ORMEDIAN RESEARCH INSTITUTE

TOPIC:

DATABASE

SUBTOPIC: SQL FUNDAMENTALS (PART 8)

By:

ADEKOLA OLUWASEUN O.

JANUARY 6, 2023

CONTENT COVERED

❖SQL VIEWS

- ✓ CREATING VIEWS
- ✓ UPDATING VIEWS
- ✓ DROPPING VIEWS

❖SQL INJECTION

- ✓ WHAT SQL INJECTION IS
- ✓ HOW SQL INJECTION COMES UP
- ✓ IMPLICATION OF SQL INJECTION
- ✓ PROTECTION AGAINST SQL INJECTION

❖SQL HOSTING

- ✓ MEANING OF SQL HOSTING
- ✓ COMMON SQL DATABASE GOSTING

❖SQL DATA TYPES

- ✓ WHAT DATA TYPES ARE
- ✓ COMMON DATATYPES

SQL VIEW

SQL VIEW: *This is nothing but a virtual table that is returned based on SQL predefined queries. Just like any database table, View also is a composition of rows and columns generated from one or many tables or another view entirely . It is simply a result-set returned based on written SQL query.*

CREATE VIEW *statement is used in SQL to create database views.*

View in SQL as a virtual table offers different advantages in the sense that it gives users the ability to do the following:

- *With the use of views, users can structure data in a more intuitive way*
- *Enables users to use summarized results from existing tables to generate reports*

SYNTAX for creation of VIEWS

CREATE VIEW view_name AS SELECT column1, column2 ……. FROM table_name WHERE [condition];

Implementation of SQL View

Let us take a table 1 called sales as an example, we will use this table to create views in this presentation.

CustomerID	ProductName	ProductPrice	Cust_Address	CustName
1	Vanilla	900	23 Belfast Avenue	Sammy Adebayo
2	Cocoa	500	23 Belfast Avenue	Dammy Osekita
3	Butter	1500	Tanke Ilorin	Seun Adekola
4	Bread	1200	5 Magodo Shangisha Lagos	Femi Adekola
5	Palm oil	1700	5 Ketu Lagos	Rasak Mubarak
6	Flour	1000	5 Ketu Lagos	Tunde Bakare

Copy Delete Export

Table link available @ https://github.com/SafersTechnologies/SQL-Tutorials/blob/master/sales.sql

SCENARIO 1: *Our idea in this example is that we want to query the database to return SQL view which will return a result-set as duplicate of the sales table.*

