Exercises for week 4
Indexing (lectured early week 3)

(Note that the exercises are from last year´s site & book, but they are equally valid)
(Try them first before you look up the solutions)

Exercise 14.1.1 Suppose blocks hold either five records, or 20 key-pointer pairs. As a function of *n*, the number of records, how many blocks do we need to hold a data file and

a. dense index,
b. sparse index?

Exercise 14.2.1 Suppose that blocks can hold either ten records or 99 keys and 100 pointers. Also assume that the average B-tree node is 70% full; i.e., it will have 69 keys and 70 pointers. We can use B-trees as part of several different structures. For each structure described below, determine

i. the total number of blocks needed for a 100,000-record file, and
ii. the average number of disk I/O's to retrieve a record given its search key.

You may assume nothing is in memory initially, and the search key is the primary key for the records.

a. The data file is a sequential file, sorted on the search key, with 20 records per block. The B-tree is a dense index.
b. The same as (a), but the data file consists of records in no particular order, packed 20 to a block.
c. The same as (a), but the B-tree is a sparse index.
d. The data file is a sequential file, and the B-tree is a sparse index, but each primary block of the data file has one overflow block. On average, the primary block is full, and the overflow block is half full. However, records are in no particular order within a primary block and its overflow block.
e. Instead of the B-tree leaves having pointers to data records, the B-tree leaves hold the records themselves. A block can hold ten records, but on average, a leaf block is 70% full; i.e., there are seven records per leaf block.

Exercise/test Take the 3-minute-test on https://use-the-index-luke.com/3-minute-test/postgresql. It has 5 exercises. The solutions are explained after you try them! And you can find many examples in the book and on the Internet!