Databases II
2019-10-03

1. Write a PL/SQL procedure, which prints out for the parameter user the creation date of his/her oldest table (which was created earliest).

```
CREATE OR REPLACE PROCEDURE oldest_table(p_user VARCHAR2) IS
   v_str VARCHAR2(2000);
   v_date dba_objects.created%TYPE;
BEGIN
   v_str := 'select min(created) from dba_objects '||
          'where object_type= "TABLE" '||
          'and owner = :y';
   EXECUTE IMMEDIATE v_str INTO v_date
   USING p_user;
   dbms_output.put_line(v_date);
END;
/
SET SERVEROUTPUT ON
execute oldest_table('NIKOVITS');
```

2. How many data blocks are allocated in the database for the table NIKOVITS.CIKK?
There can be empty blocks, but we count them too.
The same question: how many data blocks does the segment of the table have?

```
SELECT blocks
FROM dba_segments
WHERE owner='NIKOVITS' AND segment_name='CIKK' AND segment_type='TABLE';
```

3. How many filled data blocks does the previous table have?

```
select count(distinct dbms_rowid.rowid_block_number(rowid))
from nikovits.cikk;
```

4. How many rows are there in each block of the previous table?

```
SELECT dbms_rowid.rowid_relative_fno(ROWID) file_no,
      dbms_rowid.rowid_block_number(ROWID) block_no, count(*)
FROM nikovits.cikk
GROUP BY dbms_rowid.rowid_block_number(ROWID),
dbms_rowid.rowid_relative_fno(ROWID);
```

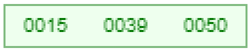
5. There is a table RUDAS.SELLS which has the following row:

szla_szam = 100 (szla_szam is a column name)
In which datafile is the given row stored?
Within the datafile in which block? (block number)
In which data object? (Give the name of the segment.)

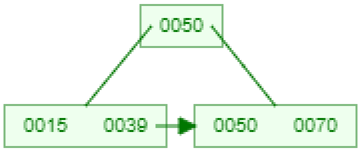
```
select o.object_name, s.relative_fno, dbms_rowid.rowid_object(e.rowid),
dbms_rowid.rowid_block_number(e.rowid)
fromrudas.sells e, dba_objects o, dba_segments s
where szla_szam = 100
and o.object_id=dbms_rowid.rowid_object(e.rowid)
and o.object_name = s.segment_name
and o.owner=s.owner;
```

6. Build a B+ tree from the following keys. Insert the keys into the tree in the given order.
39,15,50,70,79,83,72,43,75,45,60,80
Let's suppose that a node (block) can contain 3 keys and 4 pointers.

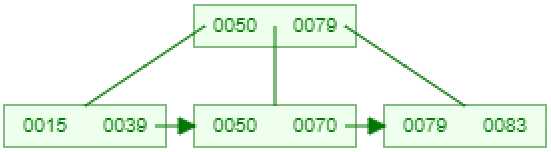
After the first three insertions:



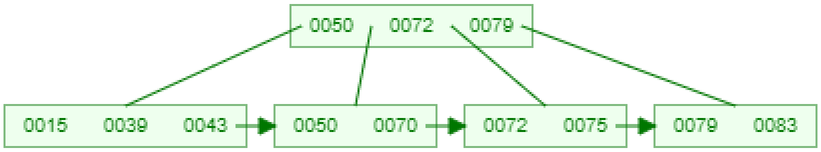
After the first split:



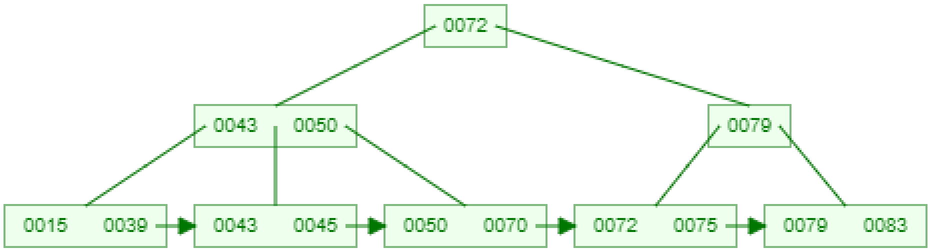
After the second split:



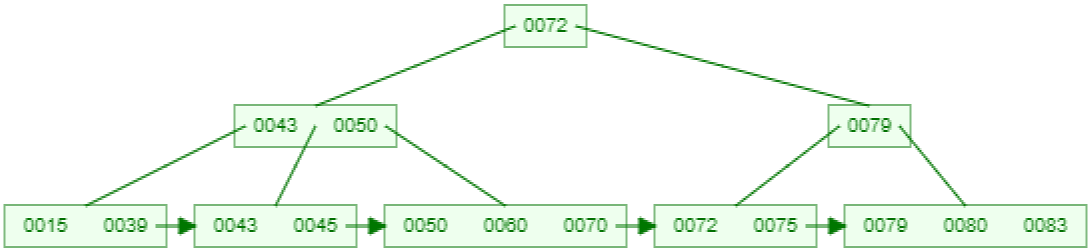
After the third split:



After the fourth split:



The final tree:



Hint: If you would like to practice more, think of random keys, build a B+ tree from them, and then check your results at https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html

7. Encode the following bitvector with run-length encoding:
000000000000100000000010011000000010001

First, we count the length of every 0-sequence. We get: 12, 9, 2, 0, 7, 3.

Note 1: there is usually a 0-sequence after the last 1, but we don't encode it as the size of the table on which the index is created tells us this information.
Note 2: if there are 2 (or more) 1's next to each other, it is important to indicate every 0 long 0-sequence!

Secondly, we convert the lengths to base 2: 12 = 1100, 9 = 1001, 2 = 10, 0 = 0, 7 = 111, 3 = 11

Thirdly, we create a prefix for every base 2 number. The prefixes have *n-1* 1’s and a 0, where n is the number of digits needed for the number. E.g.: 12 = 1100, 4 digits, so the prefix is 1110 (three 1’s and one 0). 9 = 1001, so three 1’s and one 0 again. 2 = 10, two digits, so the prefix is 10 (one 1 and one 0). For 0, we need one digit, so the prefix is one 0 (and zero number of 1’s).

The encoded bitvector is then the prefix for the length of the first 0-sequence and then the actual length. Then the second, the third, and so on. We’ll have:
(12) 1110 1100, (9) 1110 1001, (2) 10 10, (0) 0 0, (7) 110 111, (3) 10 11

The final encoded vector is:

11101100111010011010001101111011