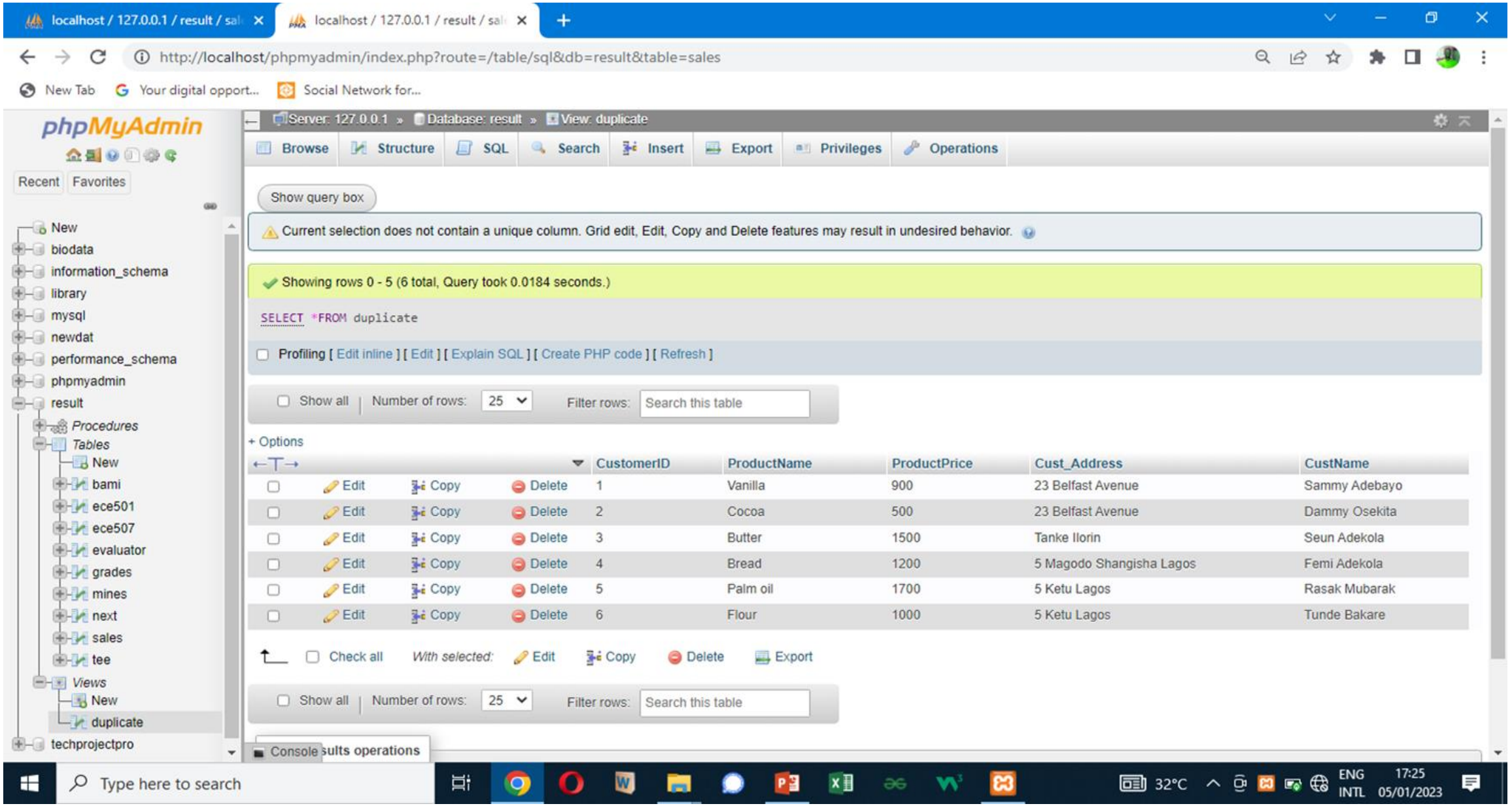
Syntax to achieve this is given as:

CREATE VIEW duplicate AS SELECT CustomerID, ProductName, ProductPrice, Cust_Address, CustName FROM Sales;

Followed by: ***SELECT *FROM duplicate;***

Pictorial result of this query is displayed in the next slide

Link to this implementation is available @ https://github.com/SafersTechnologies/SQL-Tutorials/blob/master/DUPLICATE.sql


localhost / 127.0.0.1 / result / sales
http://localhost/phpmyadmin/index.php?route=/table/sql&db=result&table=sales
phpMyAdmin
Server: 127.0.0.1 » Database: result » View: duplicate
Browse
Structure
SQL
Search
Insert
Export
Privileges
Operations
Show query box
Current selection does not contain a unique column. Grid edit, Edit, Copy and Delete features may result in undesired behavior.
Showing rows 0 - 5 (6 total, Query took 0.0184 seconds.)
SELECT *FROM duplicate
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
+ Options
CustomerID ProductName ProductPrice Cust_Address CustName
1 Vanilla 900 23 Belfast Avenue Sammy Adebayo
2 Cocoa 500 23 Belfast Avenue Dammy Osekita
3 Butter 1500 Tanke Ilorin Seun Adekola
4 Bread 1200 5 Magodo Shangisha Lagos Femi Adekola
5 Palm oil 1700 5 Ketu Lagos Rasak Mubarak
6 Flour 1000 5 Ketu Lagos Tunde Bakare
Edit Copy Delete
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table
Console
New
biodata
information_schema
library
mysql
newdat
performance_schema
phpmyadmin
result
Procedures
Tables
bami
ece501
ece507
evaluator
grades
mines
next
sales
tee
Views
duplicate
techprojectpro
Type here to search
32°C
ENG INTL
17:25 05/01/2023

