Acknowledgement

The internship opportunity I had with BCAS campus was a great chance for learning and professional development. Therefore I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having chance to meet so many wonderful people and professionals who led me through this internship period.

I express my deepest thanks to the assessor Mr. Mohamed Nizzad, who works as an IT professional and take lectures on Database Design & Development subjects and others supportive subjects. He helped me taking part in useful decision & giving necessary advices and guidance and arrange all facilities for this assignment. I choose this moment to acknowledge his contribution gratefully. I acknowledge that this assignment was done with help of internet resources.

Sincerely,
S.SHALOMSHAN
Date:29.04.2021

Introduction

The primary purpose of this assignment is to give deep brief explanation about the database concept. This assignment points out some most important essential fundamentals about database design & development. Its also focuses on computer programming paradigms and the other required tools for database such as XAMPP and more.

The task one is mainly focused on the basic idea of database ER diagram.

The second task is I build the physical relational database system.

Next task is I have write the suitable query produce the output result.

The next one is I have explained the data validation and verification process and techniques. Review and test the develop database systems against the user system requirements.

Next I have created the user and technical manual and evaluate the designed database system.

Contents

PART :- 1

Design a relational database system using appropriate design tools and techniques for the above scenario with clear statement of user and system requirements

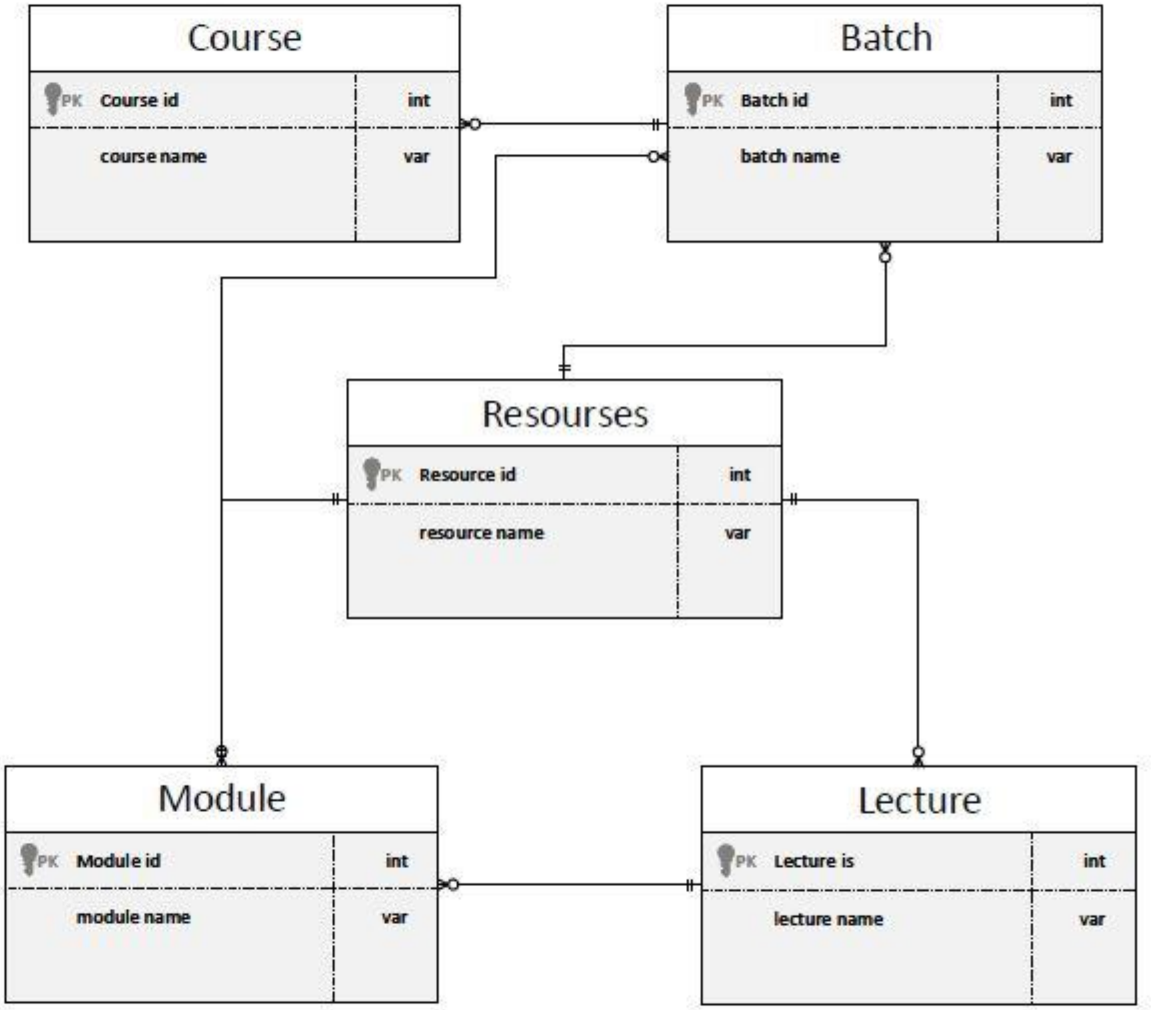


Figure 1 ER Diagram

PART:- 2

TASK:- 2.1

Introduction to Database

Relational database was proposed by Edgar Codd (of IBM Research) around 1969. It has since become the dominant database model for commercial applications (in comparison with other database models such as hierarchical, network and object models). Today, there are many commercial Relational Database Management System (RDBMS), such as Oracle, IBM DB2 and Microsoft SQL Server. There are also many free and open-source RDBMS, such as MySQL, mSQL (mini-SQL) and the embedded JavaDB (Apache Derby).

A relational database organizes data in tables (or relations). A table is made up of rows and columns. A row is also called a record (or tuple). A column is also called a field (or attribute). A database table is similar to a spreadsheet. However, the relationships that can be created among the tables enable a relational database to efficiently store huge amount of data, and effectively retrieve selected data.

A language called SQL (Structured Query Language) was developed to work with relational databases.

Primary Key

In the relational model, a table cannot contain duplicate rows, because that would create ambiguities in retrieval. To ensure uniqueness, each table should have a column (or a set of columns), called primary key, that uniquely identifies every records of the table. For example, an unique number customer ID can be used as the primary key for the Customers table; product Code for Products table; isbn for Books table. A primary key is called a simple key if it is a single column; it is called a composite key if it is made up of several columns.

One-to-Many

In a "class roster" database, a teacher may teach zero or more classes, while a class is taught by one (and only one) teacher. In a "company" database, a manager manages zero or more employees, while an employee is managed by one (and only one) manager. In a "product sales" database, a customer may place many orders; while an order is placed by one particular customer. This kind of relationship is known as one-to-many.

Many-to-Many

In a "product sales" database, a customer's order may contain one or more products; and a product can appear in many orders. In a "bookstore" database, a book is written by one or more authors; while an author may write zero or more books. This kind of relationship is known as many-to-many.

One-to-One

In a "product sales" database, a product may have optional supplementary information such as image, more Description and comment. Keeping them inside the Products table results in many empty spaces (in those records without these optional data). Furthermore, these large data may degrade the performance of the database.

Build the physical relational database system using Forward Engineering Techniques for the above designed ER Model with MySQL Database Management System, Identify and enforce all necessary constraints on the tables which include primary key, foreign key constraints and domain constraints etc.

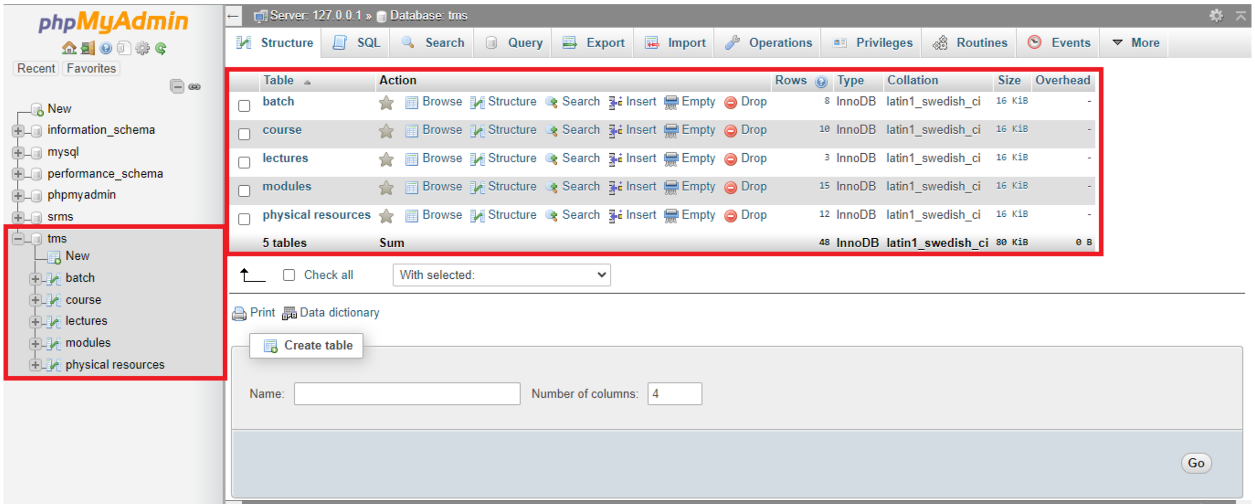
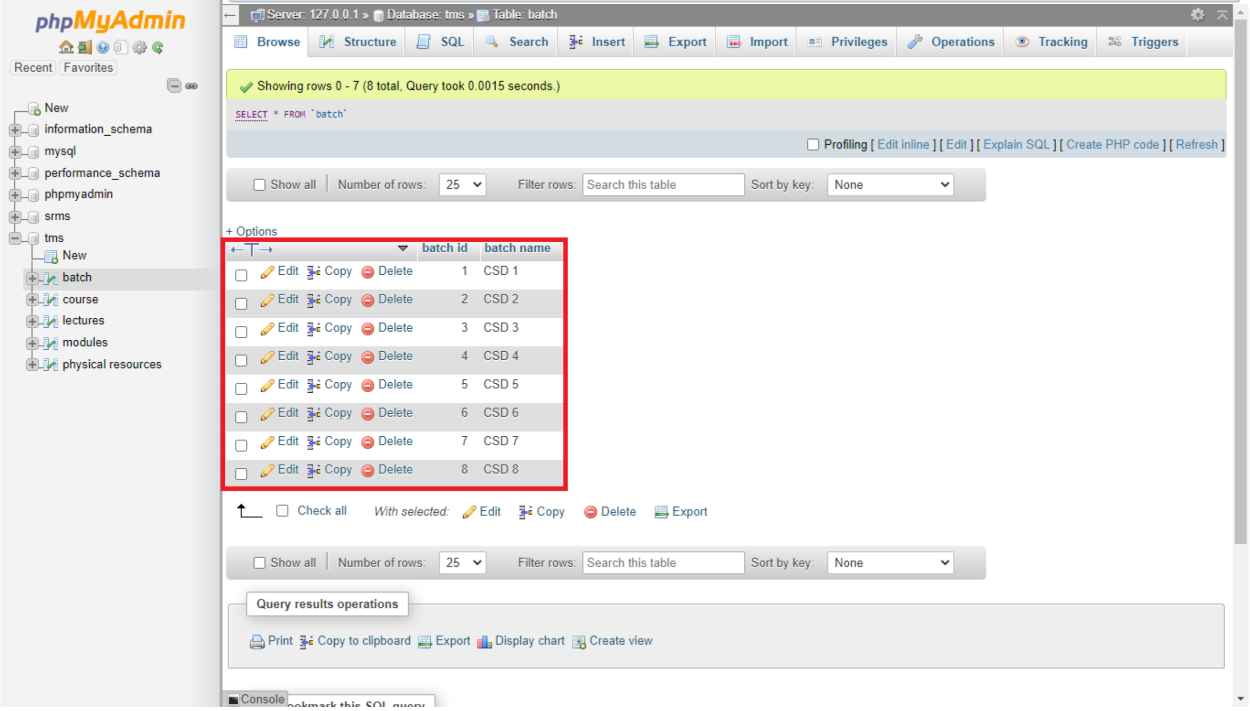


