1. **ER and Relational Model (12-04-10)**

Your best friend makes the perfect pizza. Together you come up with the idea to start a little pizza delivery service. You want to offer the service via the internet. You’ll need a database for that.

   i) Develop an entity relationship - model that meets the following requirements. Don’t forget primary keys and cardinality ratios.

   (1) Customers should be stored with their name, address (including zip code), and phone number.

   (2) The pizza service offers different kinds (e.g. Marinara, Romana, Capricciosa, etc.), different sizes (small, medium, large and family-size). The price depends on the kind and size.

   (3) A customer may order many pizzas at the same time. The date and time of the order, as well as the status (e.g. order taken, prepared, delivered, etc.) must be stored.

   (4) Data about the employees (name, contact information, orders taken care of), must be stored.

   ii) Translate your diagram into relations.

   iii) Did you need any additional relationship type table when performing the translation in (ii)? Explain why or why not.
2. **Entity Relationship Model and translation to Relational Model (2012-03-05)**

Karl Andersson has a boat rental. As summer season approaches, he thinks of the mess of paper that he has to deal with every year. It would be so much more convenient to have everything organized digitally. His friend comes up with a first draft of an ER model shown in Figure 2.

Every boat has a license plate that identifies the boat, a name, and a type (sailing-, motor boat etc.). Furthermore, Karl stores the customer’s name and address (pcode is the postal code). Every rental has a number. A boat is rented from a certain date and time to a certain date and time. The status stores the information whether the boat has been returned or not.

a) Mark the primary key in the ER model (adjust the model if necessary) and give a short explanation, why the primary keys are possible to use.

b) Add cardinality ratios and briefly explain your decision.

c) Translate the ER model to the relational model. For the relationship types: Do you need an extra table? Why or why not?

d) Karl employs a few students during the summer, when there are a lot of customers. During that time, the boat rental is organized in early and late shifts every day from 8am to 14:30pm and from 14:30pm to 21pm. Karl often schedules more than one student for one shift. The employees can rent boats to the customers. In that case, Karl wants to know which student has done the rental. Karl can contact the students via mail and phone. Extend the ERM by that information (don’t forget keys and cardinality ratios). (3 pts)

e) A friend suggests a much easier model, as shown in Figure 3 instead of the one in Figure 2. What is your reaction and supporting argument?
3. ER modeling, Relational algebra, Physical DB design (2011-12-13)

Santa Claus keeps a database over children and their wish lists, its ER diagram shown in Figure 1.

a) Add cardinality ratios to the ER diagram using the following information
   i) Each wish list belongs to only one child, but a child may have more than one wish list.
   ii) A wish list contains at least one item. The same item may occur in more than one wish list.

b) Santa wants to keep track of what items have been given to each child. Extend the diagram to accommodate this information. If you introduce any new relations, show their cardinality ratios.

c) Each item on each wish list should have a priority number, so that items on a wish list could be ordered according to priority. Extend the ER diagram to accommodate this information. If you introduce any new relations, show their cardinality ratios.

\[\text{\textbf{Figure 1. Santa’s wish list database.}}\]

\[\text{id} \quad \text{name} \quad \text{id} \quad \text{name} \quad \text{id} \quad \text{name} \quad \text{date} \quad \text{id} \quad \text{name} \quad \text{id} \quad \text{name} \]

\[\text{child} \quad \text{has} \quad \text{wish list} \quad \text{contains} \quad \text{item} \]

\[\text{zip} \quad \text{goodness} \]

d) The goodness attribute on child stores information on whether the child has been good or naughty. Formulate a relational algebra query that retrieves all names and zip codes of children that have been good.

e) Suppose Santa wants to visit all good children who live in zip code area 75000 … 75500. Formulate a relational algebra query that retrieves all names and zip codes of these children.

f) Since there are approx. 2 billion children in the world (according to the UN), the child table will contain a lot of rows. Suggest method that will speed up the execution of the query formulated in 2(b). Explain briefly why the execution is speeded up.
4. Relational algebra, Physical DB design (2011-12-13)

Suppose that the ER diagram in Fig. 1 is implemented in a relational database using the relational model so that each entity is a table.

a) The goodness attribute on child stores information on whether the child has been good or naughty. Formulate a relational algebra query that retrieves all names and zip codes of children that have been good.

b) Suppose Santa wants to visit all good children who live in zip code area 75000 … 75500. Formulate a relational algebra query that retrieves all names and zip codes of these children.

c) Since there are approx. 2 billion children in the world (according to the UN), the child table will contain a lot of rows. Suggest method that will speed up the execution of the query formulated in 2(b). Explain briefly why the execution is speeded up.