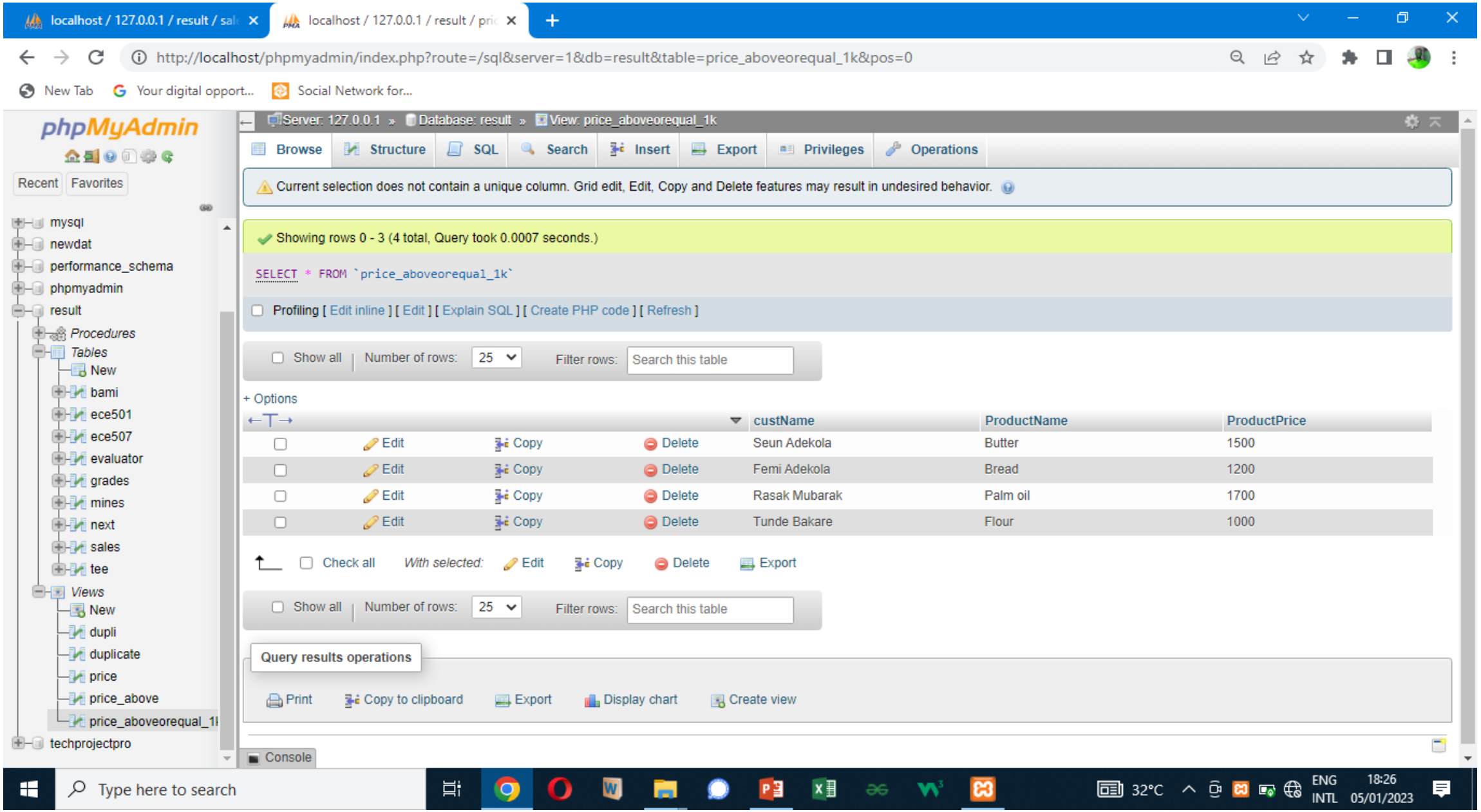
SCENARIO 2: *in this example our aim is to generate a view which will be a representation of only customer's name, product purchased, Product Price where product price is greater than or equal to 1000.*

CREATE VIEW Price_aboveorequal_1K AS SELECT custName, ProductName, ProductPrice FROM Sales WHERE ProductPrice>=1000;

