Admissions & Support:
support@mockattempt.com
Phone / WhatsApp: +91 98765 43210
Course Code: MYSQ • **Updated:** 01 Nov 2026
https://mockattempt.com/courses/mysql-training

MYSQ • RELATIONAL DATABASES & SQL ENGINEERING **BEGINNER TO PRODUCTION MYSQL & INNODB ARCHITECT**

100% LIVE INDUSTRY EXPERT COHORT + INTERNSHIP LETTER

MYSQL Training

Master relational SQL & MySQL from 3NF schema design, constraints, JOINs, GROUP BY/HAVING & Recursive CTEs to Window Functions, Clustered B+Tree EXPLAIN tuning, Triggers, JSON & MVCC ACID Transactions — with runnable SQL scripts, 10×50 MCQ Mock Tests per topic, and Spoken Voice + Gemini AI Interviews at ₹1,999.

Duration: 15–17 Weeks • Live + Hands-On Projects (212 Hrs) • **Modules:** 16 • **Live Topics:** 53 • **Labs & Tasks:** 106 • **Mocks / AI Vivas:** 386 / 386 • **Projects:** 16 Module + 1 Live Client Prod

Fixed Non-Negotiable Tuition Fee: Base Fee **₹1,999** + 18% GST (₹360) = **Total Incl. GST: ₹2,359**

Live Batch Tracks: Weekday Live Online Batch (Mon – Fri) (Monday to Friday (5 Live Interactive Classes / Week) • 1.5 Hours / Day (60 Mins Live Architecture & Coding + 30 Mins Hands-On Lab) • 18–20 Weeks Fast-Track) | Weekend Intensive Live Online Batch (Sat – Sun) (Saturday & Sunday (Designed for Working Professionals & College Students) • 3.0 Hours / Day (2 Hrs Deep-Dive Live Coding + 1 Hr Guided Project Build) • 20–22 Weeks Weekend Track)

Core Technologies & Engineering Stack: MySQL 8.0+ • InnoDB & MVCC • Complex JOINs & CTEs • Window Analytics • B+Tree Indexing & EXPLAIN • Procedures & Triggers • ACID & Backups

1. PROGRAM MASTERY OUTCOMES & ELIGIBILITY

End-to-End Engineering Outcomes:

1. Design normalized 3NF relational schemas using utf8mb4, exact DECIMAL financial types, Foreign Keys, and CHECK constraints
2. Write complex analytical SQL using INNER/LEFT/SELF JOINs, GROUP BY WITH ROLLUP, Recursive CTEs, and Window Ranking Functions
3. Eliminate slow full-table scans (type: ALL) using Composite Covering B+Tree Indexes, SARGable predicates, and EXPLAIN ANALYZE
4. Clear 350 Timed Mock Exams (17,500 MySQL MCQs) and 350 Spoken Voice + Gemini AI SQL & Database Technical Interviews
5. Normalized 3NF transactional schema design balanced with strategic denormalization for high-read reporting.
6. InnoDB clustered primary keys, covering composite B-Tree indexes, and `EXPLAIN ANALYZE` execution plan mastery.
7. Advanced analytical SQL: CTEs, Window Functions (`ROW_NUMBER`, `LAG`, `SUM OVER`), JSON columns, and partitioning.
8. ACID isolation levels (`REPEATABLE READ` vs `READ COMMITTED`), row locking (`FOR UPDATE`), and deadlock prevention.

Eligibility & Prerequisites: No prior SQL or database experience required — starts from relational tables & installation from scratch • Any laptop/PC running MySQL 8 / XAMPP MariaDB / MySQL Workbench • No prior MYSQL industry experience required — Module 1 starts from environment setup, core mental models, and plain-English fundamentals. • Basic computer & internet familiarity on a Windows, macOS, or Linux laptop (8 GB+ RAM recommended for smooth live coding labs). • Ideal for B.Tech / BCA / MCA / B.Sc IT students, non-CS beginners, career switchers, and working developers upgrading to 2026 production standards. • Commitment to attend live Weekday or Weekend classes, complete daily hands-on lab checklists, and practice topic-wise Mock Tests & AI Interviews.