Table	Action	Rows	Type	Collation	Size	Overhead
batch	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16 KiB	-
course	Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16 KiB	-
lectures	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
modules	Browse Structure Search Insert Empty Drop	15	InnoDB	latin1_swedish_ci	16 KiB	-
physical resources	Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	16 KiB	-
5 tables	Sum	48	InnoDB	latin1_swedish_ci	80 KiB	0 B

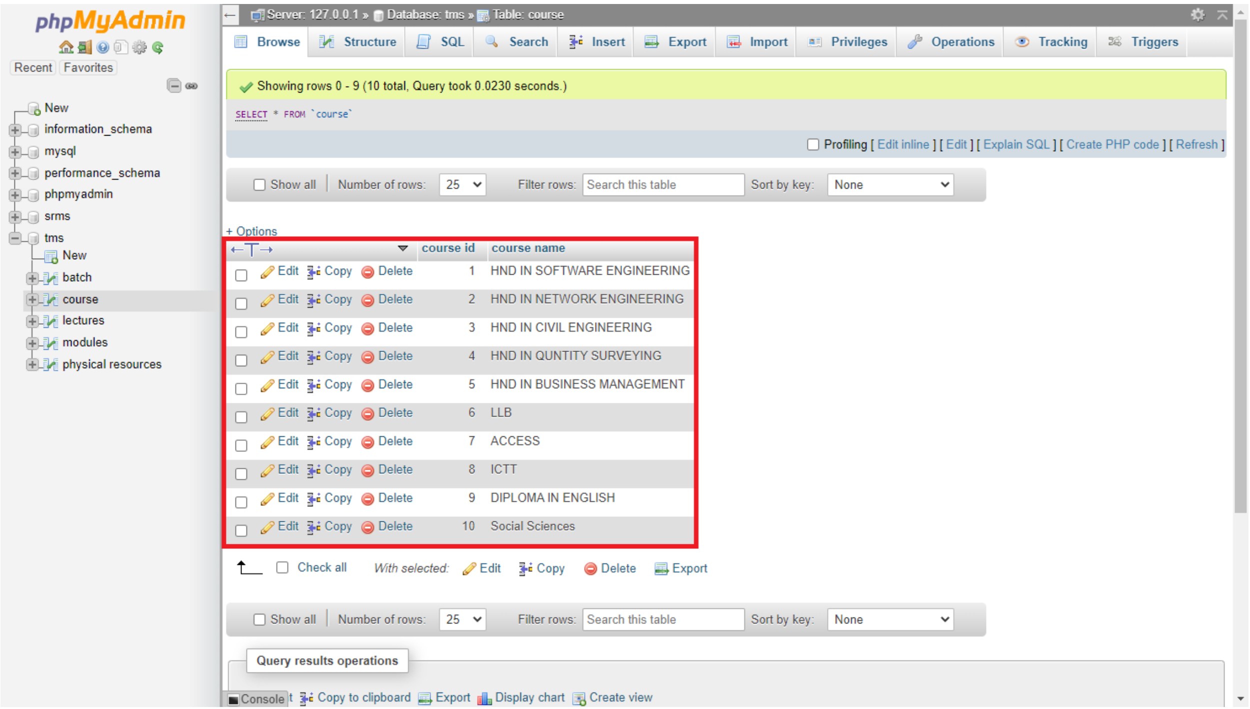
Name: □ Number of columns: 4



✓ Showing rows 0 - 7 (8 total, Query took 0.0015 seconds.)

```
SELECT * FROM `batch`
```

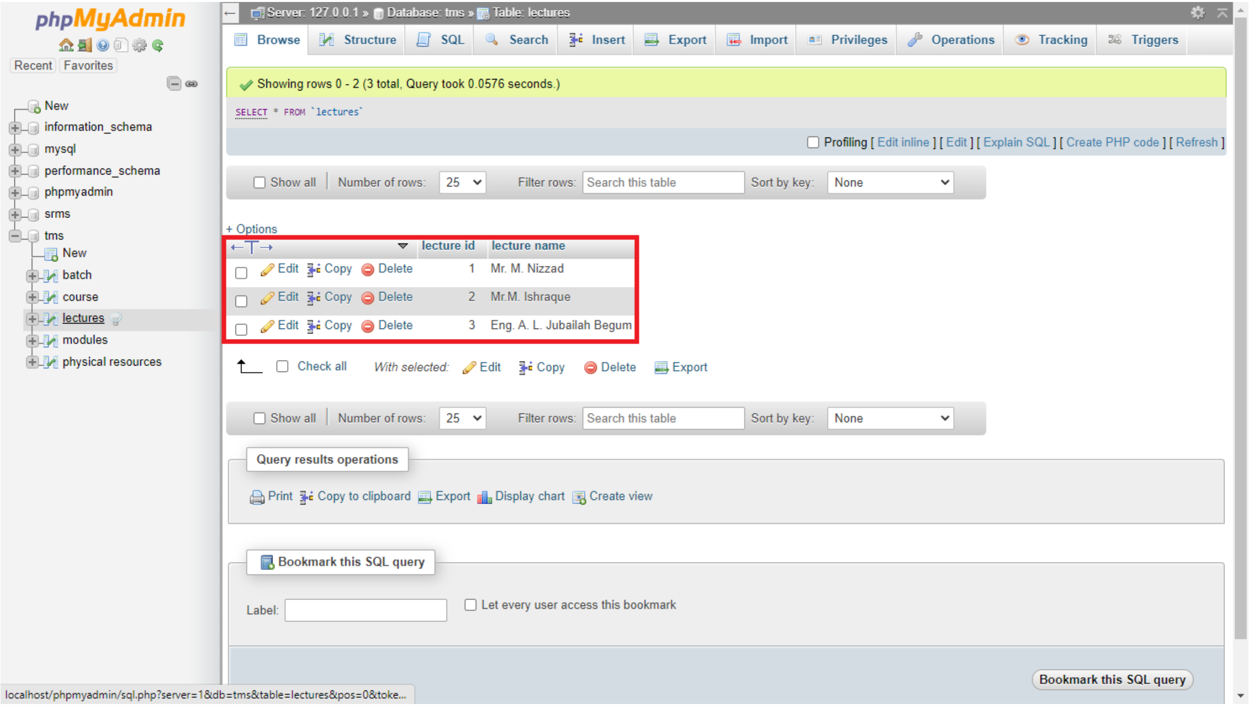
	batch id	batch name
Edit Copy Delete	1	CSD 1
Edit Copy Delete	2	CSD 2
Edit Copy Delete	3	CSD 3
Edit Copy Delete	4	CSD 4
Edit Copy Delete	5	CSD 5
Edit Copy Delete	6	CSD 6
Edit Copy Delete	7	CSD 7
Edit Copy Delete	8	CSD 8



Showing rows 0 - 9 (10 total, Query took 0.0230 seconds.)

```
SELECT * FROM `course`
```

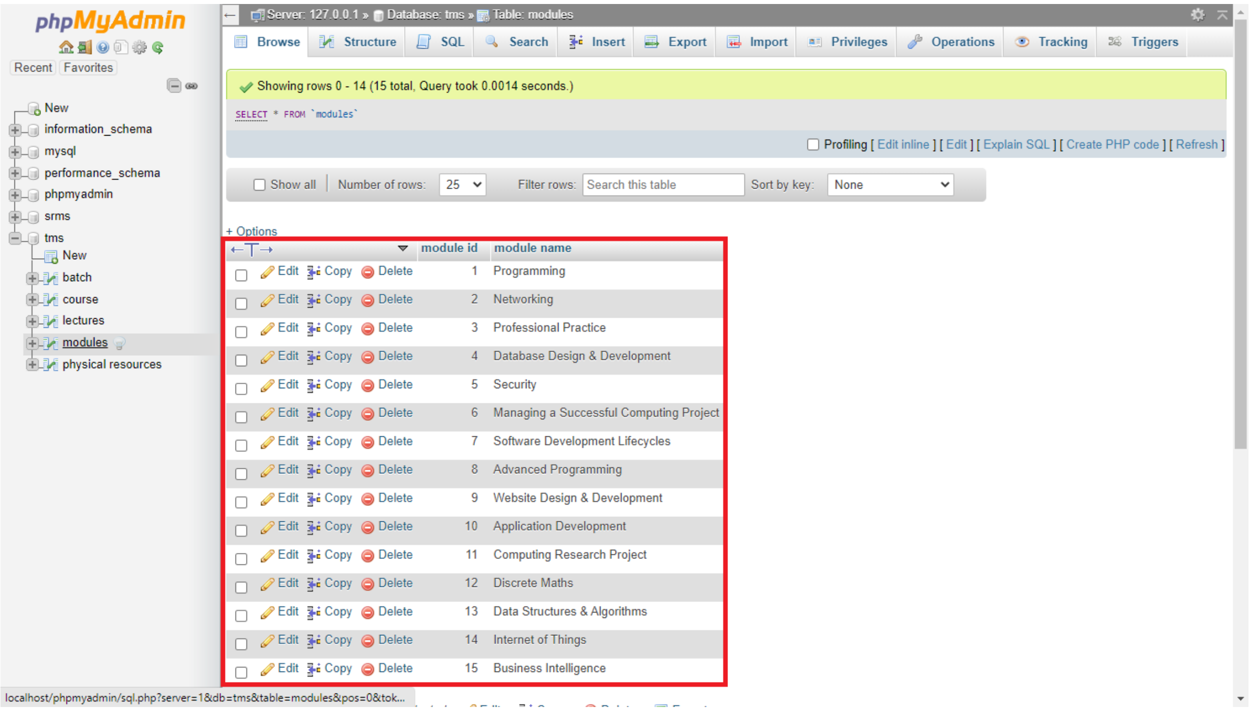
	course id	course name
☐ Edit Copy Delete	1	HND IN SOFTWARE ENGINEERING
☐ Edit Copy Delete	2	HND IN NETWORK ENGINEERING
☐ Edit Copy Delete	3	HND IN CIVIL ENGINEERING
☐ Edit Copy Delete	4	HND IN QUNTITY SURVEYING
☐ Edit Copy Delete	5	HND IN BUSINESS MANAGEMENT
☐ Edit Copy Delete	6	LLB
☐ Edit Copy Delete	7	ACCESS
☐ Edit Copy Delete	8	ICTT
☐ Edit Copy Delete	9	DIPLOMA IN ENGLISH
☐ Edit Copy Delete	10	Social Sciences



