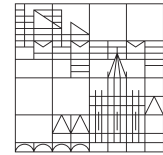


Flashback

Universität
Konstanz



Database System Architecture and Implementation
INF-12950
WS 2013/14

University of Konstanz
Database and Information Systems
Jun.-Prof. Dr. Michael Grossniklaus
Andreas Weiler, Leo Wörteler

i General Notes

In the following, you will find a bunch of questions, which might serve as inspiration for the final test.

Exercise 1: Basics

(0 Points)



- a) What is primary, secondary, tertiary storage?
- b) What are the restrictions of *32-bit* architectures? How can databases surpass these restrictions?
- c) Modern disk devices can transfer 50 or more *MB/second*. Why are they still said to be slow?
- d) How can file access be sped up without using *RAID* architectures?
- e) What's the difference between Intra- and Inter-I/O Parallelism? Which one allows more parallel I/Os?
- f) *RAID 5* is a good general-purpose solution. What's so special about it?

Exercise 2: Disk & Buffer Management

(0 Points)



- a) What are popular block sizes today?
- b) Which techniques exist to organize free blocks on disk?
- c) What are so-called dirty buffer pages?
- d) Which data structure is commonly used for implementing *LRU*?
- e) *LRU* buffers can be sequentially flooded ? what is that?
- f) What are advantages and drawbacks of using the *OS* buffer management?
- g) Which data structures exist to organize heap files?
- h) What can be done if variable-sized records don't fit in a single disk block?

Exercise 3: B+ Tree Index

(0 Points)



- a) What are clustered, sparse, and dense indexes?
- b) When is it appropriate to use multi-dimensional indexes?
- c) What are linked lists needed for in a *B+ Tree*?
- d) Usually, blocks and pointers in *B+ Trees* have fixed size. What can you do to reference variable-size strings?
- e) When building an indexed table, is it faster to... - first create the table? or first create the index structure?
- f) How does bulk-loading in *B+ Trees* work?
- g) What are *Generalized Access Paths*?

Exercise 4: Hash Indexes


(0 Points)



- a) Why is a hash index called a scattered storage structure?
- b) What are collisions?
- c) What might be appropriate hash functions for (1) strings and (2) numbers?


- d) Which approaches exist to handle full buckets in static hashing tables?
- e) What are advantages/drawbacks to dynamic hashing techniques?
- f) What happens after a bucket overflow in extendible hash tables?

Exercise 5: Duplicates & Sorting

(0 Points) 

- a) Which *SQL* operators lead to the elimination of duplicates?
- b) Why is sorting useful if duplicates are to be eliminated?
- c) Which technique might serve as alternative if sorting is too expensive?
- d) How many buffer pages are at least needed to sort large files?
- e) What are advantages and drawbacks of the external merge sort algorithm?

Exercise 6: Operator Evaluation

(0 Points) 

- a) What are logical, what are physical operators?
- b) Name two, three variants for a physical join implementation?
- c) Which data is stored in a system catalog?
- d) What is meant with *low/high selectivity* ?
- e) Which predicates are disjunctive, which are conjunctive?
- f) How do conjunctive normal forms look like?
- g) How does an index nested loop join work?
- h) What are two preconditions for applying merge joins?