2. DETAILED MODULE & TOPIC-WISE SYLLABUS (16 MODULES • 53 LIVE TOPICS)

MODULE 1 OF 16 • 12 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 1: MySQL 8 Architecture, Storage Engines (`InnoDB`), Data Types & `utf8mb4` Databases

Understand MySQL's client/optimizer/storage-engine architecture, why InnoDB dominates OLTP over MyISAM, choose exact financial & temporal data types, and provision full-Unicode `utf8mb4` databases.

Objectives: ✓ Trace a SQL query through Connection Pool, Parser, Cost-Based Optimizer, and InnoDB Buffer Pool • ✓ Compare InnoDB row-level locking, crash-safe redo logs & Foreign Keys against legacy MyISAM • ✓ Select optimal column types (`BIGINT UNSIGNED`, `DECIMAL(12,2)`, `VARCHAR`, `TIMESTAMP` vs `DATETIME`) and `utf8mb4` collations

1.1 MySQL Architecture — Relational Model, Cost-Based Optimizer & InnoDB Buffer Pool Lifecycle [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply MySQL Architecture — Relational Model, Cost-Based Optimizer & InnoDB Buffer Pool Lifecycle in real MYSQL systems — specifically: Relational tuples...

1.2 MySQL Storage Engines — `InnoDB` (ACID, Row Locks, FK) vs `MyISAM`, `MEMORY` & `CSV` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply MySQL Storage Engines — `InnoDB` (ACID, Row Locks, FK) vs `MyISAM`, `MEMORY` & `CSV` in real MYSQL systems — specifically: Pluggable storage engin...

1.3 MySQL Data Types — `BIGINT`, Exact `DECIMAL` vs `FLOAT`, `VARCHAR` vs `CHAR`, `DATETIME` vs `TIMESTAMP` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply MySQL Data Types — `BIGINT`, Exact `DECIMAL` vs `FLOAT`, `VARCHAR` vs `CHAR`, `DATETIME` vs `TIMESTAMP` in real MYSQL systems — specifically: Choo...

1.4 Database DDL & Character Sets — `CREATE DATABASE IF NOT EXISTS`, `utf8mb4` vs `utf8mb3` & Collations [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Database DDL & Character Sets — `CREATE DATABASE IF NOT EXISTS`, `utf8mb4` vs `utf8mb3` & Collations in real MYSQL systems — specifically: Creatin...

Module 1 Hands-On Capstone Project (100 Marks): Multi-Currency EdTech Database Blueprint (`utf8mb4` + Precision `DECIMAL` + `InnoDB`) — Provision a production-grade `mockattempt_edtech` MySQL 8 database using `CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci` with `courses`, `coupons`, and `payments` tables using exact `DECI...

MODULE 2 OF 16 • 12 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 2: Table Schema Design, Online `ALTER TABLE` & Relational Integrity Constraints

Architect normalized 3NF tables with `CREATE TABLE`, evolve live production schemas safely with `ALTER TABLE ALGORITHM=INSTANT`, and enforce ironclad data quality via `UNIQUE`, `CHECK`, and `FOREIGN KEY` constraints.

Objectives: ✓ Normalize schemas to 3NF while choosing surrogate vs natural keys wisely • ✓ Distinguish `DELETE` (DML row-by-row undoable) vs `TRUNCATE` (fast DDL page drop + reset `AUTO_INCREMENT`) vs `DROP` • ✓ Enforce `NOT NULL`, multi-column `UNIQUE`, MySQL 8.0.16+ `CHECK`, and indexed `FOREIGN KEY` cascades

2.1 `CREATE TABLE`, `AUTO_INCREMENT`, `TEMPORARY` Tables & 1NF–3NF Relational Normalization [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply `CREATE TABLE`, `AUTO_INCREMENT`, `TEMPORARY` Tables & 1NF–3NF Relational Normalization in real MYSQL systems — specifically: Authoring self-docum...