Showing rows 0 - 2 (3 total, Query took 0.0576 seconds.)

```
SELECT * FROM `lectures`
```

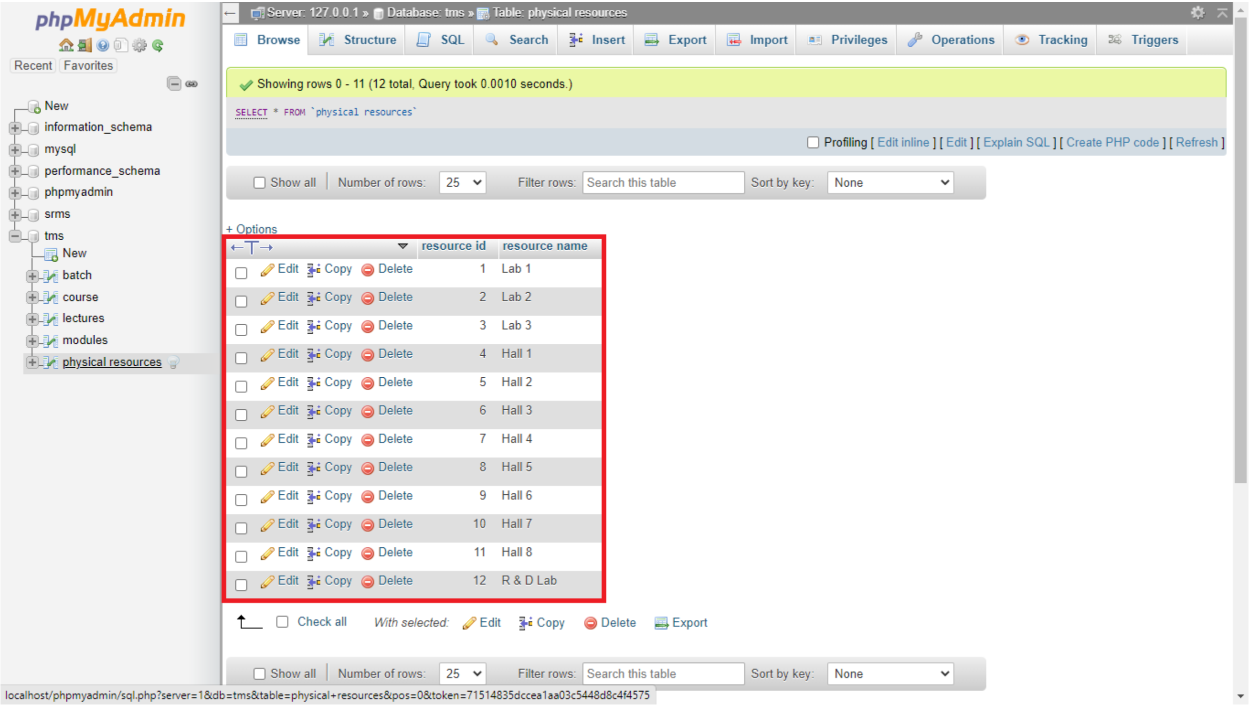
	lecture id	lecture name
☐ Edit Copy Delete	1	Mr. M. Nizzad
☐ Edit Copy Delete	2	Mr.M. Ishraque
☐ Edit Copy Delete	3	Eng. A. L. Jubailah Begum



Showing rows 0 - 14 (15 total, Query took 0.0014 seconds.)

```
SELECT * FROM `modules`
```

	module id	module name
Edit Copy Delete	1	Programming
Edit Copy Delete	2	Networking
Edit Copy Delete	3	Professional Practice
Edit Copy Delete	4	Database Design & Development
Edit Copy Delete	5	Security
Edit Copy Delete	6	Managing a Successful Computing Project
Edit Copy Delete	7	Software Development Lifecycles
Edit Copy Delete	8	Advanced Programming
Edit Copy Delete	9	Website Design & Development
Edit Copy Delete	10	Application Development
Edit Copy Delete	11	Computing Research Project
Edit Copy Delete	12	Discrete Maths
Edit Copy Delete	13	Data Structures & Algorithms
Edit Copy Delete	14	Internet of Things
Edit Copy Delete	15	Business Intelligence



Showing rows 0 - 11 (12 total, Query took 0.0010 seconds.)

```
SELECT * FROM `physical resources`
```

	resource id	resource name
Edit Copy Delete	1	Lab 1
Edit Copy Delete	2	Lab 2
Edit Copy Delete	3	Lab 3
Edit Copy Delete	4	Hall 1
Edit Copy Delete	5	Hall 2
Edit Copy Delete	6	Hall 3
Edit Copy Delete	7	Hall 4
Edit Copy Delete	8	Hall 5
Edit Copy Delete	9	Hall 6
Edit Copy Delete	10	Hall 7
Edit Copy Delete	11	Hall 8
Edit Copy Delete	12	R & D Lab

TASK:- 2.2

What the privilege system dose

Build the physical relational database system using Forward Engineering Techniques for the above designed ER Model with MySQL Database Management System, Identify and enforce all necessary constraints on the tables which include primary key, foreign key constraints and domain constraints etc. The primary function of the MySQL privilege system is to authenticate a user connecting from a given host, and to associate that user with privileges on a database such as select, insert, update, and delete.

Additional functionality includes the ability to have an anonymous user and to grant privileges for MySQL-specific functions such as LOAD DATA INFILE and administrative operations.

How the Privilege System Works

The MySQL privilege system ensures that all users may do exactly the things that they are supposed to be allowed to do. When you connect to a MySQL server, your identity is determined by the host from which you connect and the username you specify. The system grants privileges according to your identity and what you want to do.

MySQL considers both your hostname and username in identifying you because there is little reason to assume that a given username belongs to the same person everywhere on the Internet. For example, the user joe who connects from office.com need not be the same person as the user joe who connects from elsewhere.com. MySQL handles this by allowing you to distinguish users on different hosts that happen to have the same name: you can grant joe one set of privileges for connections from office.com, and a different set of privileges for connections from elsewhere.com.

MySQL access control involves two stages:

- Stage 1: The server checks whether you are even allowed to connect.
- Stage 2: Assuming you can connect, the server checks each request you issue to see whether you have sufficient privileges to perform it. For example, if you try to select rows from a table in a database or drop a table from the database, the server makes sure you have the select privilege for the table or the drop privilege for the database.

Scope fields determine the scope of each entry in the tables—that is, the context in which the entry applies. For example, a user table entry with Host and User values of 'thomas.loc.gov' and 'bob' would be used for authenticating connections made to the server by bob from the host thomas.loc.gov. Similarly, a db table entry with Host, User, and Db fields of 'thomas.loc.gov', 'bob', and 'reports' would be used when bob connects from the host thomas.loc.gov to access the reports database. The tables_priv and columns_priv tables contain scope fields indicating tables or table/column combinations to which each entry applies.

For access-checking purposes, comparisons of Host values are case-insensitive. User, Password, Db, and Table_name values are case-sensitive. Column_name values are case-insensitive in MySQL.

Privilege fields indicate the privileges granted by a table entry, that is, what operations can be performed. The server combines the information in the various grant tables to form a complete description of a user's privileges.

Privileges Provided by MySQL

Information about user privileges is stored in the user, db, host, tables_priv, and columns_priv tables in the MySQL database (that is, in the database named MySQL).

The select, insert, update, and delete privileges allow you to perform operations on rows in existing tables in a database.

SELECT statements require the select privilege only if they actually retrieve rows from a table. You can execute certain SELECT statements even without permission to access any of the databases on the server. For example, you could use the MySQL client as a simple calculator:

```
mysql> SELECT 1+1;

mysql> SELECT PI( )*2;
```

The index privilege allows you to create or drop (remove) indexes.

The alter privilege allows you to use ALTER TABLE.

The create and drop privileges allow you to create new databases and tables, or to drop (remove) existing databases and tables.

Note that if you grant the drop privilege for the MySQL database to a user, that user can drop the database in which the MySQL access privileges are stored!

The grant privilege allows you to give to other users those privileges you possess.

The file privilege gives you permission to read and write files on the server using the LOAD DATA INFILE and SELECT ... INTO OUTFILE statements. Any user to whom this privilege is granted can read or write any file that the MySQL server can read or write.

The reload command tells the server to re-read the grant tables. The refresh command flushes all tables and opens and closes the log files. flush-privileges is a synonym for reload. The other flush-* commands perform functions similar to refresh but are more limited in scope, and may be preferable in some instances. For example, if you want to flush just the log files, flush-logs is a better choice than refresh.

The shutdown command shuts down the server.

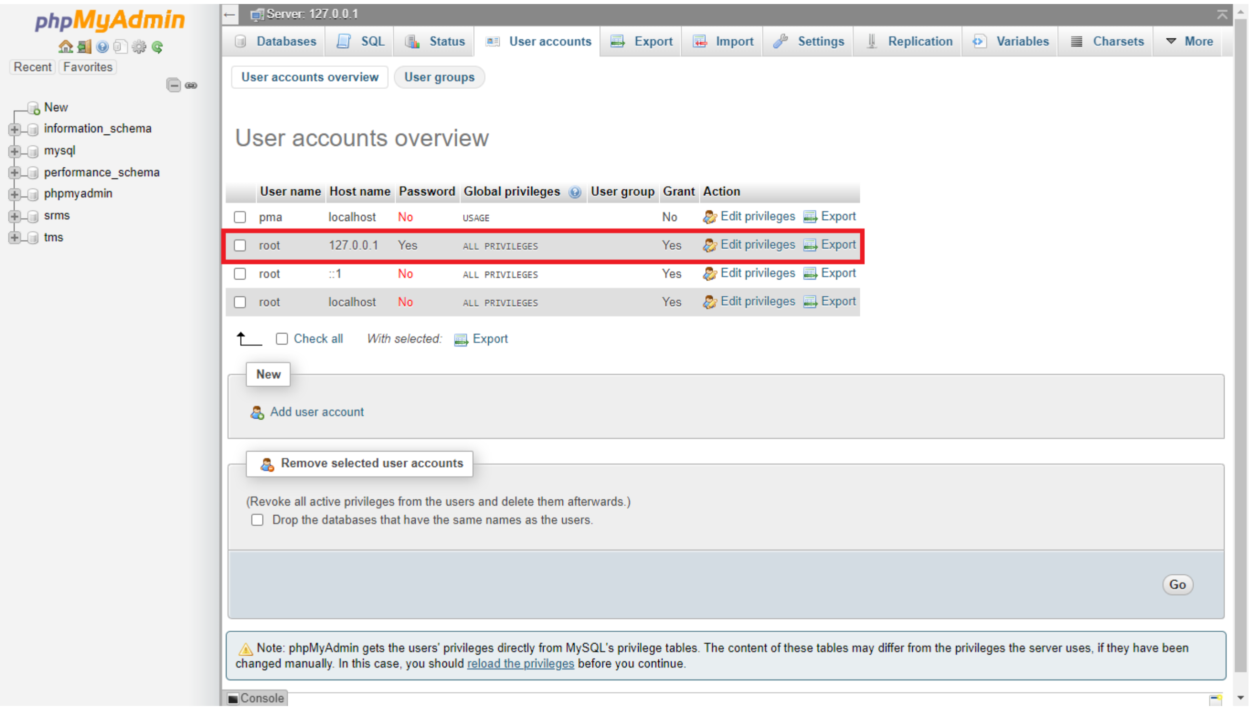
The process list command displays information about the threads executing within the server. The kill command kills server threads. You can always display or kill your own threads, but you need the process privilege to display or kill threads initiated by other users. See Section 4.5.5.

