

Mthree Alumni Training



Module 4 – SQL and finance



SQL

Objectives

In this module, we will continue to dive into more functions that will be useful for us.

Trainer Notes

We will review the new methods here then go through demos and exercises to practice

Limits and Offsets

The limit and offset commands will let you return a subset of your results to the user.

```
SELECT * FROM table  
LIMIT number;
```

```
SELECT * FROM table  
LIMIT number  
OFFSET number;
```

- The Limit number provided will limit the number of results returned. If no offset is specified, it will start from the top of the results.
- Offset allows you to specify where you want to start returning results.

Trainer/Trainee Notes

The INSERT() Function

Not to be mistaken with our INSERT INTO basic command, the INSERT() function lets us manipulate a string value.

```
INSERT(string, pos, number, substring);
```

The method allows us to insert the substring from a specific position number for a certain number of characters

Let's look at an example:

```
SELECT INSERT("test string", 5, 4, "Example");
```

Trainer/Trainee Notes

Ask the class what they think the example returns and why:

This will return the result testExampleing

LEFT and RIGHT Functions

The LEFT and RIGHT functions allow you to extract characters from a string – either starting from the left or the right.

```
SELECT LEFT(string,  
numberOfCharacters);
```

```
SELECT RIGHT(string,  
numberOfCharacters);
```

What would be the outputs of the following statements?

```
SELECT LEFT("here is an example string", 5);  
SELECT RIGHT("here is another example", 5);
```

Trainer/Trainee Notes

Answers on the output will be:

For left: "here "

For right: "ample"

The LENGTH() and LOCATE() Functions

While an extremely simple command, LENGTH returns the length of a string.

```
SELECT LENGTH(string);
```

The LOCATE() function locates a substring in a string and returns the position at which the string starts.

```
SELECT LOCATE(substr, fullString)
```

For example, *SELECT LOCATE("tring", "string")* will return 2.

Trainer/Trainee Notes

Both again are handy options when you want to manipulate a string

REPEAT and REVERSE

Other string manipulations that are available to us are the REPEAT and the REVERSE commands

The REPEAT command will simply repeat the string you provide the number of times you ask it to. Note it will return as one value.

```
SELECT REPEAT("string", numberOfTimes);
```

The REVERSE function will reverse the string that you provide it.

```
SELECT REVERSE(string);
```

Trainer/Trainee Notes

Comparing Strings

What if we want to compare two strings to see if they are the same?

SQL provides us with a function to do this:

```
SELECT STRCMP(string1, string2);
```

The results returned on this will be 0 if the strings are the same.

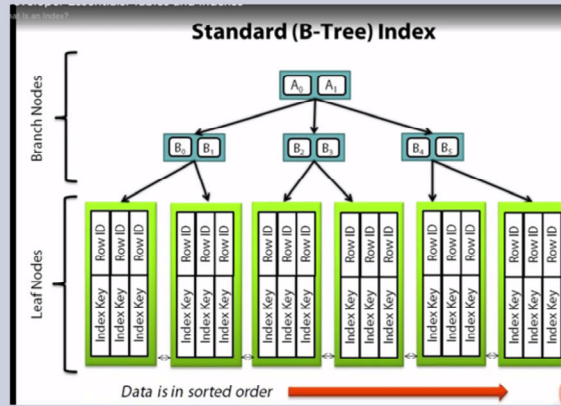
- If string1 is smaller than string2 then -1 will be returned
- If string1 is larger than string2 then 1 will be returned.

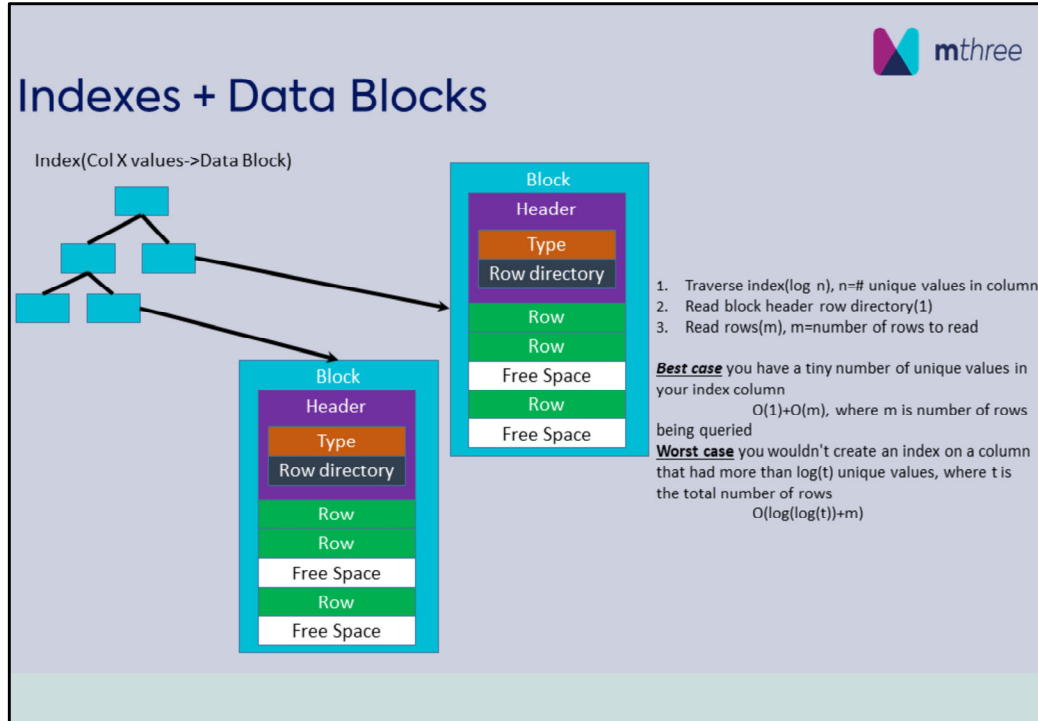
Trainer/Trainee Notes

Indexes

What are A0, A1?

- Values $< A_0$ are in left child, between A_0 and A_1 in middle, and $> A_1$ in right
- Similarly for B, but the diagram only has 2 leaf nodes for some reason





Blocks are similar to disk blocks, but the header stuff for every block seems excessive.