2.2 Online DDL Evolution — `ALTER TABLE`, `ALGORITHM=INSTANT/INPLACE` & `TRUNCATE` vs `DELETE` vs `DROP` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Online DDL Evolution — `ALTER TABLE`, `ALGORITHM=INSTANT/INPLACE` & `TRUNCATE` vs `DELETE` vs `DROP` in real MYSQL systems — specifically: Adding/...

2.3 Relational Integrity Constraints — `NOT NULL`, `UNIQUE` (Multiple NULLs!), `CHECK` & `DEFAULT` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Relational Integrity Constraints — `NOT NULL`, `UNIQUE` (Multiple NULLs!), `CHECK` & `DEFAULT` in real MYSQL systems — specifically: Declarative b...

2.4 Primary Keys, Composite Keys & Foreign Keys — `ON DELETE CASCADE`, `RESTRICT` & `SET NULL` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Primary Keys, Composite Keys & Foreign Keys — `ON DELETE CASCADE`, `RESTRICT` & `SET NULL` in real MYSQL systems — specifically: Referential integ...

Module 2 Hands-On Capstone Project (100 Marks): 3NF University & Cohort Enrollment Schema with Enforced `CHECK` & `FOREIGN KEY` Cascades — Design a 5-table 3NF schema (`departments`, `courses`, `batches`, `students`, `batch_enrollments`) enforcing composite `UNIQUE (batch_id, student_id)`, `CHECK (discount_price <= price)`, `ON...

MODULE 3 OF 16 • 12 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 3: DML Mutations, Atomic Upserts, `SELECT` Filtering & Three-Valued `NULL` Logic

Master high-throughput multi-row `INSERT`s, atomic `ON DUPLICATE KEY UPDATE` upserts, safe `UPDATE`/`DELETE` guards, `WHERE` operator precedence, and SQL's three-valued `TRUE / FALSE / UNKNOWN` `NULL` logic.

Objectives: ✓ Batch thousands of rows in multi-row `INSERT` statements and perform race-free `ON DUPLICATE KEY UPDATE` upserts • ✓ Avoid catastrophic `UPDATE`/`DELETE` without `WHERE` using transactions and `--safe-updates` (`sql_safe_updates`) • ✓ Eliminate `= NULL` and `NOT IN (NULL)` bugs using `IS NULL`, `<=>`, and `COALESCE()`

3.1 Bulk `INSERT`, `INSERT ... SELECT` & Atomic Upserts (`ON DUPLICATE KEY UPDATE` vs `REPLACE`) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Bulk `INSERT`, `INSERT ... SELECT` & Atomic Upserts (`ON DUPLICATE KEY UPDATE` vs `REPLACE`) in real MYSQL systems — specifically: Multi-row `INSE...

3.2 Safe `UPDATE` & `DELETE` Mutations — `sql_safe_updates`, `ORDER BY ... LIMIT` Batching & Multi-Table Joins [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Safe `UPDATE` & `DELETE` Mutations — `sql_safe_updates`, `ORDER BY ... LIMIT` Batching & Multi-Table Joins in real MYSQL systems — specifically: T...

3.3 `SELECT` Filtering — `WHERE`, `AND` vs `OR` Precedence, `BETWEEN`, `IN`, `LIKE` Wildcards & `LIMIT` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply `SELECT` Filtering — `WHERE`, `AND` vs `OR` Precedence, `BETWEEN`, `IN`, `LIKE` Wildcards & `LIMIT` in real MYSQL systems — specifically: Boolean...

3.4 Three-Valued `NULL` Logic — `IS NULL`, `IS NOT NULL`, Null-Safe `<=>` Equal & `COALESCE()` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Three-Valued `NULL` Logic — `IS NULL`, `IS NOT NULL`, Null-Safe `<=>` Equal & `COALESCE()` in real MYSQL systems — specifically: Why `NULL` means...