It is a good idea in general to grant privileges only to those users who need them, but you should exercise particular caution in granting certain privileges:

- The grant privilege allows users to give away their privileges to other users. Two users with different privileges and with the grant privilege are able to combine privileges.
- The alter privilege may be used to subvert the privilege system by renaming tables.
- The file privilege can be abused to read any world-readable file on the server into a database table, the contents of which can then be accessed using SELECT. This includes the contents of all databases hosted by the server!
- The shutdown privilege can be abused to deny service to other users entirely, by terminating the server.
- The process privilege can be used to view the plain text of currently executing queries, including queries that set or change passwords.
- Privileges on the MySQL database can be used to change passwords and other access privilege information. (Passwords are stored encrypted, so a malicious user cannot simply read them to know the plain text password.) If they can access the MySQL. User password column, they can use it to log into the MySQL server for the given user. (With sufficient privileges, the same user can replace a password with a different one.)

There are some things that you cannot do with the MySQL privilege system:

- You cannot explicitly specify that a given user should be denied access. That is, you cannot explicitly match a user and then refuse the connection.
- You cannot specify that a user has privileges to create or drop tables in a database but not to create or drop the database itself.



User accounts overview

	User name	Host name	Password	Global privileges	User group	Grant	Action
□	pma	localhost	No	USAGE		No	Edit privileges Export
□	root	127.0.0.1	Yes	ALL PRIVILEGES		Yes	Edit privileges Export
□	root	::1	No	ALL PRIVILEGES		Yes	Edit privileges Export
□	root	localhost	No	ALL PRIVILEGES		Yes	Edit privileges Export

□ Check all *With selected:* Export

New

Add user account

Remove selected user accounts

(Revoke all active privileges from the users and delete them afterwards.)

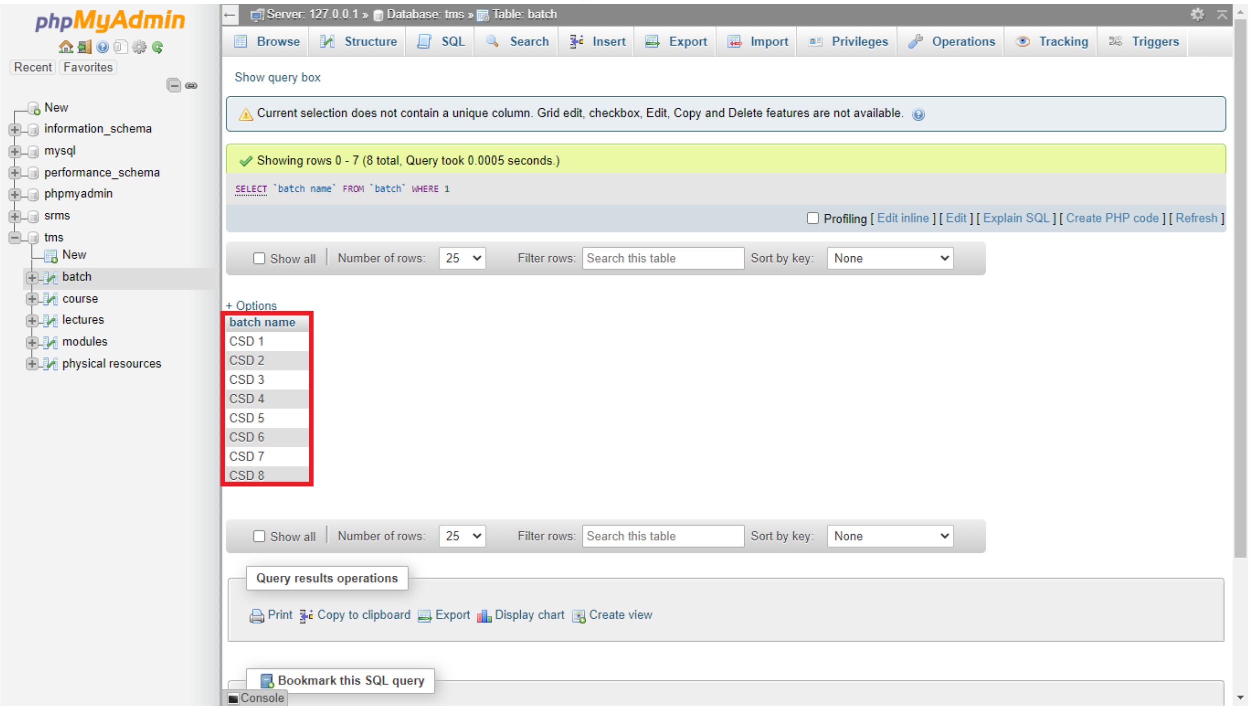
□ Drop the databases that have the same names as the users.

Go

Note: phpMyAdmin gets the users' privileges directly from MySQL's privilege tables. The content of these tables may differ from the privileges the server uses, if they have been changed manually. In this case, you should reload the privileges before you continue.

TASK:- 2.3

Write suitable queries to generate MIS reports and produce the output results.



Show query box

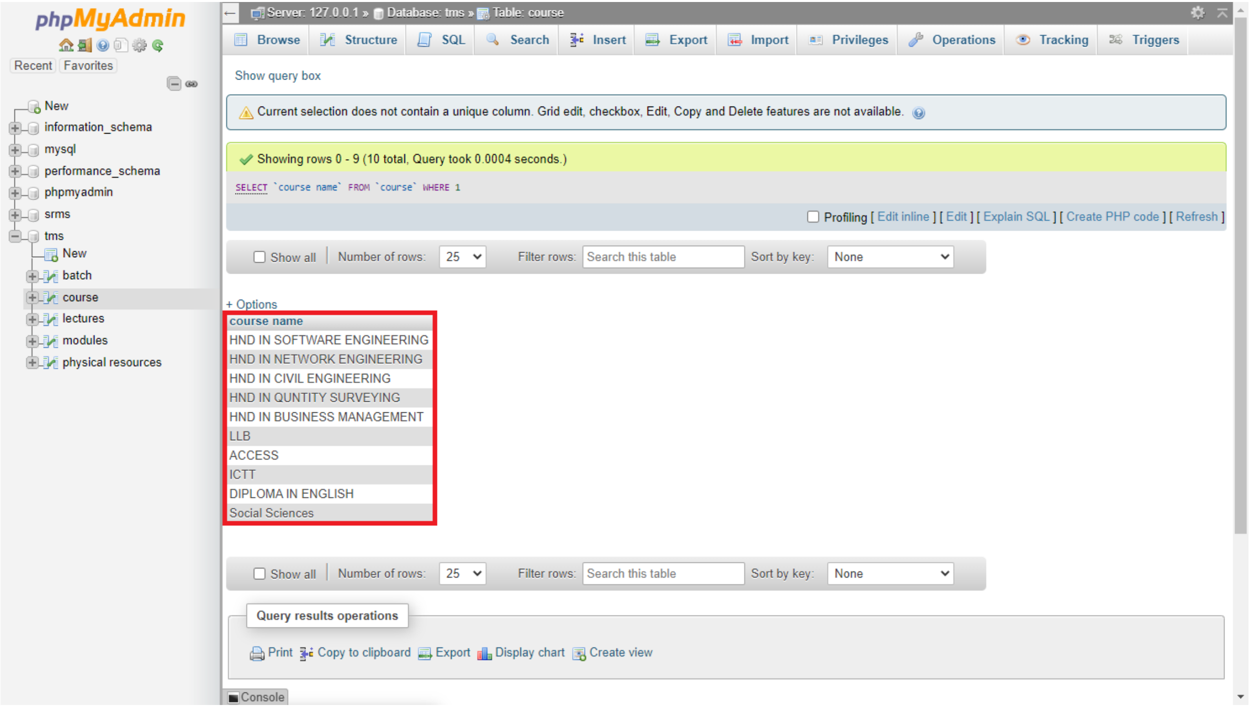
Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 7 (8 total, Query took 0.0005 seconds.)

```
SELECT `batch name` FROM `batch` WHERE 1
```

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

batch name
CSD 1
CSD 2
CSD 3
CSD 4
CSD 5
CSD 6
CSD 7
CSD 8



Show query box

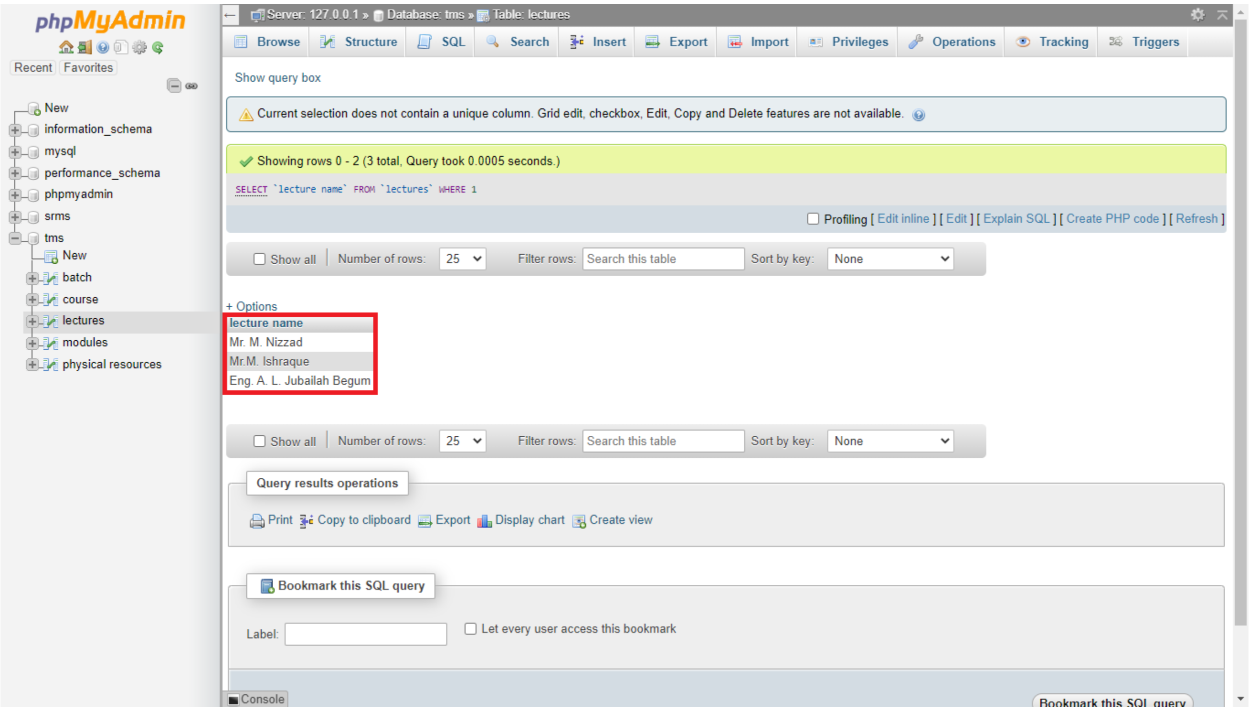
Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 9 (10 total, Query took 0.0004 seconds.)

```
SELECT `course name` FROM `course` WHERE 1
```

