ELE2742 / ENG1467 - PUC-Rio, 2019.1

Lecture 4: Database Storage II

DATE

Lecturer: *Scribe: Kelfin Lin*

1 Disk oriented

In this course, we take it for granted that all of the files are stored in disk. Because we think the memory is too small while the disk is sufficient enough to record all of the data we need.

However, in fact, this may not be the case. There is a higher level course in CMU that focus on the memory part, which assumes all the files are stored in memory.

2 Data representation

Introduction: in the last lecture, we mentioned that the tuple layout is just all about bytes. Therefore, it is the database management system's job to figure out the type of each data. For example, given a record, the software layer will tell the database management system which number of bytes are for that specific type.

For example, the software may tell you number 9 through 11 are the bytes for an integer.

Given these information, the database management system will need to know to how to interpret the integer from bits to human-readable data.

2.1 Types

There are basically four types of data introduced in this course. There could be more, if you want. Just design it yourself.

The four types are : integer, floating number, character, and time.

2.2 Fixed point of float point?

So, now we have a problem, there are two kinds of ways to record data, one is to make the number of bytes floating, like IEEE754, and the other is to make the number of bytes for each type fixed, like what we did in C. Integer always have 4 bytes of data.

Why is this a problem?

Because for the floating point, it is less accurate but faster, while for fixed point, it is slower but more accurate. Therefore, it depends on what you want. For example, in Machine learning, people care about the time more, therefore, floating point is used more often.

3 Large value

Introduction: there might be a case, where the data you want to store occupies a lot of spaces, for example more than a whole page. In this case, you may need more than one pages to store the data. There are actually two ways to do this, but they both use the idea of pointer, that is to use a pointer pointing to another space, which has the accurate data.

3.1 Overflow page

This is just an additional page pointer by the original page. This still belongs to the database management system, therefore, you still have the access to modify values in this page.

3.2 External page

Instead of using the page in database management system, you can also use the space outside of the system. For example, you have a page stored in C drive, then you can have a pointer pointing to this C drive. However, in this situation, this page is not belong to the database management system, which means you don't have the access to modify the values in the page. Only reading is allowed.

4 System catalog

Introduction: as mentioned above, tuple is just bytes. It has no meanings if the database system don't interpret them. So, how exactly does the database system interpret them? This is up to the system catalog. In a specific database management system, we will need to specify what each bytes correspond to what data. Usually, the information_schema will define this for us.

4.1 Accessing table schema

Usually, the table schema is defined or specified using real files. That is to say, if you manually create a file in the system, it will show up in the schema pages. Refer to the lecture video for more information. The professor demonstrates this with a demo.

5 Workload

Introduction: in this section, we focus on how we can "modify" data. We take the number of data into consideration, and take a look at two different models.

5.1 On-line Transaction Processing(OLTP)

In some cases, the amount of records are not light, or the number of records that needed to be modified is not light. In that case, we will consider using OLTP.

In OLTP, the transaction between multiple database could be really fast! However, the disadvantage is that when there are too many data to handle, the process could be slow.

5.2 On-line Analyitical Processing(OLAP)

Contrary to OLTP, OLAP pay more attention to how we can analyze and store tons of data. That is, it can handle many data at the same time in a faster way.

5.3 Comparison

The processor introduced this idea using an example, which I think is very vivid. Suppose we are using the database management system with Amazon. When a user tries to add things to the cart, we will need to "transact" the data instead of analyze the data. Because that single manipulation doesn't worth analyzing at all. However, when the user is doing some searches, there will be many products displayed, which requires a lot of data. In that case, we will need to do some analysis to find out the things that the user want.

6 Storage model

Introduction: this section is highly related to the previous section. Because we know the stratergy we are going to use when encountering different situations, we will need to "implement" this idea. This section will introduce two different storage model, which helps to store data in different situation.

6.1 N-ary Storage Model(NSM)

In this model, we store all the data consecutively. For example, there is a table with 10 columns. Then for a specific record, it has 10 corresponding attributes. We then store these 10 records consecutively.

Advantage Fast when we are trying to modify data, especially inserting data. Also it is friendly to pointers.

Disadvantage Takes the main section as an example. If we have 10 attributes, in some queries, we only care about probably 2 of the attributes. In that case, if we still using this model, we will need to load the entire table in order to find the data we want, which is not friendly to runtime and space.

6.2 Decompose storage model(DSM)

In this model, we no longer store the entire table consecutively. Instead, we store each column or say attribute consecutively. For example, if there is an attribute named id, then we store all the ids in a page or several consecutively pages, so on and so forth.

Advantage this is extremely fast, especially when we are handling where clause, we can filter out a lot of useless data in advance.

Disadvantage This model is not friendly to pointers.