Followed by: **SELECT *FROM Price_aboveorequal_1K;**

Pictorial result of this query is displayed in the next slide

Link to this implementation is available @ https://github.com/SafersTechnologies/SQL-Tutorials/blob/master/price_aboveorequal_1k%20(1).sql


localhost / 127.0.0.1 / result / sal
localhost / 127.0.0.1 / result / pric
http://localhost/phpmyadmin/index.php?route=/sql&server=1&db=result&table=price_aboveorequal_1k&pos=0
New Tab
Your digital opport...
Social Network for...
phpMyAdmin
Recent
Favorites
mysql
newdat
performance_schema
phpmyadmin
result
Procedures
Tables
New
bami
ece501
ece507
evaluator
grades
mines
next
sales
tee
Views
New
dupli
duplicate
price
price_above
price_aboveorequal_1k
techprojectpro
Server: 127.0.0.1 » Database: result » View: price_aboveorequal_1k
Browse
Structure
SQL
Search
Insert
Export
Privileges
Operations
Current selection does not contain a unique column. Grid edit, Edit, Copy and Delete features may result in undesired behavior.
Showing rows 0 - 3 (4 total, Query took 0.0007 seconds.)
SELECT * FROM `price_aboveorequal_1k`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
+ Options
custName
ProductName
ProductPrice
Edit Copy Delete Seun Adekola Butter 1500
Edit Copy Delete Femi Adekola Bread 1200
Edit Copy Delete Rasak Mubarak Palm oil 1700
Edit Copy Delete Tunde Bakare Flour 1000
Check all
With selected:
Edit
Copy
Delete
Export
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Console
Type here to search
32°C
ENG
INTL
18:26
05/01/2023

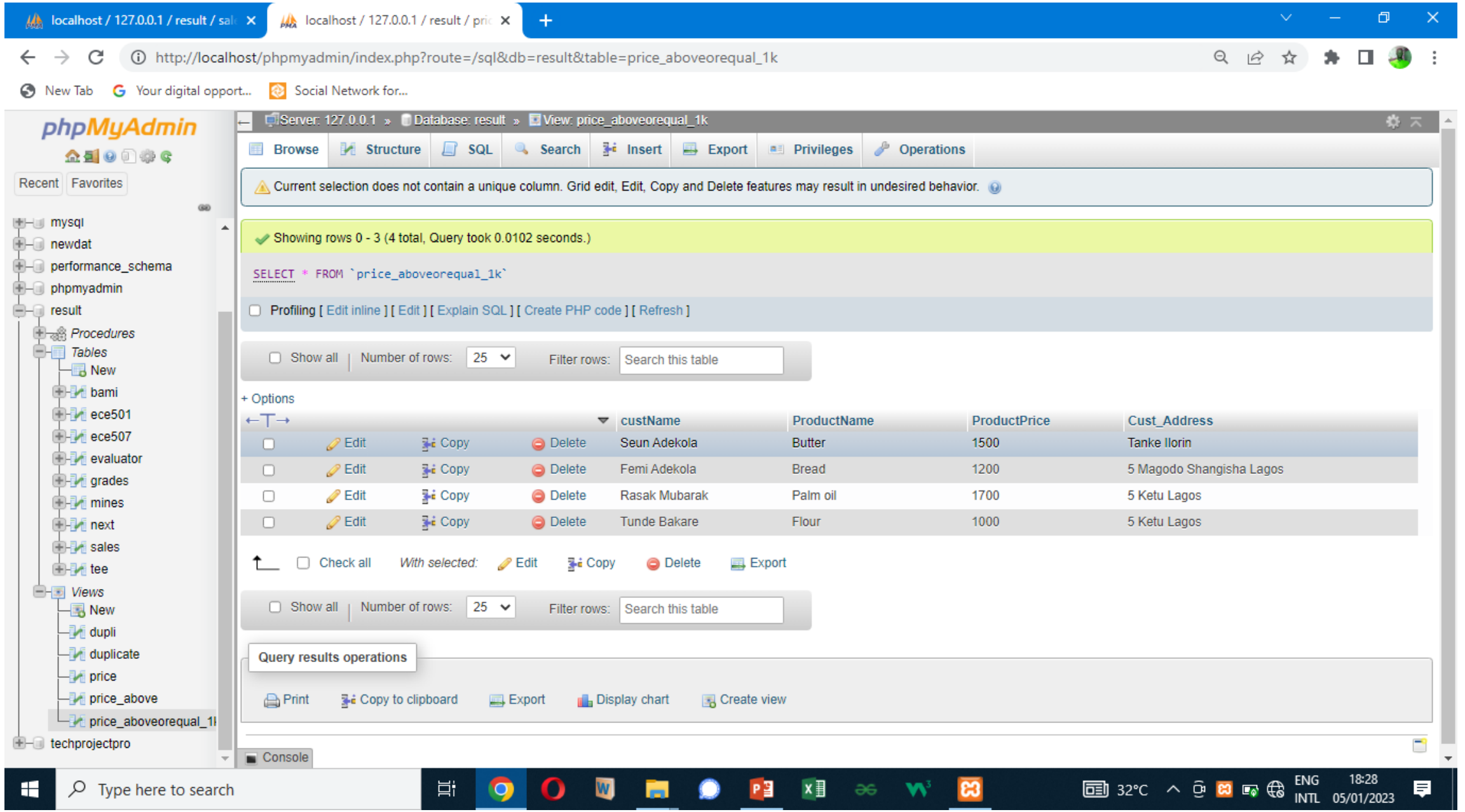
SCENARIO 3: *In this example, our aim is to include the Customer's address field into the result-set of the view generated in SCENARIO 2 i.e. we what to include the address of each customer whose product price is equal or greater than 1000.*

