Technische Universiteit Eindhoven University of Technology

Examination cover sheet

Course name: Data Modeling and Databases Course code: 2ID50, exam code 2ID51

Date: January 20, 2015

Start time: 9:00am End time: 12:00pm (except for students with extension)

Number of pages: 16 (only the first 14 pages are to be handed in by the students)

Number of questions: 10

Maximum number of points/distribution of points over questions: Indicated with each question.

Method of determining final grade: Addition of the marks for each question.

Answering style: formulation, order, foundation of arguments, multiple choice: Fill in the blanks on the pages with the exam questions.

NOTE THAT ANYTHING WRITTEN ON OTHER SHEETS OF PAPER WILL NOT BE ACCEPTED AND WILL NOT BE GRADED.

Exam inspection: The exam questions have been inspected by Paul De Bra, Joaquin Vanschoren and Alexander Serebrenik.

Other remarks: Please study the advice given on page 2 carefully before starting to answer questions.

The last page of this exam consists of descriptions you may need for several questions. Please TEAR OFF this last page for easier access. (You need not hand in that last page at the end.)

Instructions for students and invigilators

Permitted examination aids (to be supplied by students):

□ Notebook: NO

□ Calculator: NO

□ Graphic calculator: NO

□ Lecture notes/book: NO

X One A4 sheet of annotations (double sided)

X Dictionar(y)(ies). If yes, please specify: A dictionary from any language to/from English is allowed only after being inspected by the invigilator.

Important:

- examinees are only permitted to visit the toilets under supervision
- it is not permitted to leave the examination room within 15 minutes of the start and within the final 15 minutes of the examination, unless stated otherwise
- examination scripts (fully completed examination paper, stating name, student number, etc.) must always be handed in
- the house rules must be observed during the examination
- the instructions of examiners and invigilators must be followed
- no pencil cases are permitted on desks
- examinees are not permitted to share examination aids or lend them to each other

During written examinations, the following actions will in any case be deemed to constitute fraud or attempted fraud:

- using another person's proof of identity/campus card (student identity card)
- having a mobile telephone or any other type of media-carrying device on your desk or in your clothes
- using, or attempting to use, unauthorized resources and aids, such as the internet, a mobile telephone, etc.
- using a clicker that does not belong to you
- having any paper at hand other than that provided by TU/e, unless stated otherwise
- visiting the toilet (or going outside) without permission or supervision

Name and initials: Student nr:

Use this page for answers to questions you messed up during an earlier attempt.

Name and initials: Student nr:

TECHNISCHE UNIVERSITEIT EINDHOVEN Faculteit Wiskunde en Informatica

Exam, Data Modeling and Databases (2ID51), January 20, 2015

Please write your name and student number on the top of this page. If you remove the staple, please also write your name on the top of each page of this exam.

This three-hour exam is closed book/laptop/phone/ipad/tablet/etc. Other than a single (double-sided) A4 sheet of notes, no reference material may be used.

Tips for passing this exam:

- First look through *all* the questions and decide upon an order in which to try to answer them. In choosing this order of questions you can answer correctly within the given time limit and to optimize having a set of questions answered correctly that add up to enough points to pass the exam. Not all questions are valued equally! Careful planning is important because it is very likely that you will run out of time.
- Read descriptions *very carefully* before answering the question(s). Every detail in the description may count! Pay close attention to the use of words like “not”, “any”, “some”, “all”, “different”, etc. If your answer is based on a misinterpretation of a description, you get no points! *If you think that some question is confusing or has multiple possible interpretations please notify an instructor who will clarify the question and eliminate wrong interpretations.*
- For queries, translate the query back to check whether you have answered the right question. You get no credit for a nice query that answers a different question.
- For queries, make sure that your query satisfies at least some minimum requirements for having a chance at scoring: follow the correct syntax of the (right) query language; make sure you use the relations and attributes that matter in the question; make sure you do not use relations and attributes that are completely irrelevant for the question.
- For theory questions with proofs, make sure that you prove the *right property*. If you give a perfect proof for the wrong property, you will get 0 points.
- If you have only 15 minutes left you can probably improve your score more by reviewing and correcting your answers than by starting on a new question.

