
  StudentID int(5) not null,
  StudentName varchar(30),
  constraint student_pk primary key (StudentID)
);
```

```
create table Faculty (
  FacultyID int(5) not null,
  FacultyName varchar(30),
  constraint faculty_pk primary key (FacultyID)
);
```

```
create table Course (
  CourseID varchar(8) not null,
  CourseName varchar(15),
  constraint course_pk primary key (CourseID)
);
```

```
create table Qualified (  
FacultyID int(5) not null,  
CourseID varchar(8) not null,  
DateQualified, date,  
constraint qualified_pk primary key (FacultyID, CourseID)  
constraint qualified_faculty_fk foreign key (FacultyID) references Faculty(FacultyID),  
constrant qualified_course_fk foreign key (CourseID) references Course(CourseID)  
);
```

```
create table Section (  
SectionNo int(5) not null,  
Semester varchar(10),  
CourseID varchar(8) not null,  
constraint section_pk primary key (SectionNo, CourseID),  
constraint section_course_fk foreign key CourseID references Course(CourseID)  
);
```

```
create table Registration (  
StudentID int(5) not null,  
SectionNo int(5) not null,  
constraint registration_pk primary key (StudentID, SectionNo),  
constraint registration_student_fk foreign key StudentID references Student(StudentID),  
constraint registration_section_fk foregin key SectionID references Section(SectionID)  
);
```

b)

```
Insert into Student  
Values ('10001', 'Per Hagen')  
      ('10002', Lise Nilsen');
```

```
Insert into Course  
Values ('ORG-100', 'OrgTeori')  
      ('BE-107', 'Økonomi');
```

```
Insert into Faculty  
Values ('00032', 'Samfunn og økonomi')  
      ('00040', 'Kunst');
```

```
Insert into Qualified  
Values ('00032', 'ORG-100', '8.8.2014')  
      ('00032', 'BE-107', '10.8.2016')
```

```
Insert into Section  
Values ('01', 'Første', 'ORG-100')  
      ('02', 'Andre', 'BE-107');
```

```
Insert into Registration  
Values ('10001', '01')  
      ('10002', '02');
```

c)

```
create view Faculty_view as  
select FacultyName, StudentName, CourseName, DateQualified  
from Faculty
```

```
inner join Qualified
on Faculty.FacultyID = Qualified.FacultyID
inner join Course
on Qualified.CourseID = Course.CourseID
inner join Section
on Course.CourseID = Section.CourseID
inner join Registration
on Section.SectionNo = Registration.SectionID
inner join Student
on Registration.StudentID = Student.StudentID;
```

```
d)
Select StudentName
from Student
inner join Registration
on Student.StudentID = Registration.StudentID
inner join Section
on Registration.SectionID = Section.SectionNo
inner join Qualified
on Section.CourseID = Qualified.CourseID
where DateQualified like '%.2016'
order by StudentName asc;
```

```
e)
delete Student
where StudentID = 10001;
```

Besvart.