

Assignment 1

Date of Completion:

Date of Submission:

Title: MySQL

Problem Statement: Study of opensource relational database : MySQL

Objective: To learn & understand the basic database Architecture & the various components of it.

S/W & H/W Requirements: MySQL , windows 10, Mouse , keyboard, monitor

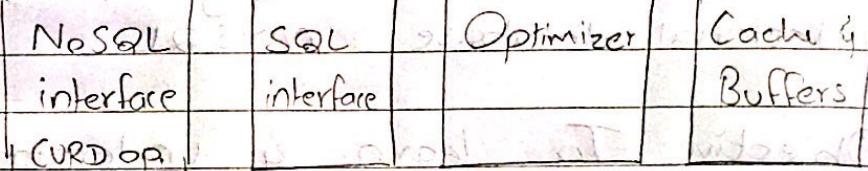
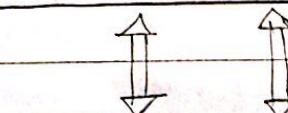
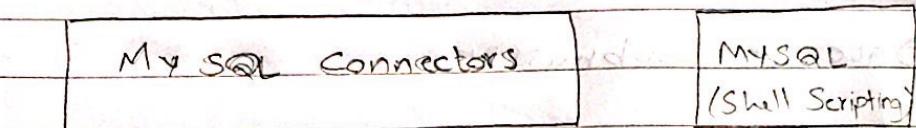
Theory:

MySQL is a fast easy to use RDBMS lang. for small & big businesses. It is developed marketed & supported by MySQL AB , which is a Swedish Company.

Features:

- 1) Client / Server Architecture.
- 2) Supports Views , stored procedures & triggers.
- 3) Supports many APIs & libraries for development
- 4) Platform independent.

- 5) Pluggable storage engine architecture  
 6) Connectors Eg:- .NET, ODBC, JDBC, etc.

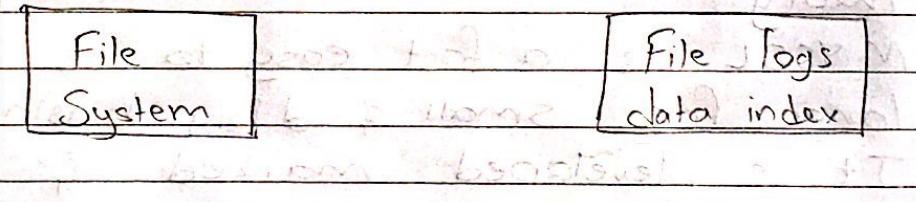


### Storage Engines

Memory Engine Relational &

Data Base Management

InnoDB, memory



MySQL Architecture

Application layer & Interfaces / End users.

1. It is a layer where client & user interact with the MySQL RDBMS.
2. There are different end users like naive users, programmers, sophisticated users, admins, etc.

Query Processor:

1. DDL interpreter
2. DML Compiler
3. Query Evaluation.

Storage Manager:

Interface between low level data & Applications.

- a) Authorization & integrity Manager
- b) Transaction Manager
- c) File manager
- d) Buffer manager.

4. Disk Storage

It contains the following Components:

- 1) Data Files
- 2) Data dictionary - Structure
- 3) Indices - Provides faster retrieval.

• Conclusion:

- 1) We studied Architecture of MySQL
- 2) We explored the different Features of MySQL
- 3) We studied the structure of a database management system.
- 4) We learnt the Functions of different components of DBMS.

## Assignment - 2

Date of Completion:

Date of Submission:

Title: Implement DDL commands in context of view, index, sequence

Problem Statement: Design & develop SQL & DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence,

Synonym

Objective:

1. To understand & implement the various DDL commands.
2. To understand database concepts like view, index, sequence & synonym

S/w & H/w Requirements:

MySQL, Windows 10 (64 bit), i5 processor, JDBC, etc.

Theory:

VIEW:

In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows & columns just like a table.

## Create View System:

### 1) Simple View

View created by involving a single table

### 2) Complex view:

View created by involving multiple tables.

### 3) Drop View

drop view view-name;

## • INDEX:

Indexes are special lookup tables that database search engine can use to speedup data retrieval

An index is a pointer to data in a table.

