Databases

TDA357/DIT621/DIT620 (4.5 hec)

Responsible: Ana Bove

Thursday 26th of August 2021, 14:00–18:00

Total: 60 points	
CTH: $\geqslant$ 24: 3, $\geqslant$ 36: 4, $\geqslant$ 48: 5	GU: $\geqslant$ 24: G, $\geqslant$ 42: VG

General notes:

- You can use any help material but you CANNOT communicate with other people other than the teacher of the course in any way.
- There will be one assignment for each of the questions in the exam.
- You need to upload the solution to each question in the corresponding assignment in canvas. You have 30 min (45 if you have extra time) for this so it should be more than enough to get it right!
- We would prefer if you use a computer to type all your answers instead of writing them by hand since handwritten solutions are not always easy to read. In addition, pictures are sometimes not too sharp and clear. Think that what we cannot read we cannot correct!
- If you take pictures of your handwritten solutions please make sure your handwriting is clearly visible and readable in the picture.
- The file extensions allowed are pdf, jpg, jpeg, png, and and txt.
- We advice you to really start working on the uploading of your solution at 18:00 since at 18:30 sharp the system will close for submissions (unless you have extra time) and we will NOT accept submissions by mail.

Good luck!

1 SQL and Constraints (10.5 pts)

Upload your solution to this question into assignment 1 in canvas.
If possible use a computer to type your answer instead of writing it by hand.
File extensions allowed are pdf, jpg, jpeg, png, and and txt.

A veterinary clinic has the following information about their clients, the pets of the clients and the veterinarians that work there. The clinic only takes care of dogs, cats or rabbits.

Clients (name, email, phonenr)
Pets (chipnr, name, owner, type, born)
  owner → Client.name
  type $\in$ {dog, cat, rabbit}
Vets (id, name, phonenr, specialisation)
  specialisation $\in$ {dog, cat, rabbit}

where chipnr and id are numbers that identify the pets and the employees respectively, and born is the year on which the pet was born. Assume name is enough to identify a client.

a) (3.5 pts) Define SQL tables for the relational schema above.

Make sure to add reasonable constraints to your data, for example no pet should have been born in the future! :-)

b) (3.5 pts) Define a table to keep track of the bookings. The table needs to contain information about the veterinarian that is booked, which animal the consultation is about, which day and time the consultation will take place, and the length of the consultation which can be 30 or 60 minutes long, with 30 minutes as default. Consultations can only be booked at 9, 10, 11, 13, 14 or 15 hours.

Observe that it might be possible that the same pet has 2 appointments the same day (on different hours) but of course, neither pets nor veterinarians can have two different appointments at the same day and time.

c) (3.5 pts) Write an SQL query that gives the total amount of money that needs to be billed to each costumer for all the bookings taking place this month. Every 30 min of consultation costs 500 kr. Each customer that needs to be billed should only appear once and those with no bill this month should not be part of the result.

Note: In order to obtain the current year one can use the following PostgreSQL expression: date_part('year', CURRENT_DATE). Similar with month.

Solution:

a)

```
CREATE TABLE Clients (
  name CHAR(20) PRIMARY KEY,
  email TEXT NOT NULL,
  phonenr INT NOT NULL CHECK (phonenr > 0));

CREATE TABLE Pets (
  chipnr INT PRIMARY KEY,
  name TEXT NOT NULL,
  owner CHAR(20) NOT NULL REFERENCES Clients,
  type TEXT CHECK (type IN ('dog', 'cat', 'rabbit')),
  born INT NOT NULL
  CHECK (born >= 2000 AND born <= date_part('year', CURRENT_DATE)));

CREATE TABLE Vets (
  id INT PRIMARY KEY,
  name CHAR(20) NOT NULL,
  phonenr INT NOT NULL CHECK (phonenr > 0),
  specialisation TEXT
  CHECK (specialisation IN ('dog', 'cat', 'rabbit')));
```