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

course name
HND IN SOFTWARE ENGINEERING
HND IN NETWORK ENGINEERING
HND IN CIVIL ENGINEERING
HND IN QUNTITY SURVEYING
HND IN BUSINESS MANAGEMENT
LLB
ACCESS
ICTT
DIPLOMA IN ENGLISH
Social Sciences



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

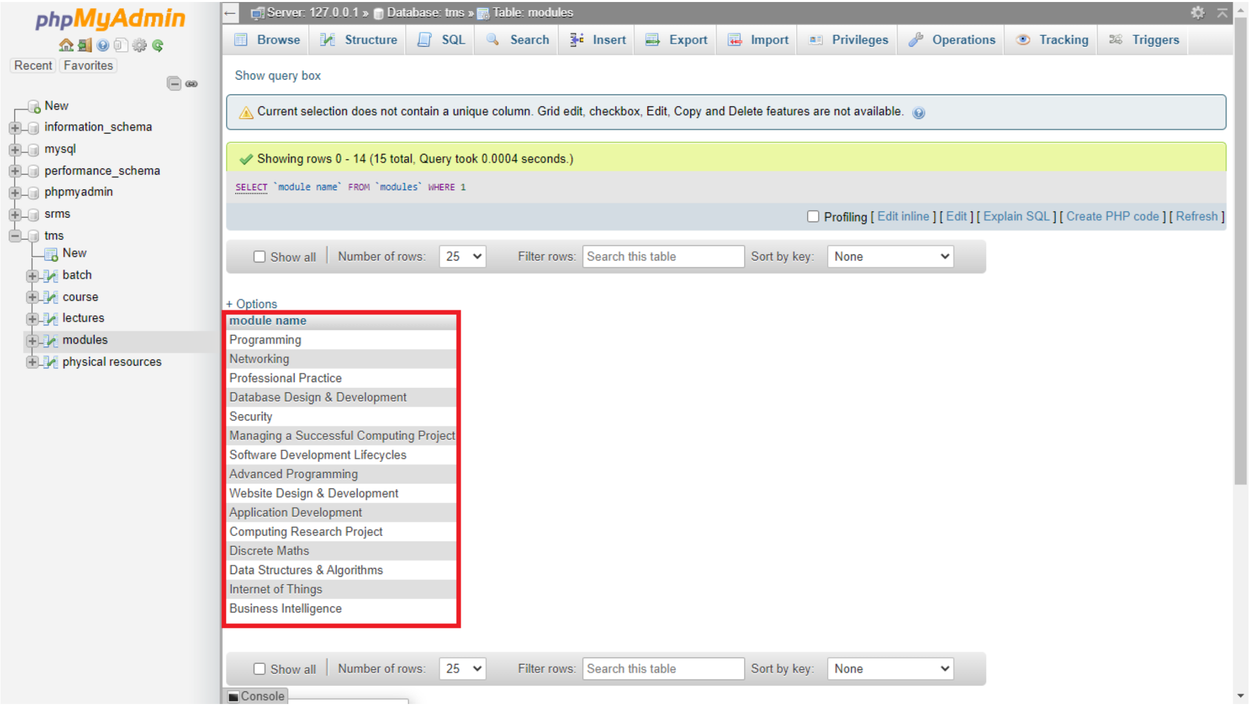
Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

```
SELECT `lecture name` FROM `lectures` WHERE 1
```

Number of rows: 25

Sort by key: None

lecture name
Mr. M. Nizzad
Mr.M. Ishraque
Eng. A. L. Jubailah Begum



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

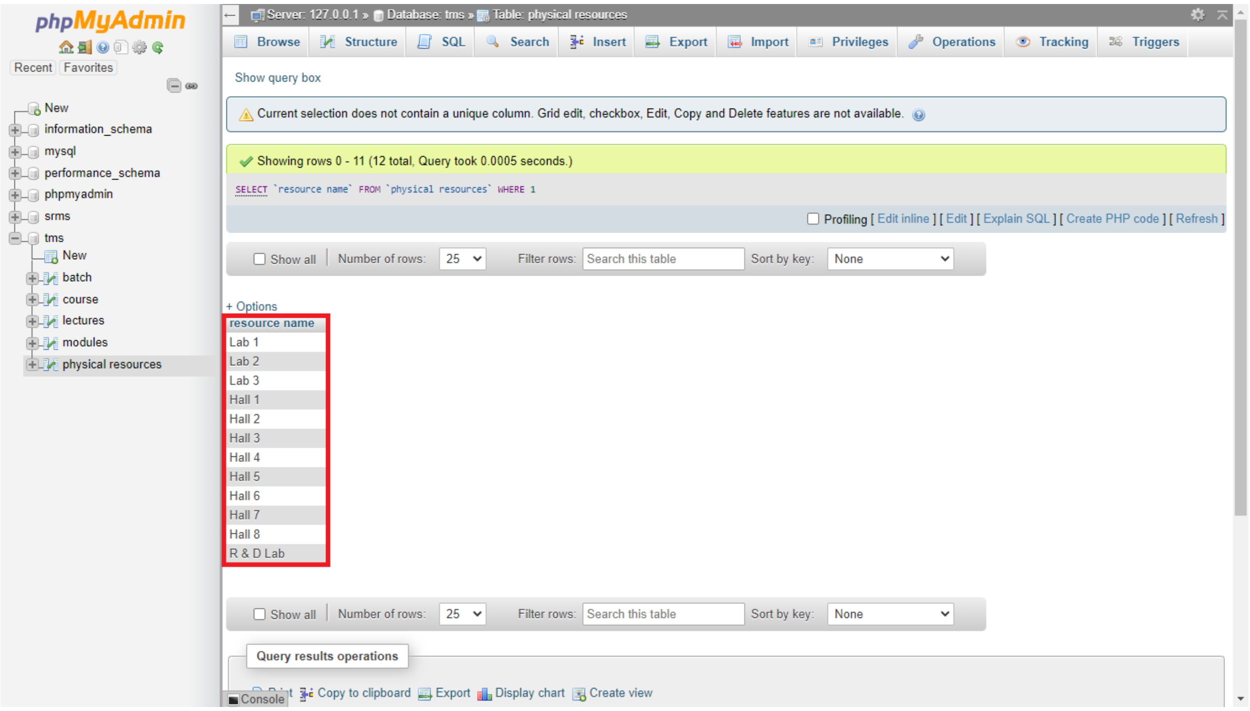
Showing rows 0 - 14 (15 total, Query took 0.0004 seconds.)

```
SELECT `module name` FROM `modules` WHERE 1
```

Number of rows: 25

Sort by key: None

module name
Programming
Networking
Professional Practice
Database Design & Development
Security
Managing a Successful Computing Project
Software Development Lifecycles
Advanced Programming
Website Design & Development
Application Development
Computing Research Project
Discrete Maths
Data Structures & Algorithms
Internet of Things
Business Intelligence



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 11 (12 total, Query took 0.0005 seconds.)

```
SELECT `resource name` FROM `physical resources` WHERE 1
```

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

resource name
Lab 1
Lab 2
Lab 3
Hall 1
Hall 2
Hall 3
Hall 4
Hall 5
Hall 6
Hall 7
Hall 8
R & D Lab

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

TASK:- 2.4, 2.5

What is database testing?

Database Testing is a type of software testing that checks the schema, tables, triggers, etc. of the Database under test. It also checks data integrity and consistency. It may involve creating complex queries to load/stress test the Database and check its responsiveness.

Why database testing is important?

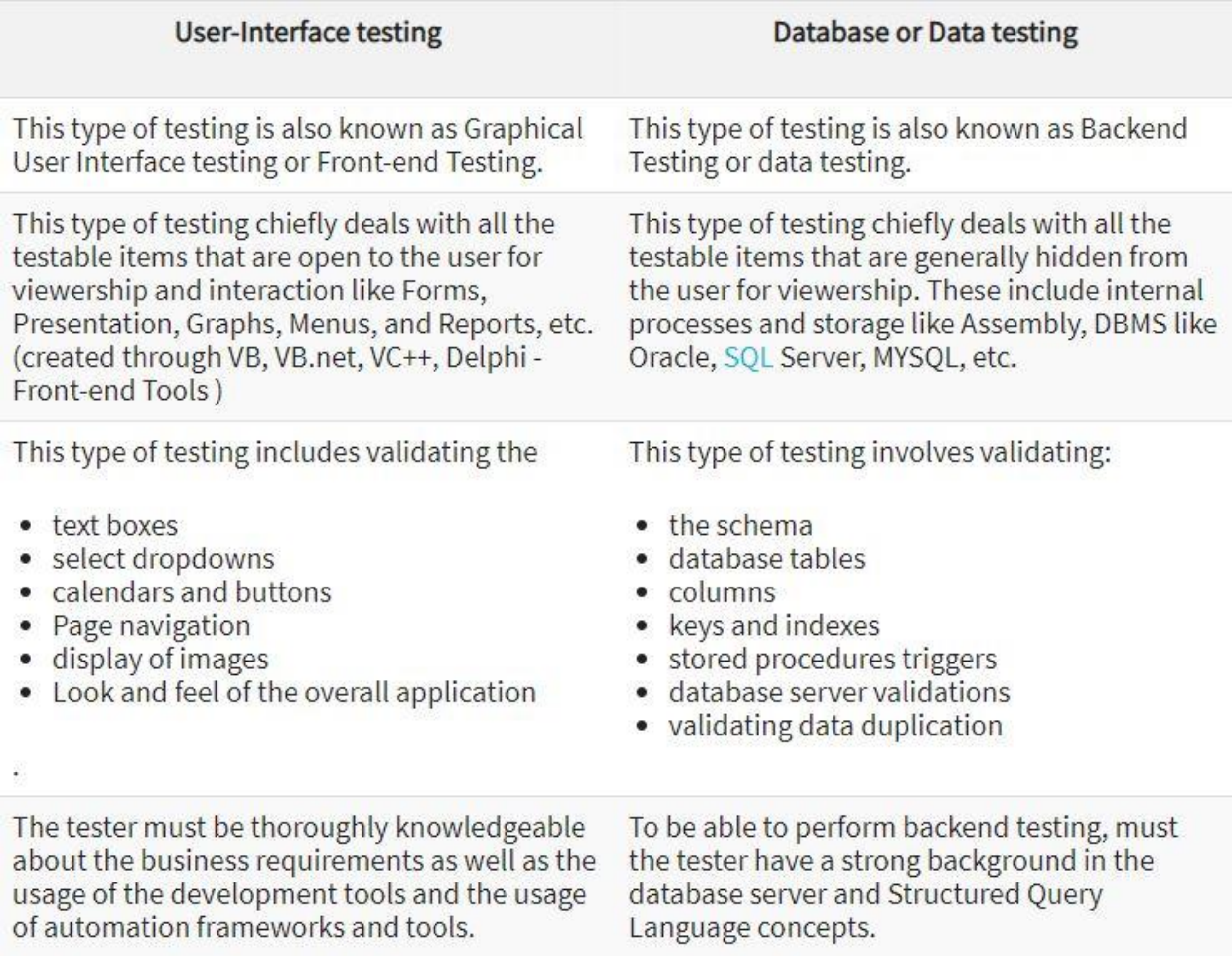
Database Testing is Important in software testing because it ensures data values and information received and stored into database are valid or not. Database testing helps to save data loss, saves aborted transaction data and no unauthorized access to the information. Database is important for any software application hence testers must have good knowledge of SQL for database testing.