### 1) SIMPLE INDEX:

- Create index on one column.

### 2) Create INDEX on multiple selected columns: - COMPOUND INDEX

### 3) UNIQUE INDEX-

created on selected column of database table & doesn't allow duplicate values of that column.

### 4) SHOW INDEX:

Shows all indexes created on table.

### 5) DROP INDEX:

Drops the given index.

## SEQUENCE:

- A sequence is a user defined schema bound object that generates a sequence of numerical values.
- Sequences are frequently used in many databases because application require each row to have a unique ID.

### Syntax:

auto-increment.

```
create table table-name( col1 type not null  
auto increment, primary key  
(col1), col2 type);
```

## SYNONYM

- Supported by Oracle.
- A synonym is an alternative name for objects such as tables, views, sequences, stored procedures, & other database objects.

### Syntax:

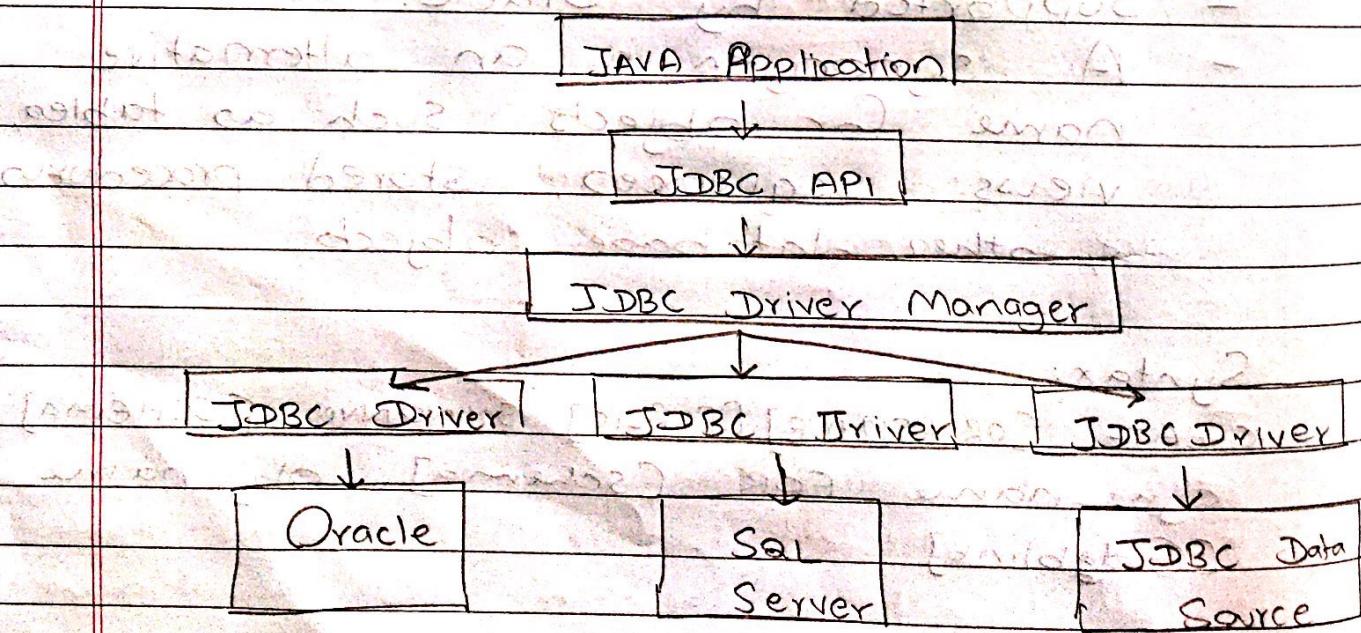
```
CREATE [OR REPLACE] [PUBLIC] SYNONYM [SCHEMA].  
syno-name FOR [schema] obj-name  
[@dgblink]
```

- JDBC:

JDBC stands for JAVA database connectivity which is a standard JAVA API for database independent connectivity.

Common JDBC components.

- 1) Driver Manager - This class manages a list of database drivers.
- 2) Driver - Handles communication with server
- 3) Connection - Contacting database
- 4) Statement - Use objects created from this interface to submit the SQL statements to the database.
- 5) ResultSet - Object data retrieved from database
- 6) SQL Exception - for error handling.