b)

```
CREATE TABLE Bookings (
  emp INT REFERENCES Vets,
  chipnr INT REFERENCES Pets,
  bdate DATE,
  time INT NOT NULL CHECK (time in (9,10,11,13,14,15)),
  length INT NOT NULL DEFAULT 30 CHECK (length IN (30, 60)),
  PRIMARY KEY (emp,chipnr,bdate,time),
  UNIQUE (chipnr, bdate, time),
  UNIQUE (emp, bdate, time));
```

c)

```
WITH MonthBookings AS
 (SELECT Clients.name AS Client, 500*length/30 AS BFee
  FROM Bookings, Pets, Clients
  WHERE Bookings.chipnr = Pets.chipnr AND Pets.owner = Clients.name
        AND date_part('month', CURRENT_DATE) = date_part('month', bdate)
        AND date_part('year', CURRENT_DATE) = date_part('year', bdate))
SELECT Client, SUM(BFee) AS Fee
FROM MonthBookings
GROUP BY Client;
```

2 Views and Triggers (11 pts)

Upload your solution to this question into assignment 2 in canvas.
If possible use a computer to type your answer instead of writing it by hand.
File extensions allowed are pdf, jpg, jpeg, png, and and txt.

Recall the relational schema from question 1 on SQL:

Clients (name, email, phonenr)
Pets (chipnr, name, owner, type, born)
owner → Client.name
type ∈ {dog, cat, rabbit}
Vets (id, name, phonenr, specialisation)
specialisation ∈ {dog, cat, rabbit}

extended with the table that was added for the bookings.

After running the clinic for a few years, the owners would like to make some improvements in their database. Propose a solution (meaning, the corresponding full SQL code) to each of the changes the company wants to make.

Recall that as a general rule, contraints, views and triggers should be applied in the appropriate order: if a constraint is enough just add it, and if a constraint or a view can adequately do the job, do not use a trigger!

a) (3 pts) Keep track of the name and phone number of all clients with at least 2 pets in the clinic.

b) (5 pts) If a pet is older than 5 years, one can only book 60 minutes consultation with a veterinarian that is specialised in that kind of animal.

c) (3 pts) Keep track of the name and specialisation of all veterinarians who had more than 1000 hours consultation in the clinic.

Note than only consultations that already took place (meaning those with date today or earlier) should be counted, not the consultations that will take place in the future.

Solution:

a)
```
CREATE OR REPLACE VIEW MultiplePets AS (
  SELECT Clients.name, phonenr
  FROM Clients, Pets
  WHERE Clients.name = owner
  GROUP BY Clients.name
  HAVING COUNT(Clients.name) > 1);
```

b)
```
CREATE OR REPLACE FUNCTION old_pets() RETURNS TRIGGER AS $$
DECLARE
  yborn INT;
  spec TEXT;
  ptype TEXT;
BEGIN
 yborn = (SELECT born FROM Pets WHERE chipnr = NEW.chipnr);
 spec = (SELECT specialisation FROM Vets WHERE id = NEW.emp);
 ptype = (SELECT type FROM Pets WHERE Pets.chipnr = NEW.chipnr);
 IF date_part('year', CURRENT_DATE) - yborn >= 5
 THEN IF spec = ptype
      THEN NEW.length = 60;
      ELSE RAISE EXCEPTION 'wong specialist';
      END IF;
 END IF;
 RETURN NEW;
END;
$$ LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS book_old_pets ON Bookings;

CREATE TRIGGER book_old_pets
  BEFORE INSERT OR UPDATE ON Bookings
  FOR EACH ROW
  EXECUTE FUNCTION old_pets();
```

c)
```
CREATE OR REPLACE VIEW ThousandHours AS (
  SELECT name, specialisation
  FROM Vets, Bookings
  WHERE id = emp AND bdate <= CURRENT_DATE
  GROUP BY emp, name, specialisation
  HAVING SUM(length)/60 > 1000);
```

3 Relational Algebra (8.5 pts)

Upload your solution to this question into assignment 3 in canvas.
If possible use a computer to type your answer instead of writing it by hand.
File extensions allowed are pdf, jpg, jpeg, png, and and txt.

