# Databases 2017 - Assignment 2

Leiden Institute of Advanced Computer Science

## Introduction

This practical assignment consists of three parts and should be delivered by Monday May 22nd 19:00. You are allowed to work as a pair on these assignments. Deliver your submission as a single .pdf report *written using LaTeX*. Each student should hand in their own submission on Blackboard. If you're working in pairs, your partner must be clearly stated.

## 1 Relational Algebra (40 points)

You are hired by a local grocery store to look at their data and answer a couple of questions to help with their marketing. Their current database schema is described below.

Product(pid:integer, timestamp:integer, name: string, price:real, location:string)
Customer(cid:integer, email: string)
Purchases(pid:integer, cid:integer, orderid:integer, amount:integer )
Totals(orderid:integer, cid:integer, totalprice:real, timestamp:integer )

A product ID can occur multiple times in the schema. Each time the location or price is updated, another line is added to the database with a timestamp that indicates the time of change. The name does not get changed, so the same pid will always imply the same name.
Totals is a summary of the purchases table which shows when the purchases were made, and what the combined price of all products were.
Whenever possible, try to do your projections as early as possible.

Give solutions for the following problems:

1. Find the names of products that at some point in time cost more than €20.00 and the names of products that have at some point cost less than €0.10.
2. Find the email addresses of customers that have spent more than €200 at once.
3. Find the pids of products that have had at least one price change.
4. Find the names of products that have both been displayed at location '5-12' and 'A3'
5. Find the cid of customers that have bought each product that at some point cost less than €1.00
6. Find the cid of customers that are registered with the store but have made no purchases.
7. Find the cid of the customer(s) that have made the largest total purchase.
8. Find the most expensive product that has been purchased at least once by each registered customer.
9. Find the pids of products that have not been sold since timestamp `20150625` but have been sold at least once before that date.
10. The grocery store wants to improve its database. Write a query that returns a table that is basically the Purchases table plus the price of the product at the time of purchase.

## 2 Schema Normalization (40 points)

You have been approached to help with the development of a database for a small blog. They currently store all their data for this website in a single table. The current table comprises of the following attributes:

- $\underline{\text{T}}$itle
- $\underline{\text{A}}$uthor
- $\underline{\text{C}}$ategory
- $\underline{\text{P}}$icture
- Te$\underline{\text{x}}$t

The above underlinings show suggested abbreviations to be used in your derivations.
The following statements have been made:

- A picture is unique per title, and can be used to identify the title.
- Each text has a specific picture linked to it
- Each article with the same title must have the same blogtext.
- Each author only posts in a single category
- The title is enough to determine the author and category of an article.

Currently, the table `R: TACPX` stores the information in the database. For each of the following questions, *show your work* and *explain your answers*.

1. Translate the statements above in functional dependences.
2. Determine the keys in this table
3. Derive a minimal cover for R
4. Derive a lossless join decomposition in BCNF and indicate a primary key for each table in the decomposition.
5. Is your decomposition also dependency preserving? If not, derive a dependency preserving decomposition in 3NF.

## 3 Transaction Management (20 points)

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T1: | R(C) | | W(A) | | | | | W(B) | | R(A) | | Cmt. |
| T2: | | | | | R(B) | | | | W(C) | | | Cmt. |
| T3: | | W(B) | | | | | R(C) | | | | | Cmt. |
| T4: | | | | R(A) | | W(C) | | | | | W(A) | Cmt. |

1. Draw the precedence graph of the above schedule.
2. Is the above conflict serializable? Explain your answer.
3. Apply Strict 2PL to the above schedule and determine the waits-for graph for the schedule right after the first deadlock occurs.