You must write your answers in the available space between the questions. **You have one extra page: THE BACK OF THE COVER PAGE, that you can use in case you “mess up” somewhere or simply need more space.**

Grading. This exam results in a grade on a scale of 1 to 10, out of 10. Your grade on this exam constitutes 60% of your final grade in the course. Note that even when you already scored a very high grade on the quizzes and the partial exam you still need to score at least a 5.0 on this final exam in order to pass the course.

Name and initials: Student nr:

Modeling

1. (2 points) Design and give in the space below an ER-diagram that models as closely as possible the description provided on the last page of this exam. Note that your ER model must be designed in such a way that the database only contains complete records, i.e., that no null values are needed for missing information. Don’t forget to indicate primary keys using solid underline and discriminators using dashed underline.

Name and initials: Student nr:

2. (1 point) Translate your ER-diagram to the relational model. Clearly indicate the primary keys in every relation with solid underlining and foreign keys with dashed underlining.
The translation needs to be faithful to your ER-diagram; if your ER-diagram diverges from the description, the relational model needs to adhere to your ER-diagram. If in the previous question only an overly simple (incorrect) diagram was given, you will automatically score 0 on this question.

Name and initials: Student nr:

Querying

The querying part concerns the database schema given on the last page of this exam.

For questions 3 and 4 you must choose which query language to use. You must choose between the relational algebra, tuple calculus and QBE and must use a different language for each question.

3. (0.75 points) Write a query in one of the basic query languages (i.e. no aggregation and no recursion allowed) for the following question:
"List all the customers who have bought (possibly in different purchases) the same name of drink in bottles of different size."
An example would be someone who bought a "Chenet" wine in 25cl and in 75cl bottles.

Name and initials: Student nr:

4. (0.75 points) Write a query in one of the basic query languages (i.e. no aggregation and no recursion allowed) for the following question:
"Give the name(s) of the strongest drink(s)."
(Careful: You may not assume that the strength (alcohol percentage) of a drink is independent of the size of the bottle it comes in! The strength of a drink is the maximum of the strengths of that drink in the different size bottles it comes in.)

Name and initials: Student nr:

5. (0.75 points) Write a query in SQL, using aggregation, to answer the following question: "Give the name(s) of the customer(s) who has bought the largest amount of liquid in a single purchase, together with that amount in centiliter."

Note: You are allowed to write something like sum(quantity*size) although you do not need to.

Name and initials: Student nr:

6. (0.75 points) What is the meaning of the following query:

```
SELECT p1.name
FROM product as p1, product as p2
WHERE p1.name = p2.name AND p1.size <> p2.size
        AND p1.alcohol <> p2.alcohol AND p1.name NOT IN (
                SELECT p3.name
                FROM product as p3, product as p4
                WHERE p3.name = p4.name AND p3.size <> p4.size
                        AND p3.alcohol = p4.alcohol )
```

Name and initials: Student nr:

7. (1 point) Consider the following problem: We wish to define pairs of "close" stores. Two different stores are "close" to each other if there is a customer who bought something from both stores on the same day. Furthermore if store A is close to store B and store B is close to store C and A and C are different then A is also close to C (even though A and C may not share any customer).

a) (0.25 points) Is this question a *Gaifman local problem*?

b) (0.75 points) Write a datalog query for this question. Don't forget the "question mark part" at the end!

Name and initials: Student nr:

8. (1 point) Consider a liquor database with a relation *product* with attributes (LID, name, type, size, price, alcohol). Unlike in the query part of the exam we assume that the following constraints hold: name → type,alcohol and name,size → price. (Question 9 refers to the same relation.)

Is this relation in 3NF? Prove your answer. (Without proof you get no points.)

Name and initials: Student nr:

9. (1 point) Consider a liquor database with a relation *product* with attributes (LID, name, type, size, price, alcohol). Unlike in the query part of the exam we assume that the following constraints hold: name → type,alcohol and name,size → price. (Question 8 refers to the same relation.)

You may assume that this relation is not in BCNF. Perform a BCNF decomposition by using the BCNF algorithm we studied in the course. You may postpone the computation of closures when you can properly motivate this.

Name and initials: Student nr:

10. (1 point) Consider the following new type of constraints which we call the "*contains constraint*". Given a relation instance *r* over schema *R* and let X and Y be sets of attributes from *R*. We say that X contains Y holds in *r*, denoted $X \supseteq Y$ if and only if $\forall t1 \in r$ and $\forall A \in Y$, $\exists\, t2 \in r$ and $\exists B \in X$ such that t1[A]=t2[B].

