Høgskolen i Østfold

EKSAMEN

Course code: ITF30307	Course: Databaseadministrasjon og -systemer
Date: 04.12.13	Examination: 09.00 - 12.00.
No aids allowed.	Lecturer: Edgar Bostrøm / Per O. Bisseberg

The examination set consists of 3 pages, this page included. The appendix consists of one page. Please check that the examination papers are complete before you start answering.

The examination set consists of three assignments. All assignments are to be answered.

In several of the assignments you should respond point by point. In some cases just a few sentences are sufficient, in other cases there should be a description / comment / discussion on each of these points.

The time allocated for each assignment indicates how detailed you should answer. Each subtask counts equally.

Grades available : 8. January 2014

The examination results will be made available on the Studentweb no later than two workdays after the announcement of the examination results (www.hiof.no/studentweb).

Assignment 1 Time: 60 minutes.

a) Describe the tasks generally associated with the database administrator role. Make a brief comparison between the roles of the database administrator and the data administrator.

b) What is replication? Explain the advantages and disadvantages of replication as opposed to non-replicated systems.

c) Explain database recovery. The course literature (and partially the lecture notes) describe different recovery techniques – Explain these.

Assignment 2 Time: 60 minutes.

For sub-questions a) and b), we begin with a simplified order-order line-item structure.
For task a) and b): See syntax in the appendix.

ORDER
Orderno Orderdate Customerno

PRODUCT
Productno Productname

ORDER_LINE
Orderno Productno Quantity

a) Make relational algebra statements for the following:
   - Customerno for the customer who has bought 100 "green spike mats" in **one** order. Use natural joins so that you "connect" the tables as your first step.
   - As the above task, but instead we want the result set to contain **everything in the Order relation**. We also want the most **effective** statement. Hint: It might be wise to use something else than a natural join to create the statement.

b) Make relational algebra statements for the following:
   - Productno and Productname on products that have not been sold at all (i.e. they don't exist in an order line).
   - Customerno for the customers who have bought every item in the Product relation.
   - Describe briefly: The course literature and other sources define a grouping operator. Give an example of a relational algebra statement that incorporates this operator.

c) Describe how the concepts of relational algebra may be used to explain the transformation of data from an ordinary OLTP system to a data warehouse.

Assignment 3. Time: 60 minutes.

a) Explain and compare the behaviors of triggers, stored procedures and stored functions.

b) Discuss the advantages and disadvantages in regards to placement of the "application" logic at the database level. What options do we have for placement of "application" logic?

c) The lectures presented and contextualized several database models (e.g. hierarchical, network, relational databases and OODB). Explain, preferably using a table, how XML as a database model fit into this pattern?

Relasjonsalgebra - vanlige operasjoner.

Mengdeoperasjoner:	*Notasjon, variant 1*	*Notasjon, variant 2*
Union	$R \cup S$	R union S
Snitt	$R \cap S$	R intersect S
Mengdedifferanse	R - S R \ S	R difference S R minus S
Mengdeprodukt, kartesisk produkt ("alle mot alle")	$R \times S$	R product S R times S
Spesielt for relasjoner:		
Horisontalt utvalg	$\sigma_{<beting.>}(R)$	R where <bet.> R where <bet.>
Vertikalt utvalg	$\pi_{<feltliste>}(R)$	R[<feltliste>]
Mengdedivisjon. (Gitt R[c,d] og S[d]. c er med i mengden R dividert med S hvis c i R forekommer sammen med alle d-er som finnes i S.)	$R \div S$ R / S	R divideby S
Spesialiteter av produkt:		
θ-join (produkt med en eller annen betingelse på kompatible attributter, f.eks. >, <, og kombinasjoner	$R \bowtie_{<bet.>} S$	$R\ join_{<betingelse>}\ S$ (R join S) where <bet.>
Equi-join (θ-operasjonen er =)	som over	som over
Natural join (Equi-join hvor felles attributt kommer bare en gang) ** den mest vanlige jointypen **	som over	som over
Varianter for produkt:		
Outer join, normalt venstre. (alle i R, samt alle fra S som oppfyller koblingsbetingelsen)	$R \supset\!\bowtie S$	$R\ left\ join_{<bet.>}\ S$
Full join (alle i R, alle i S, samt alle som oppfyller koblingsbet.)	$R \supset\!\subset S$	$R\ full\ join_{<bet.>}\ S$
Semijoin (de i R som tilfredsstiller $R\ join_{<betingelse>}\ S$)	$R \triangleright_{<bet.>} S$	$R\ semijoin_{<bet.>}\ S$

Legg merke til at operasjonene her er på mengder, slik at evt. duplikater tas bort – tilsvarende select distinct i SQL.

Dersom betingelsen er på entydige primær/fremmednøkkelkombinasjoner, droppes ofte <bet>.