Recall the relation schema from question 1 on SQL:

Clients (name, email, phonenr)
Pets (chipnr, name, owner, type, born)
  owner → Client.name
  type ∈ {dog, cat, rabbit}
Vets (id, name, phonenr, specialisation)
  specialisation ∈ {dog, cat, rabbit}

extended with the table that was added for the bookings.

Note: For this question, it is fine to use the same expression in relational algebra as in SQL to work with dates, so for example use CURRENT_DATE even in relational algebra to get today's date.

a) (2.5 pts) Write a relational algebra expression that gives all the pets' names and their owners that had a booking during the last year (the last 365 days). The result cannot contain repetition.

   Note: the result should only contain the bookings that took place the last year, not future bookings!

   Hint: CURRENT_DATE - 1 gives yesterday's date.

b) (3 pts) Write a relational algebra expression that gives the ids of all cat specialists that have no bookings today at 13.

c) (3 pts) Write a relational algebra expression that gives the name and phone number of all clients with at least 2 pets in the clinic.

Solution:

a) $\delta(\pi_{\texttt{owner},\texttt{Pets.name}}(\sigma_{\texttt{CURRENT\_DATE}\geqslant\texttt{bdate}>\texttt{CURRENT\_DATE}-365}(\texttt{Pets} \bowtie \texttt{Bookings})))$

b) $\pi_{\texttt{id}}(\sigma_{\texttt{specialisation}='\texttt{cat}'}\ \texttt{Vets}) - \pi_{\texttt{emp}\rightarrow\texttt{id}}(\sigma_{\texttt{bdate}=\texttt{CURRENT\_DATE}\wedge\texttt{time}=13}\ \texttt{Bookings})$

c)
$$
\begin{array}{l}
\pi_{\texttt{cname},\texttt{phonenr}}\\
\quad\sigma_{\texttt{nrpets}>1}\\
\qquad\gamma_{\texttt{Clients.name}\rightarrow\texttt{cname},\texttt{phonenr},\texttt{COUNT(Clients.name)}\rightarrow\texttt{nrpets}}\\
\qquad\quad(\texttt{Clients} \bowtie_{\texttt{Clients.name}=\texttt{owner}} \texttt{Pets})
\end{array}
$$

4 ER Modelling (11 pts)

Upload your solution to this question into assignment 4 in canvas.
If you write your diagram by hand, please make sure the picture you upload is legible.
If possible use a computer to type your relational schema instead of writing it by hand.
File extensions allowed are pdf, jpg, jpeg, png, and and txt.

a) (6 pts) Make an ER-diagram for parts of an online book repository, where users can write their own stories and read stories written by others.

The text of each book is divided into chapters, each chapter has its own title in addition to the text. Book titles are unique across the whole repository but chapters in different books may have the same title.

All users have an email and a name. The users adding books to the repository are designated authors. Each author has a short biography text for their presentation page.

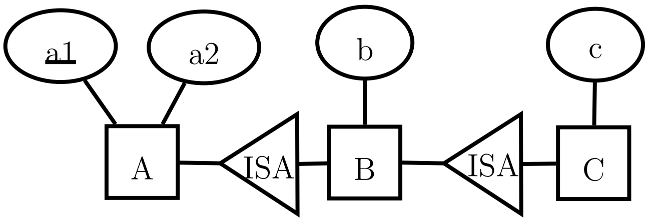
Every book has an owner (must be an author) and may have any number of additional co-authors.

Users (including authors) can rate books; rates consist of 1-5 stars.

b) (2 pts) Translate the part of the diagram related to books and their chapters into a schema.

