

Besides creating and dropping tables and columns (Data Definition Requests), what other types of requests can you make to a MySQL database? How is data retrieved from a MySQL database? Include the syntax for this type of request.

You can also RENAME TABLE in MySQL database. In order to do this you have to have ALTER and DROP privileges on the original table. You will also need CREATE and INSERT privileges on the new table.

The Syntax for this would be:

```
mysql> RENAME old_table TO new_table;
```

To retrieve the new table the Syntax is:

```
Mysql> SHOW CREATE TABLE new_table\G
```

<https://dev.mysql.com/doc/refman/8.0/en/rename-table.html>

**Research different constructs that are available in MySQL when retrieving data. What types of constructs can be put into the WHERE clause?**

The following examples uses SELECT statements but the same optimization of the WHERE clause can be used and applied for the DELETE and UPDATE statements.

The WHERE clause allows for MySQL to read rows from the Index without even consulting the file as long as the columns are numeric. This allow for joining of tables to be combined by the best possibility. The each table is queried and the best index is used for the more efficient table scan. An example is:

```
SELECT key_part1, key_part2 FROM tbl_name WHERE key_part1=val1;
```

```
SELECT COUNT(*) FROM tbl_name
```

```
WHERE key_part1=val1 AND key_part2=val2;
```

---

This example is only the index tree, assuming the index column is numeric. An example of the construct is:

```
SELECT MAX(key_part2) FROM tbl_name
```

```
WHERE key_part1=constant;
```

.....

These examples show that the table will be read first before any other table in the query. A constant table is usually due to any of the following reasons:

- Empty table or only one row
- WHERE clause is on the primary key or unique key and are not defined as NOT NULL.
- Examples of constructs are:

```
SELECT * FROM t WHERE primary_key=1;
```

```
SELECT * FROM t1, t2
```

```
WHERE t1.primary_key=1 AND t2.primary_key=t1.id;
```

<https://dev.mysql.com/doc/refman/8.0/en/where-optimization.html>

### **What else have you learned about MySQL from your research?**

I learned that computers could pull a list of 100,000's faster than anything. Computers are able to dig through the data within a database to find exactly what you need if you are willing to learn how to code and put it in the time to write it out for a software for everyone to use easily.

More importantly I learned this week we must have a translator to talk from the database to the user. They are not on the same language – this I would think in today's world we would be more advance by now and be able to have them talk to one another but considering I am still confused in class I can see why we need a translator for it to talk to it.

I liked that we can take a table with a primary key and reference it with another table and use foreign keys within a table to keep pulling more cross-referencing data on multiple tables. This is a function I have used numerous times in my past but had no real clue how it was being done in the background.

<https://www.mysqltutorial.org/mysql-foreign-key/>

### **What is your favorite thing you learned this week?**

This week learning that I am not going nuts and the first 6 weeks didn't makes sense with what the teacher was showing was not me going crazy! It was nice to hear from the instructor that we have two different classes being taught at once in this class. I am on the "wrong" class that is being taught so this is why none of the teacher information makes any sense to me!

I now feel better that the teachers really were on a different agenda and I was looking at something different. I don't feel like I am completely out of it. This week it was nice to get my work done and follow along again with Dr. Rob – I love his video's. I did think I was done and couldn't tell but then it all broke when I went to Rachel my mentor. I walked away – uninstalled and well it all worked again! I am happy that I am starting to figure this all out. I still feel like I am missing a big chunk but there are piece of the puzzle I am starting to see. I am not 100% lost anymore!