To add a column to SQL view, there's need for us to introduce a new statement called ***CREATE OR REPLACE VIEW****. The functionality of this statement revolves around getting already generated view(s) updated.*

SYNTAX FOR THIS:

```
CREATE OR REPLACE VIEW Price_aboveorequal_1K AS SELECT custName, ProductName, ProductPrice, Cust_Address FROM Sales WHERE ProductPrice >=1000;
```

Link to this implementation is available @ https://github.com/SafersTechnologies/SQL-Tutorials/blob/master/price_aboveorequal_1k_add.sql


localhost / 127.0.0.1 / result / sal
localhost / 127.0.0.1 / result / pric
http://localhost/phpmyadmin/index.php?route=/sql&db=result&table=price_aboveorequal_1k
New Tab
Your digital opport...
Social Network for...
phpMyAdmin
Recent
Favorites
mysql
newdat
performance_schema
phpmyadmin
result
Procedures
Tables
New
bami
ece501
ece507
evaluator
grades
mines
next
sales
tee
Views
New
dupli
duplicate
price
price_above
price_aboveorequal_1l
techprojectpro
Server: 127.0.0.1 » Database: result » View: price_aboveorequal_1k
Browse
Structure
SQL
Search
Insert
Export
Privileges
Operations
Current selection does not contain a unique column. Grid edit, Edit, Copy and Delete features may result in undesired behavior.
Showing rows 0 - 3 (4 total, Query took 0.0102 seconds.)
SELECT * FROM `price_aboveorequal_1k`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
+ Options
custName
ProductName
ProductPrice
Cust_Address
Edit Copy Delete Seun Adekola Butter 1500 Tanke Ilorin
Edit Copy Delete Femi Adekola Bread 1200 5 Magodo Shangisha Lagos
Edit Copy Delete Rasak Mubarak Palm oil 1700 5 Ketu Lagos
Edit Copy Delete Tunde Bakare Flour 1000 5 Ketu Lagos
Check all
With selected:
Edit
Copy
Delete
Export
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Console
Type here to search
32°C
ENG
INTL
18:28
05/01/2023

DELETING AN EXISTING VIEW IN SQL

We can always erase the existence of any view in SQL by simply using the DROP statement. This is just like saying we want to delete a table.

SYNTAX

DROP VIEW view_name;

For example, let us say our focus is on getting the duplicate view we created in the previous sections of this presentation deleted.

DROP VIEW duplicate;

SQL INJECTIONS

SQL Injection : *is a very popular term in the world of computing, it refers to code injection technique whereby malicious SQL statements are entered into an entry field for execution to destroy data-driven applications. Although SQL injection is commonly known as attack vector for websites. It affects web application that uses SQL database such as Oracle, MySQL, SQL Server etc.*

SQL injection could be very dangerous because it allows for complete disclosure of all data available in the database system. With SQL injection attack, it gives attackers the privilege to spoof identity, tamper with existing data etc.