The GUI is usually given the most emphasis by the test and development team members since the Graphical User Interface happens to be the most visible part of the application. However, what is also important is to validate the information that is the heart of the application, aka DATABASE.

Let us consider a Banking application wherein a user makes transactions. Now from Database Testing or DB Testing viewpoint following, things are important:

1. The application stores the transaction information in the application database and displays them correctly to the user.
2. No information is lost in the process.
3. No partially performed or aborted operation information is saved by the application.
4. No unauthorized individual is allowed to access the user's information.
5. To ensure all these above objectives, we need to use data validation or data testing.

Differences between User-Interface Testing and Data Testing

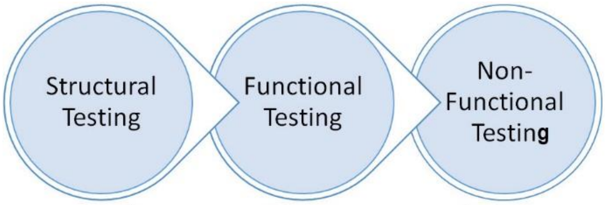


User-Interface testing	Database or Data testing
This type of testing is also known as Graphical User Interface testing or Front-end Testing.	This type of testing is also known as Backend Testing or data testing.
This type of testing chiefly deals with all the testable items that are open to the user for viewership and interaction like Forms, Presentation, Graphs, Menus, and Reports, etc. (created through VB, VB.net, VC++, Delphi - Front-end Tools)	This type of testing chiefly deals with all the testable items that are generally hidden from the user for viewership. These include internal processes and storage like Assembly, DBMS like Oracle, SQL Server, MYSQL, etc.
This type of testing includes validating the • text boxes • select dropdowns • calendars and buttons • Page navigation • display of images • Look and feel of the overall application .	This type of testing involves validating: • the schema • database tables • columns • keys and indexes • stored procedures triggers • database server validations • validating data duplication
The tester must be thoroughly knowledgeable about the business requirements as well as the usage of the development tools and the usage of automation frameworks and tools.	To be able to perform backend testing, must the tester have a strong background in the database server and Structured Query Language concepts.

Types of Database Testing

The 3 types of Database Testing are

1. Structural Testing
2. Functional Testing
3. Non-functional Testing



Types of Database Testing

Structural Database Testing

Structural Database Testing is a database testing technique that validates all the elements inside data repository that are mainly used for data storage and which are not allowed to be directly manipulated by end-users. The validation of database servers is also an important consideration in structural database testing. A successful completion of this testing needs mastery in SQL queries.

Database Table, Column Testing

Let us look into various checks for database and column testing.

1. Whether the mapping of the database fields and columns in the backend is compatible with those mappings in the front-end?
2. Validation of the length and naming convention of the database fields and columns as specified by the requirements.
3. Validation of the presence of any unused/unmapped database tables/columns.
4. Validation of the compatibility of the
    - datatype
    - field lengths

of the back-end database columns with that of those present at the front-end of the application.

5. Whether the database fields allow the user to provide desired user inputs as required by the business requirement specification documents.

Keys and indexes testing

Important checks for keys and indexes -

1. Check whether the required
    - Primary Key
    - Foreign Key

constraints have been created on the required tables.

2. Check whether the references for foreign keys are valid.
3. Check whether the data type of the primary key and the corresponding foreign keys are the same in the two tables.
4. Check whether the required naming conventions have been followed for all the keys and indexes.
5. Check the size and length of the required fields and indexes.
6. Whether the required
    - Clustered indexes
    - Non Clustered indexes

Database Server Validations

1. Check the database server configurations as specified by the business requirements.
2. Check the authorization of the required user to perform only those levels of actions that are required by the application.
3. Check that the database server is able to cater to the needs of the maximum allowed number of user-transactions as specified by the business requirement specifications.

Functional Database Testing

Functional Database Testing is a type of database testing that is used to validate the functional requirements of a database from the end-user's perspective. The main goal of functional database testing is to test whether the transactions and operations performed by the end-users which are related to the database works as expected or not.

Following are the basic conditions that need to be observed for database validations.

- Whether the field is mandatory while allowing NULL values on that field?
- Whether the length of each field is of sufficient size?
- Whether all similar fields have the same names across tables?
- Whether there are any computed fields present in the Database?

This particular process is the validation of the field mappings from the end-user viewpoint. In this particular scenario, the tester would perform an operation at the database level and then would navigate to the relevant user interface item to observe and validate whether the proper field validations have been carried out or not.

The vice versa condition whereby, first operation is carried out by the tester at the user interface, and then the same is validated from the back end is should also be done.

Non-functional testing

Non-functional testing in the context of database testing can be categorized into various categories as required by the business requirements. These can be load testing, Stress Testing, Security Testing, Usability Testing, and Compatibility Testing, and so on. The load testing, as well as stress testing, which can be grouped under the gamut of Performance Testing serves two specific purposes when it comes to the role of non-functional testing.

Risk quantification- Quantification of risk helps the stakeholders to ascertain the various system response time requirements under required levels of load. This is the original intent of any quality assurance task. We need to note that load testing does not mitigate risk directly, but through the processes of risk identification and risk quantification, presents corrective opportunities and an impetus for remediation that will mitigate risk.

Minimum system equipment requirement- The minimum system configuration that will allow the system to meet the formally stated performance expectations of stakeholders. So that extraneous hardware, software, and the associated cost of ownership can be minimized. This particular requirement can be categorized as the overall business optimization requirement.

PART :- 3

TASK :- 3.1

User and technical manual

In this part is explained about time table management system database this data base is mainly focused by physical resource, lecture ,module course and batch this database is apply privilege Metrix

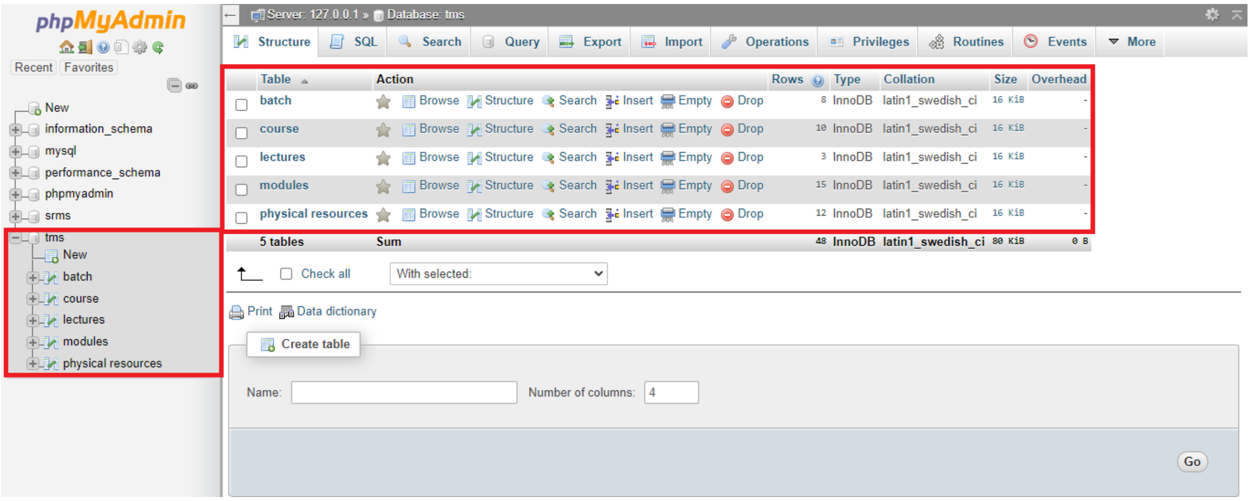
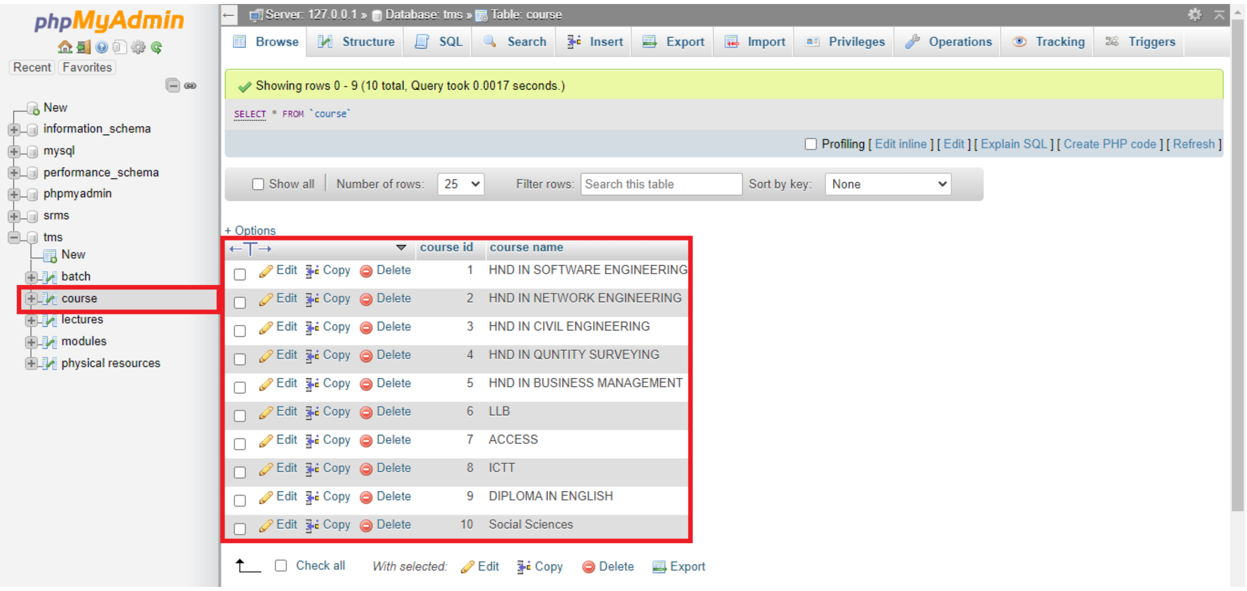


Table	Action	Rows	Type	Collation	Size	Overhead
batch	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16 KiB	-
course	Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16 KiB	-
lectures	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
modules	Browse Structure Search Insert Empty Drop	15	InnoDB	latin1_swedish_ci	16 KiB	-
physical resources	Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	16 KiB	-
5 tables	Sum	48	InnoDB	latin1_swedish_ci	80 KiB	0 B

This is my data base interface I have 5 table is are created. first one is course table, batch table, module table, lecture table and physical resource table.

