**Principles of Database Management**

**Prof. Dr. B. Baesens**

**Prof. Dr. W. Lemahieu**

**Assignment 2022-2023**

# Guidelines

* This assignment must be handed in electronically as **1** **pdf** before December 9th, 2022, 4PM by mailing it to: Manon Reusens [manon.reusens@kuleuven.be](mailto:manon.reusens@kuleuven.be) The subject of your email should be: **assignment PDBM**.
* Do not send multiple e-mails! If you do send multiple e-mails, we randomly choose one to mark.
* The assignment has to be made individually. Multiple solutions may be possible. If certain assumptions are made: please list them clearly in your solution!
* Do not exceed the maximum number of pages. Marks will be subtracted for exceeding the page limit or including appendices.
* Don’t e-mail questions to us about the assignment but put them on the Toledo forum instead.

# Question 1 (10 marks, 3 pages)

Construct the EER model for the below problem. Document your solution: if there are aspects that can’t be included in the EER model, list them clearly.

One of your (hipster) acquaintances thinks he has the next billion dollar start-up idea for an app: Pizza Delivery with Entertainment. He heard from other people that you are following the course on database management, and asks you to design the EER model. Afterwards, he will use the EER model to ask programmers to implement the app.

He explains the basic functionality of the app as follows: customers can order pizzas from restaurants to get delivered to a specific address, and if they want to, they can choose a special ‘entertainment order’. When an order is an entertainment order, the delivery person stays with the customer after delivering the pizza and entertains the customers (by e.g. singing, making jokes, doing magic tricks, …) for a certain amount of time. Now follows a detailed explanation of the range of capabilities of the app.

When people create an account for the app and become app users, they have to indicate their birthday and fill in their name and address. Every user should also be uniquely identifiable.

Once the account is created, the users should be presented with three options.

The first option in the app is to select ‘business owner’. Of these business owners, we also ask them to provide their LinkedIn account so we can add them to our professional network. Every business owner can own a number of pizza restaurants. Of these pizza restaurants, we want to register the zipcode, address, phone number, website and the opening hours.

Each pizza restaurant can offer a number of pizzas. Of those pizzas, we want to keep the name (e.g.: margharita, quatro stagioni, …), the crust structure (for example, classic Italian crust, deep dish crust, cheese crust), and the price. While two pizzas from different pizza restaurants may have the same name, they will not be exactly the same as the taste will be different, and thus should be considered unique. Moreover, pizzas should be distinguishable even if they have the same price, e.g., a pizza Margharita from Pizza Pronto in Leuven which costs 12 euro must be distinguishable from a pizza Margharita from Pizza Rapido in Brussels, which also costs 12 euro.

The second option in the app is to select ‘hungry customer’. Of those hungry customers, we need a delivery address. Hungry customers can make orders for pizzas. Each order gets assigned an ID, and we want our app to log the date and time when the order was placed. We also allow the hungry customer to indicate the latest time of delivery, and ask for how many people the order is. An order can be for one or more pizzas.

Also a special type of order can be made: the entertainment order. Not every order has to be an entertainment order. But when a hungry customer indicates that he or she wants to be entertained while eating the pizza, we not only want to register all the regular order information, but also the type of entertainment the user requests, and for how long (a duration).

The third option in the app to select is that of ‘entertainer’. When users select entertainer, they need to provide a stage name, write a short bio about themselves, and indicate their price per 30 minutes. Every entertainment order is fulfilled by exactly one entertainer. Every entertainer can choose for which pizza restaurant(s) he or she wants to work. For each pizza restaurant an entertainer wants to work with, he or she should indicate his or her availability by day (Monday, Tuesday, Wednesday, …).

## Question 2 (10 marks, 2 pages)

A famous and very popular theme park has designed the following EER model for their business. The model keeps track of all venues in the park and the assigned locations of employees on a given date. Popular rides have multiple queues: a regular “standby queue”, a queue for guests riding alone, filling the empty seats and a Fast Pass queue for guests that hold special Fast Pass tickets, which allow them to skip the queue during a certain timeslot.

Given the following EER model for a theme park:

Diagram

Description automatically generated

Map the above EER model to a relational model. Also discuss the possible loss in semantics in the relational model. Include the following aspects:

• Indicate the primary keys/foreign keys

• NULL declarations (if there are any)

• On Delete and On Update integrity rules

• Some examples of extra integrity rules that could be enforced by the application code

## Question 3 (10 marks, 2 pages)

Map the EER of question 2 model to a UML data model. Also include some examples of operations that could be added to model extra semantics.

## Question 4 (10 marks, 1 page)

Given the following well-known normalized relational model which was also used in the course (primary keys are underlined, foreign keys are in italics):

**SUPPLIER**(SUPNR, SUPNAME, SUPADDRESS, SUPCITY, SUPSTATUS)

**PRODUCT**(PRODNR, PRODNAME, PRODTYPE, AVAILABLE\_QUANTITY)

**SUPPLIES**(*SUPNR, PRODNR*, PURCHASE\_PRICE, DELIV\_PERIOD)

**PURCHASE\_ORDER**(PONR, PODATE, *SUPNR*)

**PO\_LINE**(*PONR, PRODNR*, QUANTITY)

* Write an SQL join query to retrieve all product numbers together with the number of the supplier supplying it including the ones without a supplier.
* Write an SQL query to get the product numbers on the purchase order lines that have a total order quantity of 16.
* Write an SQL query that returns the numbers of all products that are supplied by supplier number 84 and are in the five highest product numbers.
* Write an SQL query using ALL or ANY to retrieve the names of the products of which the product type has the highest average available quantity.
* Write an SQL query using the (NOT) EXISTS to retrieve all the names and numbers of products that are not supplied by any supplier.