Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

December 15, 2017

Databases IIB: DBMS-Implementation
— Exercise Sheet 10 —

As requested by the students, the repetition questions a) will not be discussed in class unless somebody asks for the solution to a specific question. So please have a look at them before the meeting and decide which questions do you wish to be discussed. Of course, you can also ask any question of your own on the topics of the course.

We will work on Part c) to f) in the lab session, you only have to submit solutions to the homework exercises, i.e. Part g) to i). The official deadline is December 20, 12:00. (before the problem/lab session).

Repetition Questions

a) What would you answer to the following questions in an oral exam?

- Explain a typical block format for a heap file, when rows can have variable length. How can one reuse the space of a deleted row for a slightly larger row if the block has in total enough space (but not in one piece)?
- In view of this block format, explain again the components of a ROWID. How is the row found within the block?
- What happens when a row is updated and no longer fits into its block? How is the row found with its ROWID after the update?
- What happens if the row is updated a second time and must be moved again to a new block? How many block accesses will be needed to find the row with a given ROWID?
- Why does this ROWID concept prescribe a minimum size for the rows (basically, the storage size of a ROWID)?
- Suppose that there are many migrated rows. For which operations does this influence performance? What can you do to improve performance?
- Consider an update done in a full table scan. How do the DBMS handle migrated rows, i.e. when should it update them: When the full table scan reaches the original storage place or when the full table scan reaches the current storage place? What happens if the row must be moved again?
- If you anticipate that there will be updates that increase the storage size of the rows, what should you do in Oracle?

- How can a good value for `PCTFREE` be calculated? What are the assumptions for this calculation?
- What are the consequences of a `PCTFREE` value that is too large or a value that is too small? When does it make sense to set `PCTFREE` to 0 (which is not the default value)?
- If you would develop your own DBMS, can you think of an alternative to `PCTFREE` (that needs less or other assumptions)?
- What are advantages and disadvantages of fixed-length rows compared to rows of variable length? Consider also the case of an `ALTER TABLE` that adds a column. You may also think about data compression techniques.
- Which blocks does Oracle read in a full table scan? Give an example where this looks very inefficient. What are the reasons for this problem? If you would program your own DBMS, would you use the same technique as Oracle?
- Consider a block format for fixed size rows and a linked list of blocks which have free space. How is this linked list managed (i.e. where do you find the start of the linked list and what happens on insertions and deletions of rows)? When a block was full, and a single row was deleted, why might it be good to put the block not immediately again at the start of the linked list (but wait instead until it has space for several rows)? Discuss advantages and disadvantages of this technique.
- Explain the row format of Oracle. How does one find the value of a given column? Discuss alternatives. What are advantages and disadvantages of the Oracle row format? If you develop your own DBMS, which row format would you choose?
- Why do columns at the end of the row that contain a null value need no storage space in Oracle?
- How can one calculate the storage size of a `NUMERIC` column value in Oracle? What are the components of the value in the internal representation? How much space would the value `123` need? And what about the value `-12.34`?
- Explain the `ANALYZE TABLE` command. Why is it no problem if the statistics in the data dictionary are not 100% current? When should the DBA run the `ANALYZE TABLE` command? Why should this not be done during the main office hours? (The `ANALYZE TABLE` command is treated in the next lecture.)

Some Intersting Information

b) Have a look at at least one of the following documents:

- Rachid Belaid: Introduction to PostgreSQL Physical Storage.
[http://rachbelaid.com/introduction-to-postgres-physical-storage/]
- Heikki Linnakangas: New Free Space Map and Visibility Map.
[https://wiki.postgresql.org/images/8/81/FSM_and_Visibility_Map.pdf]

In-Class Exercises

c) Create the relation for which you calculated the storage size in the homework:

`R(A: numeric(5), B: varchar(10), C: varchar(50))`

Make sure that `PCTFREE` is 10. For the homework, we need to insert 10 000 rows of the form:

`(99999, null, 'abcdefghijklmnopqrstuvwxyz')`

It is also possible to use unique numbers instead of `99999`, but please make sure that they need five digits.