In a nutshell, it is very important to ensure that your database is protected against SQL injection attack.

To summarize this, SQL Injection alternatively known as SQLi is simply a web security vulnerability.

HOW SQL INJECTION OCCURS

It is when you as a database administrator unknowingly runs a malicious query on your database after the user had submitted such malicious SQL statement as entry in one of the input requested you created to get information from users.

If a database is not well protected, attackers might get access to such database by the use of some SQL statements or batched SQL statements.

SQL INJECTIONS

Continuation……..

Implication of SQL Injection based on Batched SQL Statement.

For the fact that most databases support batched SQL statement. Attackers take advantage of such opportunity and apply group of SQL statements separated by semicolon to gain unauthorized access to the database.

This happens because these semicolon separated SQL statements are valid and have similar interpretation as SQL query.

For example:

*If in an input field, an attacker inputs **2; DROP TABLE ECE_records** this could result into deletion of the table called ECE_records if the database security is not strong enough because the direct interpretation of this in the database is:*

SELECT *FROM ECE_records WHERE StudentID = 2; DROP TABLE ECE_records;

PROTECTING DATABASES AGAINST SQL INJECTION

Use of SQL Parameters: *This is one of the ways to protect database from being vulnerable to SQL injection. This technique involves the inclusion of values to SQL query in a more controlled manner at the time of execution.* ***@*** *marker is usually the notation used to represent parameters so that the SQL engine can check whether an input to a field should be accepted or rejected.*

In summary, it is generally believed that the best way to prevent SQL injection is still through the technique of input validation and parametrized queries including prepared statements.

It is imperative that all input fields must be well sanitized by manager or developer of such database to avoid SQL injection.

SQL HOSTING

SQL Hosting *refers to the process of managing a database through the use of SQL.*

It is always good to consider factors such as hosting plans, security and threats associated to any hosting services when choosing any SQL hosting for our database. Typical SQL hosting databases are MySQL, MS

Access, Oracle etc.

Each of these SQL hosting databases offers different capabilities. It is left to us to carefully select which one will be suitable for our intended project.

DATA TYPES IN SQL

Data Types in SQL: *is a very important concept in database development. It is one of the ways to ensure that the wrong value is not passed into the input field by users. Data type is specified in database table columns to define what type of data such column should hold. Whenever we give names to fields in the database table, it is important to include the datatype too at the same time. This is a so important because it identifies how SQL interacts with the stored data.*

There are several types of Datatype but it varies as we interact with different database.

It is important to note that there are 3 data types common to almost all databases. These are:

- *String Data Type*
- *Numeric Data Type*
- *Date Time Data Types*

Each of these data types is further broken down into individual modules based on what actions needed to be performed.

In this presentation, I will

STRING DATA TYPES

- CHAR(size)
- VARCHAR(size)
- BINARY(size)
- VARBINARY(size)
- TINYBLOB
- TINYTEXT
- TEXT(size)
- BLOB(size)
- MEDIUMTEXT
- MEDUMBLOB
- LONGTEXT
- LONGBLOB
- ENUM(val1, ...)
- SET(val1,…)

NUMERICAL DATA TYPES

- BIT (size)
- TINYINT(SIZE)
- BOOL
- BOOLEAN
- SMALLINT(size)
- MEDIUMINT(size)
- INT(size)
- INTEGER(size)
- BIGINT(size)
- FLOAT (size, d) where size is total number of digits while d implies number of digit after the decimal point
- FLOAT(p)
- DOUBLE(size, d)
- DOUBLE PRECISION(size, d)
- DEC (size, d)

DATE AND TIME DATA TYPES

- DATE
- DATETIME(fsp)
- TIMESTAMP(fsp)
- TIME(fsp)
- YEAR

Generally speaking, there are many other string, numeric and date data types. The most important thing is that we should look up documentation for what we want to use a particular data type for and on which platform do we want to use it because keywords used to get some data types work differ as we move from one database platform to another.

THANKS FOR VIEWING

In the next lecture, SQL REFERENCES WILL BE DISCUSSED.

You can always visit my GitHub Profile for SQL tutorials covered so far.

https://github.com/SafersTechnologies/SQL-Tutorials