

Teaching Guidelines for Database Technologies PG-DAC Feb 2025

Duration: 72 hours (36 theory hours + 36 lab hours)

Objective: To introduce students to RDBMS and NoSQL Databases and facilitate hands-on experience on SQL (using MySQL) and MongoDB.

Prerequisites: Working knowledge of Windows and Linux, familiarity with programming.

Evaluation: 100 Marks

Weightage: CCEE – 40%, Lab exam – 40%, internals – 20%

Text Book:

- Murach's MySQL by Joel Murach / Shroff Publisher

References:

- Database System Concepts by Abraham Silberschatz, Henry Korth and S. Sudarshan / McGraw Hill
 - Database Design and Relational Theory: Normal Forms and All That Jazz by C. J. Date (Author) / O'Reilly
 - Fundamentals of Database System by Shamkant B. Navathe, Ramez Elmasri / Pearson
 - MySQL: The Complete Reference by Vikram Vaswani / McGraw Hill
 - SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management by Andreas Meier and Michael Kaufmann / Springer
 - MongoDB: The Definitive Guide by Shannon Bradshaw, Eoin Brazil and Kristina Chodorow / O'Reilly
 - <http://bigdata.stratebi.com/?language=en>
-

(Note: Each Lecture and Lab Session is of 2 hours)

Session 1:

Lecture

Introduction to DBMS, Basic Database Terminology

Types of DBMS: Relational, Object Relational and NoSQL Databases

Introduction to MySQL, MySQL Clients (Monitor, Shell, Workbench)

Lab

Using MySQL Monitor, Shell, and Workbench

Session 2:

Lecture

Data Models (Conceptual, Logical, Physical)

Database Design, Entity-Relationship Diagram (ERD)

Codd's 12 rules for RDBMS

Introduction to SQL, Categories of SQL Commands: DDL, DML, DCL, DTL/TCL

DDL (CREATE/ALTER/DROP/TRUNCATE)

Lab

Performing basic CREATE, ALTER, DROP Commands

Sessions 3 & 4:

Lecture

Data Redundancy, Data Anomalies, Functional Dependency

Normalization, Need for Normalization

Normal Forms (1st NF, 2nd NF, 3rd NF, BCNF) with examples, Introduction to 4th and 5th NF

DML (INSERT/UPDATE/DELETE)

Lab (2 hours)

DML (INSERT/UPDATE/DELETE), TRUNCATE

Session 5:

Lecture

MySQL Data Types, Database Constraints (Primary Key, Unique, Not Null, Foreign Key, Default, Check*)

Aggregate Functions, Grouping Things Together (Group By, Having)

LIKE Operator, DISTINCT, Sorting (Order by clause)

BETWEEN... AND Operators, Comparing Nulls (IS NULL/IS Not NULL), IN/NOT IN

Lab

Defining Data Types for Columns

Creating, Altering, Dropping Constraints

Aggregate Functions: SUM(), AVG(), COUNT(), MAX(), MIN(), COUNT(), Group By, Having Clause

Using Like, Distinct, Order By, Between...And

Comparing Nulls, Using IN/Not-In

Session 6:

Lecture

Relational Algebra Operations (Selection, Projection, Union, Intersect*, Minus*, Cross/Cartesian)

Joins (Eqvi, Inner, Outer, Natural, Cross), SQL Standard Syntax for Joins

Copying table structure/data, Sequences (AUTO_INCREMENT)

Lab

Union/Union ALL

Queries on Various type of Joins using OLD and SQL Standard Syntax

Copying table structure, Copying data from one table to another

Using AUTO_INCREMENT

Session 7:

Lecture

Subquery, Correlated Subquery, EXISTS/NOT EXISTS

TCL Commands (Commit/Rollback/Savepoint), DCL Commands (GRANT/REVOKE/GRANT OPTION)

Views, Types of Views, Simple and Complex Views

Lab

Subqueries, Correlated Queries

Using Exists/Not-Exists

Using Commit/Rollback/Savepoint

Granting/revoking privileges on database objects

Creating Views, Querying using Views

Creating Indexes

Creating Temporary Tables

Session 8:**Lecture**

Indexes, Benefit of Indexes, Type of Indexes, Temporary Tables
ACID Properties, Concept of Database Instance and Schema
MySQL Storage Engines (InnoDB, MyISAM and others),

Lab

Indexes, Temporary Tables
All other SQL Commands Revision

Session 9:**Lecture**

Introduction to MySQL Programming, Use of MySQL Programs,
Introduction to Stored Procedures, Benefits of Stored Procedures
Procedure Parameters (IN, OUT and INOUT).

Lab

Creating procedure without parameters
Creating Procedure with (IN/OUT/INOUT) Parameters

Session 10:**Lecture**

Flow Control Statements (LOOP, WHILE and REPEAT)
Using above statements in Stored Procedures/Functions
Conditional Statements (IF, IF-ELSE-THEN, SWITCH CASE)
Example of each type of statement

Lab

Use of flow control statement in Stored Procedure
Use of conditional statements in Stored Procedure

Session 11:**Lecture**

Loop constructs (ITERATE, LEAVE)
Functions with and without parameters
MySQL Built-in functions (string, numeric, date etc.)

Lab (4 hours)

Creating Function and returning value from it
Use of built-in functions in queries

Session 12:**Lecture**

Cursors (Asensitive, Insensitive, Read only, Nonscrollable)
Cursors example and real time use

Lab:

Writing procedures with Declare, fetch and close cursor
Example of each type of cursors

Session 13:**Lecture**

Triggers (BEFORE, AFTER), New and Old trigger variables

Trigger Examples and real time use

Lab

Create Before Triggers

Create After Triggers

Session 14:**Lecture**

Error Handling and Exceptions, Types of Handler Actions, How to write Handler

Defining and handling exceptions in Stored Procedures and Functions

Lab

Exception handling in Stored Procedure

Exception handling with various handler actions

Session 15:**Lecture**

Introduction to NoSQL database, Features of NoSQL Database

Structured vs. Semi-structured and Unstructured Data

Difference between RDBMS and NoSQL databases

CAP Theorem, BASE Model

Categories of NoSQL Databases: Key-Value Store, Document Store, Column-Oriented, Graph

No Lab**Session 16, 17, 18:****Lecture**

Introduction to MongoDB, Features of MongoDB

MongoDB command interface and MongoDB compass

MongoDB Documents & Collections

RDBMS & MongoDB analogies: relations/tables => collections; tuples/records => documents

JSON and BSON documents

Performing CRUD (CREATE, READ, UPDATE, DELETE) Operations, UPSERT

MongoDB – Operators, Sorting, Indexing

Lab (8 hours)

Using MongoDB Shell and Compass

Creating database, Connecting to a database, Creating Collections

Performing CRUD operations

MongoDB: Complex Read Using Operators, Sorting Operations, Creating Indexes