Course table



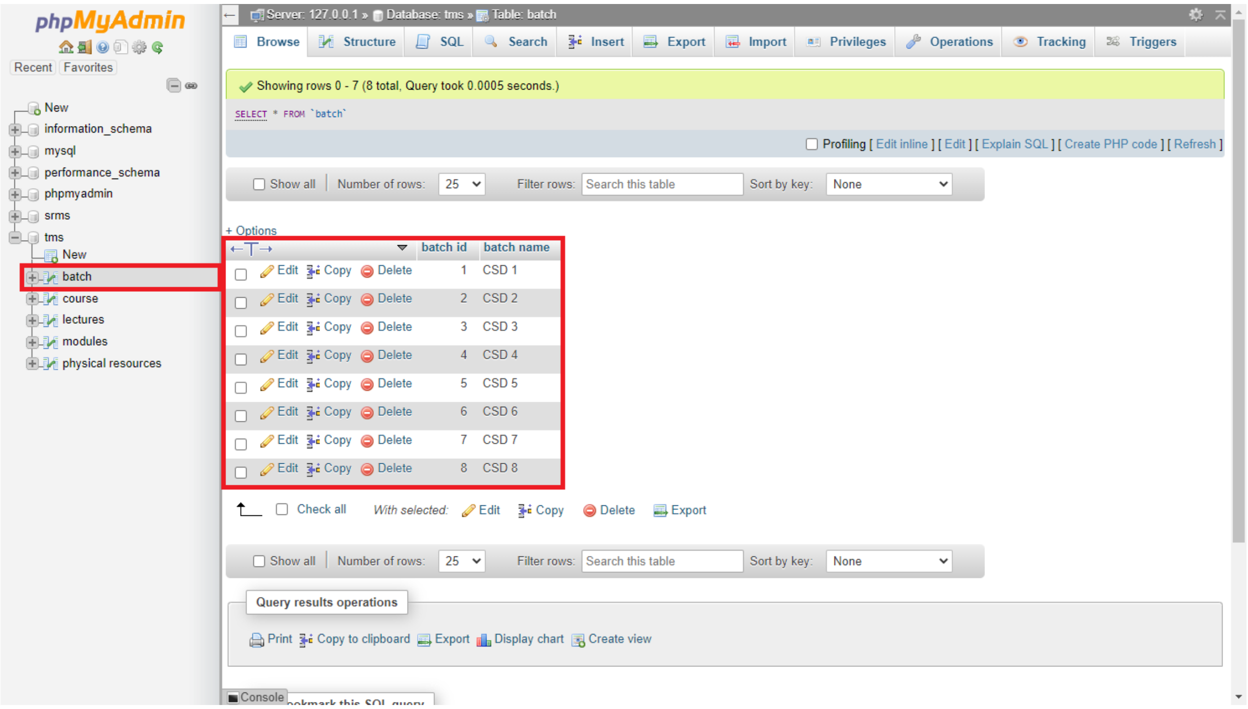
Showing rows 0 - 9 (10 total, Query took 0.0017 seconds.)

```
SELECT * FROM `course`
```

	course id	course name
Edit Copy Delete	1	HND IN SOFTWARE ENGINEERING
Edit Copy Delete	2	HND IN NETWORK ENGINEERING
Edit Copy Delete	3	HND IN CIVIL ENGINEERING
Edit Copy Delete	4	HND IN QUNTITY SURVEYING
Edit Copy Delete	5	HND IN BUSINESS MANAGEMENT
Edit Copy Delete	6	LLB
Edit Copy Delete	7	ACCESS
Edit Copy Delete	8	ICTT
Edit Copy Delete	9	DIPLOMA IN ENGLISH
Edit Copy Delete	10	Social Sciences

This table you can add course name and course id.

Batch table


phpMyAdmin
Recent Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
srms
tms
New
batch
course
lectures
modules
physical resources
Server: 127.0.0.1 » Database: tms » Table: batch
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
Check all With selected: Edit Copy Delete Export
Query results operations
Print Copy to clipboard Export Display chart Create view
Console

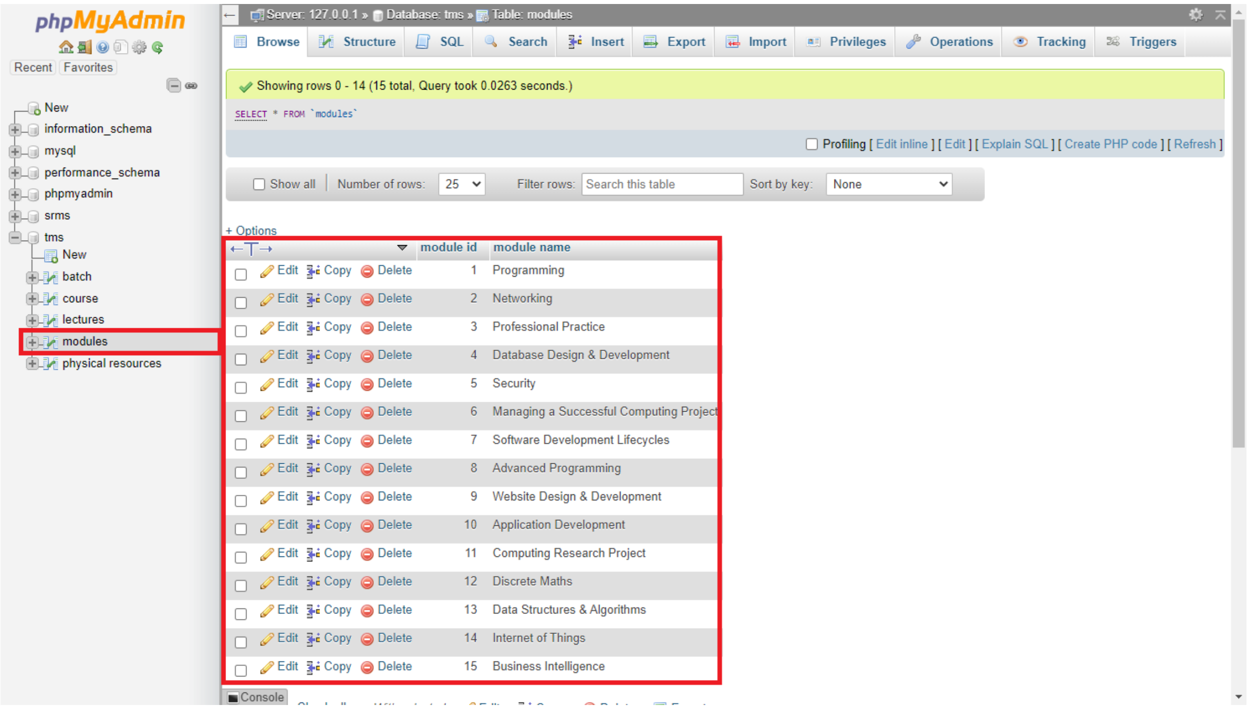
Showing rows 0 - 7 (8 total, Query took 0.0005 seconds.)

```
SELECT * FROM `batch`
```

batch id	batch name
1	CSD 1
2	CSD 2
3	CSD 3
4	CSD 4
5	CSD 5
6	CSD 6
7	CSD 7
8	CSD 8

This is batch details you can add your batch details.

Module table


phpMyAdmin
Recent Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
srms
tms
New
batch
course
lectures
modules
physical resources
Server: 127.0.0.1 » Database: tms » Table: modules
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
Console

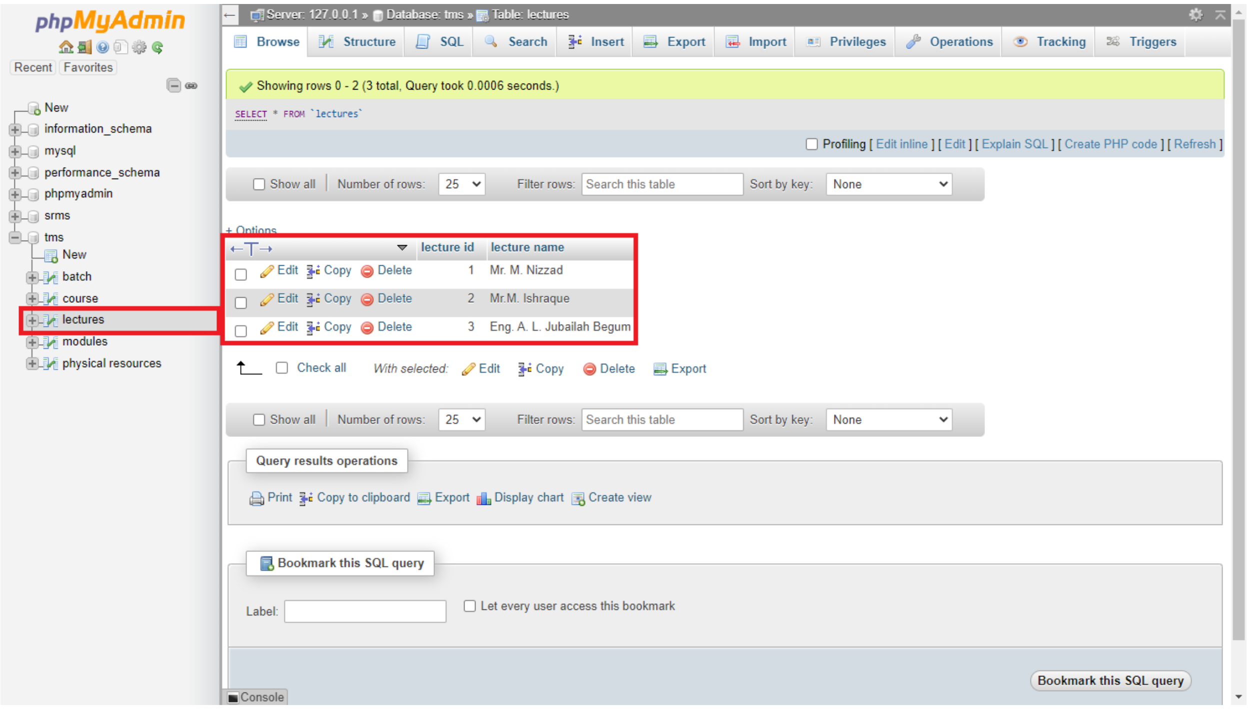
Showing rows 0 - 14 (15 total, Query took 0.0263 seconds.)

```
SELECT * FROM `modules`
```

module id	module name
1	Programming
2	Networking
3	Professional Practice
4	Database Design & Development
5	Security
6	Managing a Successful Computing Project
7	Software Development Lifecycles
8	Advanced Programming
9	Website Design & Development
10	Application Development
11	Computing Research Project
12	Discrete Maths
13	Data Structures & Algorithms
14	Internet of Things
15	Business Intelligence

This is the module table you can add module name and ect.