Note: no other part of the diagram should be translated into a schema! This applies also to any relationship between books and their chapters, and the rest of the domain.

c) (3 pts) Translate the ER-diagram below into a schema. For full points the NULL-approach for ISA-relationships should be used whenever possible.

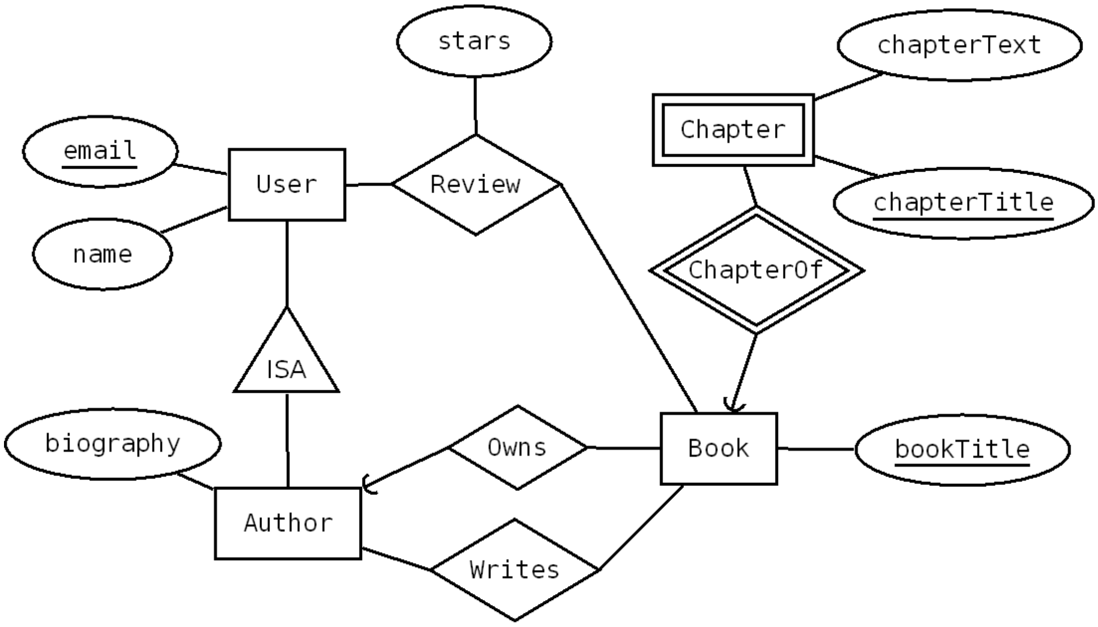


Note: Mark nullable attributes in your schema by writing (or null) after their names.

Hint: "possible" means that the translation preserves the properties we expect when using inheritance, meaning every B is an A and every C is a B.

Solution:

a) Important stuff:
Chapter should be a weak entity supported by Book.
Author ISA User.
There should be two relationships between Author and Book for owner and co-authors, with owner being many-to-one.
Reviews should be a relationship between User and Book, with an attribute for number of stars.



b)
```
Books (_bookTitle_)
Chapters (_bookTitle_, _chapterTitle_, chapterText)
```

c)
```
A(_a1_, a2)
B(_a_, b, c (or null))
  a -> A.a1
```

Note that using null approach (schema below) for A-B is not correct, because it allows being a C without beeing a B.

```
A(_a1_, a2, b (or null), c (or null))
```

Adding a constraint like “c IS NULL OR b IS NOT NULL” would solve this and would also give full points.

5 Functional and Multivalued Dependencies (11 pts)

Upload your solution to this question into assignment 5 in canvas.
If possible use a computer to type your answer instead of writing it by hand.
File extensions allowed are pdf, jpg, jpeg, png, and and txt.