- First, insert only a single row.
- Have a look at the storage size of the columns with the function `VSIZE`, e.g. select `VSIZE(A)` (and compare it with the value you calculated in the homework).
- If you like, you can also have a look at the internal representation of a data value, e.g. by selecting `DUMP(A, 16)` (which gives hexadecimal output). If you want to experiment with different data values, the dummy table `DUAL` might be helpful, e.g. the following query shows the internal representation of the number 1:

  `SELECT DUMP(1, 16) FROM DUAL`
- Execute the command

  `ANALYZE TABLE R COMPUTE STATISTICS`
- Then look at the data dicionary table `TABS`, in particular the columns `NUM_ROWS`, `BLOCKS`, `EMPTY_BLOCKS`, `CHAIN_CNT`, `AVG_ROW_LEN`, `AVG_SPACE`. Check whether all values are what you would expect. Of course, `AVG_ROW_LEN` should be the row length that you computed in the homework.

d) An example for a PL/SQL procedure that inserts rows until one block is full was given on the Exercise Sheet 7:

```
CREATE OR REPLACE PROCEDURE FILL_BLOCK AS
    N NUMBER;
BEGIN
    N := 1;
    WHILE N < 2 LOOP
    INSERT INTO R VALUES(34, 'uvwxy');
        SELECT COUNT(DISTINCT DBMS_ROWID.ROWID_BLOCK_NUMBER(ROWID))
        INTO N FROM R;
    END LOOP;
END;
/
```

You can download a file that contains this procedure definition from the following address:

[http://www.informatik.uni-halle.de/~brass/dbi17/fill_block.sql]

Of course, you have to adapt this procedure. As a first step, change the `INSERT` command. Execute the file in SQL*Plus to declare the procedure. If you should get the message "procedure created with errors", enter the command "`SHOW ERRORS`". Then you run the procedure with: `call fill_block();`

- How many rows does the table now contain? The number is one greater than the number of rows that fit into one block.
- You can also have a look at the ROWIDs of the actual rows, and count the rows per block (using `GROUP BY`). As a reminder, you can get the block number in which a row is stored with `DBMS_ROWID.ROWID_BLOCK_NUMBER(ROWID)`. The functions for the other components of a ROWID are: `ROWID_OBJECT` (check whether this is the segment number), `ROWID_RELATIVE_FNO` (find the corresponding data file), `ROWID_ROW_NUMBER`.
- Execute the `ANALYZE TABLE` command again.
- Then look at the data dicionary table `TABS` again, and check whether the values of the columns `NUM_ROWS`, `BLOCKS`, `EMPTY_BLOCKS`, `CHAIN_CNT`, `AVG_ROW_LEN`, `AVG_SPACE` are what you expect. You can also look at the table `COLS`, and check `NUM_DISTINCT` and `NUM_NULLS`. The table `USER_TAB_COL_STATISTICS` might also be interesting.

e) Insert the remaining rows so that the table contains 10 000 rows. (You have to modify the procedure again for that or create a new procedure). Do the `ANALYZE TABLE` again and check the size information. In particular, you need the number of used blocks to compare it with your homework result.

f) Do the update that sets column `B` of all rows to `'ABCDEFGHIJ'`. Do the `ANALYZE TABLE` again and check whether there are migrated rows (i.e. the column `CHAIN_CNT` in `TABS`). If there should be migrated rows, do the table reorganisation. If there are no migrated rows, experiment with longer updates.

Homework Exercises (Homework 10)

g) Consider the following relation:

```
R(A: numeric(5), B: varchar(10), C: varchar(50))
```

Suppose that `PCTFREE` is 10, and one inserts 10 000 tuples of the form

```
(99999, null, 'abcdefghijklmnopqrstuvwxyz')
```

Answer the following questions:

- What is the storage size of the number `99999` in Oracle? (Do not count the column length byte.)
- How long is a row of this example table in Oracle?
- How many bytes are reserved in each block by the `PCTFREE` setting? (Assume a block size of 8 KByte, i.e. 8192 Byte).
- How many tuples fit into one block? (Use 90 Bytes as the overhead for the block header.)
- How many blocks will be used for the table? (Do not forget the segment header block.)

Do all these calculations theoretically. We will do a practical test in the lab session.

h) Suppose an update sets the attribute `B` for all tuples to `'ABCDEFGHIJ'`. Will there be migrated rows? Answer this question theoretically:

- How long will the rows be after the update, i.e. how many additional bytes will be needed?
- Is the `PCTFREE` space reserve sufficient for the growth of all rows stored in one block?
- If not, what would have been a good value for `PCTFREE` to avoid migrated rows for this update?

i) Please download the article "Data Page Layouts for Relational Databases on Deep Memory Hierarchies" von Anastassia Ailamaki, David J. DeWitt und Mark D. Hill published in the VLDB Journal 11:3, Nov. 2002, pages 198-215. You can get the article from the web page of the authors:

[`http://www.cs.wisc.edu/multifacet/papers/vldbj02_pax.pdf`]

Get some overview of the paper. You do not have to read the entire article carefully, it suffices if you understand the basic idea. Then write 1–3 sentences about this idea.