Lecture table



Server: 127.0.0.1 » Database: tms » Table: lectures

Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```
SELECT * FROM `lectures`
```

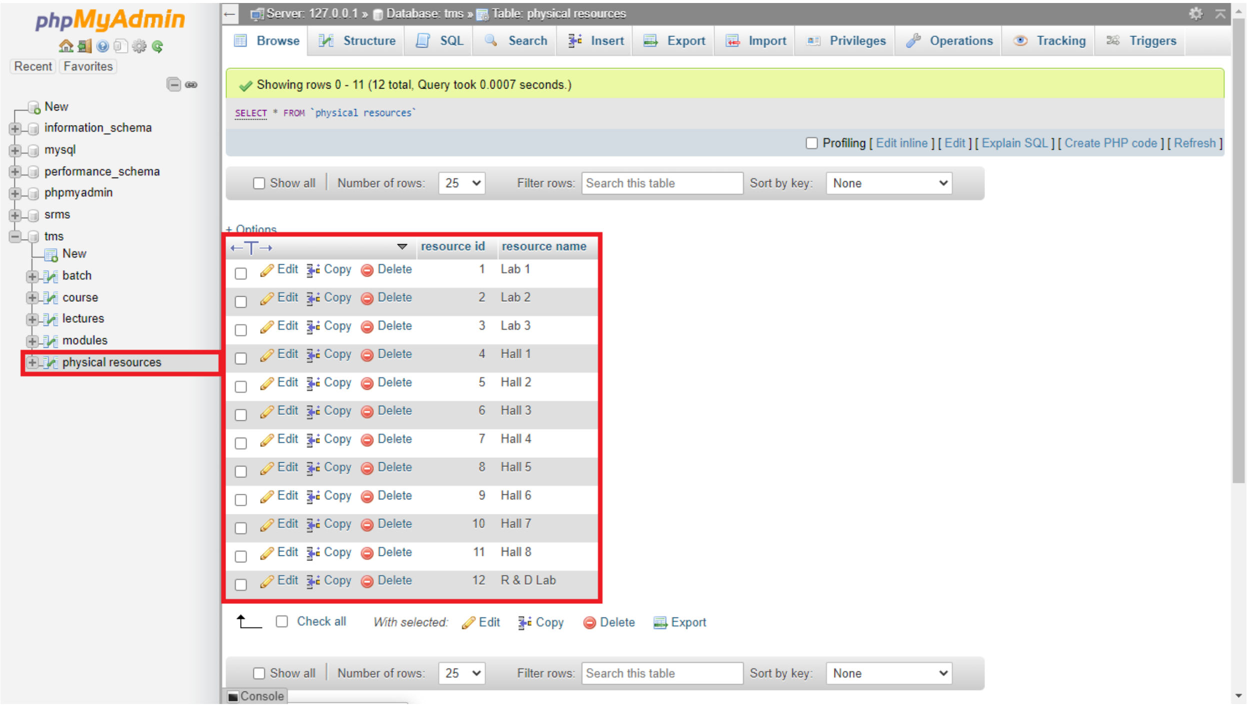
□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	lecture id	lecture name
□ Edit Copy Delete	1	Mr. M. Nizzad
□ Edit Copy Delete	2	Mr.M. Ishraque
□ Edit Copy Delete	3	Eng. A. L. Jubailah Begum

Label:

This is the lecture table you can add lecture name and id and ect.

Physical resource



Server: 127.0.0.1 » Database: tms » Table: physical resources

Showing rows 0 - 11 (12 total, Query took 0.0007 seconds.)

```
SELECT * FROM `physical resources`
```

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	resource id	resource name
□ Edit Copy Delete	1	Lab 1
□ Edit Copy Delete	2	Lab 2
□ Edit Copy Delete	3	Lab 3
□ Edit Copy Delete	4	Hall 1
□ Edit Copy Delete	5	Hall 2
□ Edit Copy Delete	6	Hall 3
□ Edit Copy Delete	7	Hall 4
□ Edit Copy Delete	8	Hall 5
□ Edit Copy Delete	9	Hall 6
□ Edit Copy Delete	10	Hall 7
□ Edit Copy Delete	11	Hall 8
□ Edit Copy Delete	12	R & D Lab

This is the resource table you can add you eligible physical resource and ect.

TASK :- 3.2

Evaluation Database

Before you refine the data structure diagram, you need to evaluate the design for performance. To satisfy performance requirements for each individual business transaction, you need to consider the following issues:

- **Input/output (I/O) performance -** Is the number of I/O operations performed against the database sufficiently low to provide satisfactory transaction performance?
- **CPU time -** Does the structure of the physical database optimize the use of CPU processing?
- **Space management -** Do design choices help to conserve storage resources?

Once you have refined the database to satisfy each individual transaction, you need to determine how the system will be affected by the concurrent execution of several transactions. To avoid excessive contention for database resources, you need to make appropriate changes to the physical model.

Refining the database design

Like many other database design procedures, refining the database design is an iterative process, as shown below. As you refine the design, you need to evaluate the design for performance. When you make changes, you should review the design to ensure that it will optimize processing for all critical transactions and also minimize the likelihood of contention.

Improvement of database system

Is your database server healthy?

First and foremost, make sure that the host that's serving your database process has sufficient resources available. This includes CPU, memory, and disk space.

CPU

CPU will most likely not be a bottleneck, but database servers induce continuous base load on machines. To keep the host responsive, make sure that it has at the very least two CPU cores available.

I will assume that at least some of your hosts are virtualized. As a general rule of thumb, when monitoring virtual machines, also monitor the virtual host that the machines run on. CPU metrics of individual virtual machines won't show you the full picture. Numbers like CPU ready time are of particular importance, as they tell you, how much CPU time the virtual machines are really able to utilize.

Besides the technical implications, the number of CPUs also has impact on licensing. You might want to make sure that your database server license closely matches your CPU count. Otherwise, you might be overspending or not be able to fully leverage on your hardware.

Memory

Keep in mind that memory usage is not the only metric to keep an eye on. Memory usage does not tell you how much additional memory may be needed. The important number to look at is page faults per seconds.

Having thousands of page faults per second indicates that your hosts are out of memory (this is when you start to hear your server's hard drive grinding away).

Disk space

Because of indices and other performance improvements, databases use up a LOT more disk space than what the actual data itself requires (indices, you know). NoSQL databases in particular (Cassandra and MongoDB for instance) eat up a lot more disk space than you would expect. And while MongoDB might take up less RAM than a common SQL database, it's a real disk space hog.

I can't emphasize this too much: make sure you have lots of storage available for your database server. Also, make sure your database runs on dedicated hard drives, as this should keep disk fragmentation caused by other processes to a minimum.

One number to keep an eye on is disk latency. Depending on hard drive load, disk latency will increase, leading to a decrease in database performance. What can you do about this? Firstly, try to leverage your application's and database's caching mechanisms as much as possible. There is no quicker and more cost-effective way of moving the needle.

If that still does not yield the expected performance, you can always add additional hard drives. Read performance can be multiplied by simply mirroring your hard drives. Write performance really benefits from using RAID 1 or RAID 10 instead of, let's say, RAID 6. If you want to get your hands dirty on this subject

If that still doesn't help, give Solid State Drives a try. Make sure you pick a model that's designed for database usage, as databases will apply more read/write cycles to storage than most common applications. Despite decreasing prices, they still come at a higher price than traditional hard disks. But if you need the boost in performance, they are worth the price.

Who is accessing the database?

Once your database is residing on healthy hardware you should take a look at which applications are actually accessing the database. If one of your applications or services suffers from bad database performance, do not jump to the conclusion that you know which application or service is responsible for the bad performance.

When talking about inferior database performance, you're really talking about two different things. On one hand, the database as a whole may be affected. On the other hand, there may be just a single client that's experiencing bad performance.

If all of the database's clients experience bad performance, go back and check if your host is truly healthy. Chances are that your hardware is not up to the challenge. If there is only a single service that's suffering from bad database response times, dig deeper into that service's metrics to find out what's causing the problem.

Understand the load and individual response time of each service instance

If an individual service is having bad database performance, you should take a deeper look into the service’s communication with the database. Which queries are executed? How often are the queries executed per request? How many rows do they return?

Also, if you’re running multiple instances of a single service (and you really should), you should check whether all instances are affected, or just single on

It’s important to know that issues that materialize on the database level may be rooted elsewhere. Very often there is an issue related to the way a database is accessed.

Look at how often queries are called per request. Maybe you can reduce the number of actual database queries by improving the database cache of your service. Question everything. Is there any reason why a single query should be executed more than once per request? If there is, maybe you can unlock some potential performance by applying smart caching strategies.

Do you have enough database connections?

Even if the way you query your database is perfectly fine, you may still experience inferior database performance. If this is your situation, it’s time to check that your application’s database connection is correctly sized.

When configuring a connection pool there are two things to consider:

1. What is the maximum number of connections the database can handle?
2. What is the correct size connection pool required for your application?

Why shouldn’t you just set the connection pool size to the maximum? Because your application may not be the only client that’s connected to the database. If your application takes up all the connections, the database server won’t be able to perform as expected. However, if your application is the only client connected to the database, then go for it!

How to find out the maximum number of connections

You already confirmed in Step #1 that your database server is healthy. The maximum number of connections to the database is a function of the resources on the database. So to find the maximum number of connections, gradually increase load and the number of allowed connections to your database. While doing this, keep an eye on your database server’s metrics. Once they max out—either CPU, memory, or disk performance—you know you’ve reached the limit. If the number of available connections you reach is not enough for your application, then it’s time to consider upgrading your hardware.

Determine the correct size for your application’s connection pool

The number of allowed concurrent connections to your database is equivalent to the amount of parallel load that your application applies to the database server. There are tools available to help you in determining the correct number here.

Increasing load will lead to higher transaction response times, even if your database server is healthy. Measure the transaction response time from end-to-end to see if Connection Acquisition time takes up increasingly more time under heavy load. If it does, then you know that your connection pool is exhausted. If it doesn’t, have another look at your database server’s metrics to determine the maximum number of connections that your database can handle.

By the way, a good rule of thumb to keep in mind here is that a connection pool’s size should be constant, not variable. So set the minimum and maximum pool sizes to the same value.

Don’t forget about the network

We tend to forget about the physical constraints faced by our virtualized infrastructure. Nonetheless, there are physical constraints: cables fail and routers break. Unfortunately, the gap between works and doesn’t work usually varies. This is why you should keep an eye on your network metrics. If problems suddenly appear after months or even years of operating flawlessly, chances are that your infrastructure is suffering from a non-virtual, physical problem. Check your routers, check your cables, and check your network interfaces. It’s best to do this as early as possible following the first sign that there may be a problem because this may be the point in time when you can fix a problem before it impacts your business.

Very often, over-stressed processes start to drop packets due to depleted resources. Just in case your network issue is not a hardware problem, process level visibility can definitely come in handy in identifying a failing component.

Conclusion

In conclusion this assignment has more detailed information about the database design & development concept. As a student I have I have gained a lot of knowledge about database design & development and other essentials things about database concept.

In this assignment I have done lot things which are very to improve my knowledge. Specially I have a done a draw the ER diagram.

This assignment was very helpful for me to create and understand database which is used for primality test. I was able to create sql query which does the same thing in different perspective. But some are fast and quick.

Finally I express my deepest thanks to the assessor. Mr. Mohamed Nizzad. who gave this awesome opportunity to my database design & development knowledge.

Reference

https://www.dynatrace.com/news/blog/five-easy-steps-improve-database-performance/

https://techdocs.broadcom.com/us/en/ca-mainframe-software/database-management/ca-idms/19-0/administrating/administrating-database-design/refining-the-database-design/evaluating-the-database-design.html

https://www3.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html

https://www.oreilly.com/library/view/mysql-reference-manual/0596002653/ch04s02.html

https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html

https://www.guru99.com/data-testing.html