In plain English: X "contains" Y if all the values that appear in the relation for some attribute of Y must also appear (somewhere) in the relation for some attribute of X.

For "contains constraints" we define the following set of inference rules:

(cc1): if X is a set of attributes of R and $Y \subseteq X$ then $X \supseteq Y$.

(cc2): if $X \supseteq Y$ and Z is a set of attributes of R then $XZ \supseteq YZ$.
Note that XZ is shorthand notation for union between sets of attributes $X \cup Z$.

(cc3): if $X \supseteq Y$ and $Y \supseteq Z$ then $X \supseteq Z$

You may assume/believe that cc1, cc2 and cc3 are correct (sound) inference rules for *contains constraints*.

Are the following two inference rules for *contains constraints* also correct (sound)?
If a rule is sound, prove it using only cc1, cc2 and cc3.
If a rule is not sound, prove it by means of a counterexample.

(cc4): if $X \supseteq Y$ and $X \supseteq Z$ then $X \supseteq YZ$ (0.5 points)

(cc5): if $X \supseteq YZ$ and $Z \supseteq W$ then $XY \supseteq WY$ (0.5 points)

(Use this page and the next.)

Name and initials: Student nr:

Name and initials: Student nr:

Please remove this page without removing the staple from the other pages.
DO NOT HAND IN THIS PAGE.

Modeling

You are asked to create a database for a messaging or notification system for the staff of the Department of Mathematics and Computer Science of the TU/e. (Students are not included to reduce the complexity of this exam question. In a real system students would also be included.)

Staff members are identified by an ID (taken from the human-resources database). They have a name, initials, email, primary phone and secondary phone. Although not the primary key the email address must uniquely identify a staff member.

Every staff member belongs to a (single) *category*: OBP (support staff) or AP (academic staff) and for AP there are (disjoint) subcategories for AIO (PhD students), PD (post-docs), UD (assistant professor), UHD (associate professor), HGL (full professor). Every staff member also belongs to a (single) *subdepartment* of which there are three: Mathematics, Computer-Science and Bureau. (The reality is more complex but ignored for this exam.)

Everyone can send messages to an individual (email address), to a set of individuals and to (all members of) one category (or one subcategory). A message cannot be sent to some individuals and some category(ies) at once, it is either to individuals or to category(ies). Only the members of Bureau can send messages to more than one category at once. For every message we keep track of the *timestamp (date and time)* of the message, the sender, the recipient (persons or categories) and a message ID (which might for instance be a private URL linking to the message content). While it may not be possible to model all restrictions in an ER diagram we do want to model explicitly the difference in permissions between members of the Bureau and of the other subdepartments.

Querying

For the query questions we refer to the following database schema for a chain of liquor stores:

customer(<u>cName</u>, street, city, age) We do realize that *age* is not a very smart choice.
store(<u>sName</u>, street, city)
product(<u>LID</u>, name, type, size, price, alcohol)
Bottle size is in centiliter (cl), alcohol in percentage (%)
purchase(<u>PID</u>, <u>LID</u>, cName, sName, quantity, date, time)

Primary key attributes are underlined. (Foreign keys are not indicated.)
The interpretation of the schema is as follows. Customers are uniquely identified by their names, have a home address (street, city) and an age in years. Stores have a unique name and a location (street, city). Products are uniquely identified through an id called LID (liquor id). Each product has a name, type, size, price and alcohol percentage. The same name may appear several times. Type indicates whether it is beer, port wine, wodka, whiskey, brandy, etc. Typically the same drink (name+type) is available in different size bottles: 25cl, 75cl, ... Customers buy one or more products in a single purchase. The purchase has a purchase-id (PID), and each product (LID) appears only once on a purchase. The quantity identifies the number of bottles of that liquor that are bought (so not the size in cl but in units like bottles). Then there are the store (sName), the purchase date and time. The purchase-ids are unique in the whole database, generated globally and not by an individual store.
We assume every customer (cName) appears in *customer* and *purchase*, every store (sName) in *store* and *purchase* and every product (LID) in *product* and *purchase*.

Name and initials: Student nr:

Do not use this page except for "scratch" because it is not handed in!