### Conclusion:

- 1) We understood how to create simple/complex views in a database.
- 2) We understood indexes, Sequences & synonyms & JDBC connection to MySQL

### Assignment 3

Date of Completion: 12 - 8 - 20

Date of Submission: 14 - 9 - 20

Title: Design at least 10 SQL queries for suitable database application using SQL.

#### Problem Statement:

Design at least 10 SQL queries for suitable database application using SQL DML statements. Insert, Select, Update, Delete with operators, functions & set operators.

#### Objective:

1. To understand & implement various DML commands.
2. To understand database concepts like functions & set operators.

S/w & H/w requirements: MySQL, i7 processor, 8 GB RAM.

#### Theory:

1. DML (Data Manipulation Language)  
It deals with data management & includes the most common SQL statements & is used to store, modify, retrieve, etc data in the database.

1) Select: To select entries

→ select col-name(s) from table-name

2) Insert: To insert entries

→ insert into table-name (...) values (...)

3) Update: To update entries

→ update table-name set col1=val1,  
col2=val2, ... where someCol=someVal.

4) Delete: To delete entries.

→ delete from table-name where someCol=someVal

- Set operators:

1) Union:

Returns union of 2 select statements.

→ select \* from table1 UNION select \* from table2

2) Union all:

Similar, but returns the duplicate  
values too.

3) Minus: Set(A) - Set(B)

→ select \* from table1 minus  
select \* from table2

4) Intersect: Set(A) ∩ Set(B)

→ select \* from table1 intersect  
select \* from table2

## • Integrity Constraints:

- Set rules used to maintain quality of info.
- It ensures that data integrity is maintained
- Used as guard against damage to database.

### Types:-

#### 1) Domain Constraint:

Definition of a valid set of values for an attribute. The value of attribute must be in corresponding domain.

#### 2) Entity Integrity Constraints:

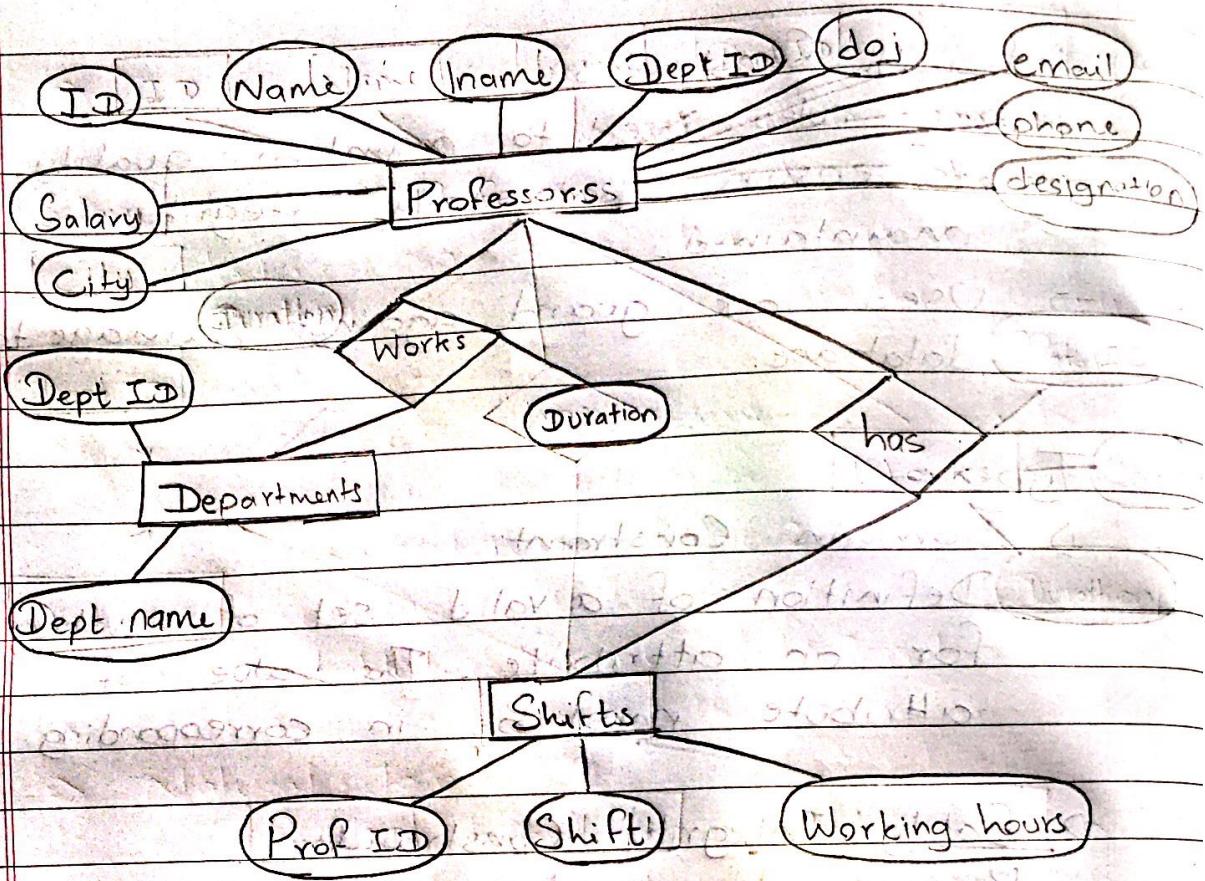
Primary key value can't be NULL & it should be unique.