5. Relational algebra

Considering the relations in Fig. 4 formulate a relational algebra query that retrieves all the authors from Greece along with their papers presented at the main conference.

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Presented</th>
<th>University</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directed Graphs</td>
<td>Georgis Aronis</td>
<td>Main Conference</td>
<td>University of Ioannina</td>
</tr>
<tr>
<td>Directed Graphs</td>
<td>Dimitros Papasus</td>
<td>Main Conference</td>
<td>University of Ioannina</td>
</tr>
<tr>
<td>Data Integration</td>
<td>Dimitros Papasus</td>
<td>Workshop 1</td>
<td>University of Ioannina</td>
</tr>
<tr>
<td>Graph Patterns in Graph Databases</td>
<td>Jens Carlsson</td>
<td>Workshop 2</td>
<td>Chalmers</td>
</tr>
<tr>
<td>Graph Patterns in Graph Databases</td>
<td>Mats Axelsson</td>
<td>Workshop 2</td>
<td>Chalmers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Country</th>
<th>University</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greece</td>
<td>University of Ioannina</td>
</tr>
<tr>
<td>Sweden</td>
<td>Uppsala University</td>
</tr>
<tr>
<td>Sweden</td>
<td>Chalmers</td>
</tr>
</tbody>
</table>

Fig. 4
6. Physical database design

Formulate a SQL query on a relation (relations) in Figure 4 whose execution would be speeded up by the following SQL:

    create index presentedUniversities on Paper(University);

Explain briefly why your query would be speeded up!
7. Recovery

a) What is the system log used for? What are checkpoints, and why are they important? What are transaction commit points, and why are they important?

b) What are BFIMs and AFIMs?

c) What are the UNDO-type and REDO-type log entries?

d) Discuss the UNDO and REDO operations

8. Recovery (12-08-28)

a) Describe the basic steps in the recovery procedure according to a multi-user version of the deferred update model (sv. uppskjuten uppdatering) with check points and where strict schedules is assumed.

b) Describe the basic steps in the recovery procedure according to a multi-user version of the immediate update model (sv. omedelbar uppdatering) with check points and where strict schedules is assumed.

c) What type of recovery operations have to be applied to the physical database in applying a recovery procedure after a non-catastrophic failure using the deferred and the immediate update models respectively

Proposal about 8: Instead of describing the steps in general in (a) and (b) look at example below (next page) and do the steps for that exact scenario.
$T_1$  
read_item(A)  
read_item(D)  
write_item(D)  

$T_2$  
read_item(B)  
write_item(B)  
read_item(D)  
write_item(D)  

$T_3$  
read_item(A)  
write_item(A)  
read_item(C)  
write_item(C)  

$T_4$  
read_item(B)  
write_item(B)  
read_item(A)  
write_item(A)  

(b)  

<table>
<thead>
<tr>
<th>Event</th>
<th>Log Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_transaction, $T_1$</td>
<td></td>
</tr>
<tr>
<td>write_item, $T_1$, D, 20</td>
<td></td>
</tr>
<tr>
<td>commit, $T_1$</td>
<td></td>
</tr>
<tr>
<td>checkpoint</td>
<td></td>
</tr>
<tr>
<td>start_transaction, $T_4$</td>
<td></td>
</tr>
<tr>
<td>write_item, $T_4$, B, 15</td>
<td></td>
</tr>
<tr>
<td>write_item, $T_4$, A, 20</td>
<td></td>
</tr>
<tr>
<td>commit, $T_4$</td>
<td></td>
</tr>
<tr>
<td>start_transaction, $T_2$</td>
<td></td>
</tr>
<tr>
<td>write_item, $T_2$, B, 12</td>
<td></td>
</tr>
<tr>
<td>start_transaction, $T_3$</td>
<td></td>
</tr>
<tr>
<td>write_item, $T_3$, A, 30</td>
<td></td>
</tr>
<tr>
<td>write_item, $T_3$, D, 25</td>
<td></td>
</tr>
</tbody>
</table>

$T_2$ and $T_3$ are ignored because they did not reach their commit points. 
$T_4$ is redone because its commit point is after the last system checkpoint.

Figure 23.3
An example of recovery using deferred update with concurrent transactions. (a) The READ and WRITE operations of four transactions. (b) System log at the point of crash.