UNIVERSITY OF CALIFORNIA
EECS, Computer Science Division
CS 186, Fall 2008

Written Homework 2: Crash Recovery and Database Design

The following questions are intended to test your knowledge of the concepts of ARIES recovery and database design, since we will not have a programming project based on these topics. Please provide complete answers to the following questions in a separate document.

Crash Recovery

For questions 1-5, consider the execution of the ARIES recovery algorithm given the following log. Assume a checkpoint is completed before LSN 0 and the Dirty Page Table (DPT) and Transaction Tables for that checkpoint are empty. Assume further that the data updates corresponding to LSNs 20 and 40 have made it to disk before LSN 70, and that no other updates reach disk before the crash. Our database is using Strict 2PL, and a STEAL/NO FORCE buffer policy.

LSN	Log Record	PrevLSN	UndoNextLSN
10	Update: T1 writes P1	Null	
20	Update: T2 writes P2	Null	
30	T2 abort	20	
40	CLR: Undo T2 LSN 20	30	Null
50	Update: T1 writes P2	10	
60	T1 commit	50	
70	BEGIN CHECKPOINT		
80	T2 end	40	
90	Update: T3 writes P1	Null	
100	END CHECKPOINT		
110	Update: T4 writes P4	Null	
120	Update: T3 writes P3	90	
130	T3 commit	120	
140	T3 end	130	
150	Update: T4 writes P1	110	
160	T4 abort	150	
X	**CRASH—RESTART**		

1. What are the contents of the Dirty Page Table (DPT) and the Transaction Table that are stored in the checkpoint that begins at LSN 70?

2. Assuming Strict 2PL at a page-level granularity, a) On what pages must T2 acquire exclusive locks? b) What is the earliest point at which T2 can release these locks?

3. During Analysis: a) Which log records are read (List them in order)? b) What are the contents of the Dirty Page Table (DPT) and the transaction table at the end of the analysis stage?

4. During Redo: a) Which log records are read (List them in order)? b) Explain your choice of the oldest log record (smallest LSN) that is read. c) Which data pages are read (List them in order)? d) What operations are redone?

5. During Undo: a) Which log records are read (List them in the order they are accessed)? b) Explain your choice of the oldest log record (smallest LSN) that is read. c) What operations are undone? d) Show any new log records that are written (for CLRs, be sure to show the undoNextLSN).

Database Design: CatSnap

Congratulations! You've been hired as the lead architect on an exciting new project: **CatSnap.** CatSnap is the trendiest way for cats to update their furry brethren on all the insignificant micro-details of their feline existences. The idea is simple: kitties use a mobile app to snap photos of themselves, and then send these 'snaps' to their friends. And a brilliant twist: snaps disappear after a set time limit, so risqué kitty snaps are totally safe! As a database guru, your first task is to design a database schema that can handle the massive influx of data coming from the vast user base of feline trendsetters. Specifically, let's start with storing kitties and the snaps they take. The requirements are:

i) For every kitty, we want to store a name, the date they joined CatSnap, their age, and the snaps they've taken.
ii) For every snap, we want to store the photo in the snap and the time it expires.
iii) We wish to ensure that no two snaps are the same, so we'll assign each snap an id.
iv) We don't want to get confused about our kittens (they look so similar sometimes!), so we'll require that no two kitties have the same name and age.
v) For every snap a kitty took, we want to know the resolution of their camera phone
vi) Finally, every snap is of EXACTLY ONE kitty, but kitties can take 0, 1 or more snaps.

6. In the ER diagram on the answer sheet underline the primary keys and connect the given entity and relationship sets using the appropriate line and/or arrow. If bolding a line/arrow, be sure to clearly make it bold.

7. Complete the SQL DDL statements in the answer sheet which will create tables for the Kitties entity set and the Snap'd relationship set in our ER diagram. Don't worry about declaring field types or sizes.

8. Let's add Snap Delivery! As a next step, you'd like to model the friends to whom kitties have sent their snaps, and the number of snaps they've sent. As before, connect the relations for the Sent relationship set in the ER diagram on the answer sheet. Don't worry about underlining primary keys.

9. Again, complete the SQL statement in the answer sheet to create a table for the Sent_To relationship set.

10. Let's say we wish to allow kitties to rate to the snaps they receive with either a 'paws up' or a 'paws down', and more than one kitty might receive/rate a given snap. Which of the following approaches would model this situation correctly? (**Note**: more than one answer may be correct!)

(A) Add a new attribute 'rating' to the Snap'd relationship set that stores 'thumbs up' or 'thumbs down'.

(B) Add a new attribute 'rating' to the Snaps entity set that stores 'thumbs up' or 'thumbs down'.

(C) Add a new relationship set 'ratings' between kitties and snaps with an attribute for the rating type.

(D) Add a new entity set 'ratings' to store the rating and connect it to both snaps and kitties with a ternary relation.

(E) None of the above.