#### 3) Referential Integrity Constraints:

If a foreign key in table refers to primary key in table then every value of foreign key must be null or be in table.

#### 4) Key constraints:-

keys are entity set that is used to identify an entity within its entity set uniquely.



- Conclusion:

Hence we learnt how to create table & handle basic data manipulation using SQL.

## Assignment 4

Date of Completion: 16-9-20

### • Title:

Design at least 10 SQL queries for all suitable database applications using SQL DML statements. All types of Join, Sub-query & view.

### • Objective:

1. Types of joins
2. Subquery & its types
3. Complex views.

### • Theory:

→ JOIN

SQL join is used to take data from two or more tables & then represent it as a single set of data. A table joining to itself is self join.

Types:-

i) Cross Join:

It returns Cartesian product from the rows from table in join.

Syntax:

```
SELECT <column-name-list> from table1  
CROSS JOIN table2;
```

### 2) Inner Join:-

It is a simple join in which the result is based on matched data as per the equality condition specified in the query.

Syntax :-

SELECT <col-name-list>

FROM table1

INNER JOIN table2

where

table1.col-name = table2.col-name

### 3) Natural Join:-

It is a type of inner join which is based on column having same name & datatype present in both table to be joined.

Syntax :-

SELECT \* from

table1, table2 from table1

NATURAL JOIN table2

from table2

### 4) Outer Join:-

#### i) Left Outer Join => old records

It returns a result table with matched data of two table then remaining rows of left table & null for right table column.

Syntax-

```
SELECT <col-name-list>
FROM table1
LEFT OUTER JOIN
table2
on table1.col-name = table2.col-name.
```

ii) Right Outer Join:-

It returns result table with matched data of two table then remaining rows of right table & null from left table column.

Syntax:-

```
SELECT <col-name-list>
FROM table1
RIGHT OUTER JOIN
table2
on table1.col-name = table2.col-name.
```

iii) Full outer Join:

It returns a result table with matched data & then remaining rows from both left & right tables.

Syntax:

```
SELECT <col-name-list>
from table1
FULL OUTER JOIN
table2
on table1.col-name = table2.col-name.
```

## \* Subquery:

- 1) A Subquery or inner query or a nested query is a query within another SQL query & embedded within a WHERE clause.
- 2) A Subquery is used to return data that will be used in main query as a condition to further restrict the data to be retrieved.

### Rules:

- 1) Must be enclosed with Parathesis
- 2) Sub-query must have only 1 column unless the condition is related to more than 1 column
- 3) "ORDER BY" cannot be used in sub-query.
- 4) A sub-query cannot be immediatly enclosed in a Set Function.

## \* Conclusion:-

We learned about various Joins & their uses along with different operations on views & about Subqueries.