Consider the domain of books with authors and reviews from question 4.

a) (1.5 pts) Write a "domain relation" containing all the attributes (and no duplicate attributes or such). This attribute set will be used in the other parts of this question. The solution should be on the form `D (list, of, attributes, here)`. Name the attributes so their intention is self-explanatory.

b) (3x(1+1) pts) Write three functional dependencies, using the attributes from the domain relation above:

- A sensible FD (based on the domain description) where the right hand side attribute represents author biography text.
- A sensible FD (based on the domain description) where the right hand side attribute represents chapter text.
- A sensible FD (based on the domain description) where the right hand side attribute represents the number of stars given in a book review.

For each funcional dependency, explain why they make sense in the domain. All FDs should be non-trivial and minimal (no unnecessary attributes on the left hand side).

c) (2 pts) A fellow database designer claims that `bookTitle` ↠ `chapterText` is a sensible multivalued dependency to express the fact that a book contains multiple chapter texts. Explain why this is incorrect.

d) (1.5 pts) Give a counterexample of the MVD in the previous question in the form of a small table (a header row with attribute names and at least two rows of contents). The table should contain possible valid data for the database that violates the MVD. There is no need to include all the attributes of the domain, only those that are relevant to the example.

Solution:

a) Something like this:
D(chapterText, chapterTitle, bookTitle, userEmail, userName, presentationText, reviewStars)

b) email → presentationText
The email of an author uniquely identifies it and thus it can have (at most) one presentation.

chapterTitle bookTitle → chapterText
This assumes chapterTitle are unique within books.

email bookTitle → reviewStars
Users can post a single review for each book.

c) That would mean that from the perspective of a book, chapter text is independent from all other attributes - including e.g. chapter titles.

d)

```
bookTitle | chapterTitle | chapterText
---------------------------------------
book1     | chapter1     | text1
book1     | chapter2     | text2
```

6 JSON (8 pts)

Upload your solution to this question into assignment 6 in canvas.
If possible use a computer to type your answer instead of writing it by hand.
File extensions allowed are pdf, jpg, jpeg, png, and and txt.

Consider the domain of books with authors and reviews from question 4.
The JSON Schema below describes a JSON format for parts of this domain.

```
{ "type": "array",
  "description": "Every item is a book",
  "items": {
    "type": "object",
    "description": "A single book",
    "required": ["bookTitle", "authors", "reviews" ],
    "properties": {
      "bookTitle": { "type": "string" },
      "authors": {
        "type": "array",
        "description": "All authors, the first one is the owner.",
        "minItems":1,
        "items": {
          "type": "object",
          "required": [ "email", "name" ],
          "properties": {
            "email": { "type": "string" },
            "name": { "type": "string" }
          } } } },
    "reviews": {
      "type": "array",
      "description": "Every item is a review left by a user",
      "items": {
        "type": "object",
        "required": ["email", "stars" ],
        "properties": {
          "email": { "type": "string"  },
          "stars": { "type": "integer" }
        } } } } }
```

a) (3 pts) Make an example document that validates against the schema (and uses it the way it is intended). The example should contain a single book with two authors and one review.

b) (2.5 pts) Show how the schema can be modified to make sure that there is always between 1 and 5 stars in a review. No need to include the whole modified schema, just enough to make it clear where the addition/modification is placed.

c) (2.5 pts) Write a JSON path that finds the name of the owner of the book "JSON Tutorial".

Solution:

a)
```
[
  {"bookTitle":"JSON Tutorial"
  ,"authors": [{"email":"a1@example.com", "name":"Owner Ownsthings"}
              ,{"email":"a2@example.com", "name":"Coauthor Writesstuff"}]
  ,"reviews": [{"email":"a1@example.com", "stars":5}]
  }
]
```

b) Modifying the stars property of reviews to this should do it:

```
"stars": {
  "type": "integer",
  "minimum":1,
  "maximum":5
}
```

c) $[@(bookTitle="JSON Tutorial")].authors[0].name

Note: variations on how the filter is applied may be possible.