Exercise 1 "Implementation of Databases"
Due until Tuesday, 18\textsuperscript{th} of April 2006 (before lecture) SS 06

1.1 Terminology

Define shortly in your own words the following terms:

- Integrity Constraint (give examples of integrity constraints)
- Transaction
- Synchronisation in DBS
- Access paths

1.2 ACID principle, Synchronisation

1. One of the main goals of a DBMS is the protection of databases in multi user mode. Transaction management plays a key role in reaching that goal and is based on the so called ACID principle.

   a) Explain the ACID principle in detail.
   b) Why is the ACID principle necessary? What will happen to the data in a database when the ACID principle is not valid?

2. In the introductory lecture you came across the database transactions synchronisation problems. Describe those problems by giving an example for each of them.

1.3 System architecture

The 5 layer architecture of a DBMS was introduced in the lecture.

   a) Describe the principles and concepts used in the architecture and discuss their advantages.
   b) Describe the six interfaces and the respective level of abstraction in the architecture by explaining what kind of objects and procedures are used in each one of them.
   c) Describe the processing of a simple, fictitious query in the different layers of the DBMS architecture. Which operations are performed on which layer? Suppose you are working with a relational DBMS which is implemented by means of B* trees.
d) The simplified layer model of a DBS is composed of three layers: data system, access system and storage system. Describe the mapping between the simplified layer model and the 5 layer model of a DBS.

1.4 Tree structured access paths

The most frequently used access paths, in addition to hash methods, are search trees (e.g. B- and B* trees).

1) Discuss the differences between B- and B* trees.

2) Let k=1 and k*=2. Given the following B*-Tree:

```
   34
  /   \
26  34---
   |
46 50 61---
```

Perform the listed operations on the B*-Tree and redraw the resulting B*-Tree after each step:

- a) insert 11
- b) insert 52
- c) insert 31
- d) insert 18
- e) insert 25
- f) insert 35
- g) insert 56
- h) insert 29
- i) remove 26
- j) remove 34
- k) remove 29
- l) remove 25
- m) remove 18
- n) remove 56
- o) remove 61