Module 3 Hands-On Capstone Project (100 Marks): Idempotent Daily Attendance & Leaderboard Upsert Pipeline — Write a batch DML ingestion script that bulk-upserts daily student mock scores using MySQL 8 `INSERT INTO ... VALUES ... AS new_row ON DUPLICATE KEY UPDATE best_score = GREATEST(best_score,...

MODULE 4 OF 16 • 12 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 4: Aggregations, `GROUP BY`, `ONLY_FULL_GROUP_BY`, `HAVING` & Conditional `CASE` Pivots

Build executive KPI reports using `COUNT`, `SUM`, `AVG`, `GROUP_CONCAT`, master MySQL 8 `ONLY_FULL_GROUP_BY` compliance and `WITH ROLLUP` subtotals, separate `WHERE` from `HAVING`, and pivot rows into columns with `CASE WHEN`.

Objectives: ✓ Understand exact `NULL` handling across `COUNT(*)`, `COUNT(col)`, `COUNT(DISTINCT col)`, `AVG(col)`, and `GROUP_CONCAT` • ✓ Write strict `ONLY_FULL_GROUP_BY`-compliant aggregations and grand-total rows via `WITH ROLLUP` • ✓ Build single-pass conditional aggregations (`SUM(CASE WHEN status = "paid" THEN amount ELSE 0 END)`)

4.1 MySQL Aggregate Functions — `COUNT(*)` vs `COUNT(col)`, `COUNT(DISTINCT)`, `SUM`, `AVG` & `GROUP_CONCAT` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply MySQL Aggregate Functions — `COUNT(*)` vs `COUNT(col)`, `COUNT(DISTINCT)`, `SUM`, `AVG` & `GROUP_CONCAT` in real MYSQL systems — specifically: How...

4.2 `GROUP BY` Mechanics, `ONLY_FULL_GROUP_BY` Mode, Functional Dependency & `WITH ROLLUP` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply `GROUP BY` Mechanics, `ONLY_FULL_GROUP_BY` Mode, Functional Dependency & `WITH ROLLUP` in real MYSQL systems — specifically: Grouping rows into su...

4.3 `HAVING` vs `WHERE` — Pre-Grouping Row Filtering vs Post-Grouping Aggregate Filtering [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply `HAVING` vs `WHERE` — Pre-Grouping Row Filtering vs Post-Grouping Aggregate Filtering in real MYSQL systems — specifically: Why `WHERE` cannot ref...

4.4 Conditional SQL — `CASE WHEN ... THEN ... ELSE ... END`, `IF()`, `IFNULL()` & Conditional Pivot Aggregation [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Conditional SQL — `CASE WHEN ... THEN ... ELSE ... END`, `IF()`, `IFNULL()` & Conditional Pivot Aggregation in real MYSQL systems — specifically:...

Module 4 Hands-On Capstone Project (100 Marks): Single-Pass Cohort Revenue, Pass-Rate & Monthly Pivot Analytics Dashboard SQL — Write a set of single-scan analytical SQL queries computing per-course enrollment counts, paid vs pending revenue via `SUM(CASE WHEN ...)`, pass-rate percentages protected by `NULLIF`, `GROU...

MODULE 5 OF 16 • 14 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 5: Relational `JOIN` Mastery (`INNER`, `LEFT`, `SELF`, `CROSS`) & `UNION` vs `UNION ALL`

Combine normalized multi-table data accurately using `INNER JOIN`, `LEFT JOIN` anti-joins, hierarchical `SELF JOIN`s, calendar `CROSS JOIN`s, and high-speed `UNION ALL` set operations.

Objectives: ✓ Understand Nested-Loop & Hash Join execution in MySQL 8 and index foreign join keys • ✓ Avoid the classic `LEFT JOIN` + `WHERE right.col = X` trap that silently turns an outer join into an inner join • ✓ Choose streaming `UNION ALL` over sort-deduplicating `UNION` when inputs are already disjoint

5.1 `INNER JOIN` Mechanics — Explicit `ON` Predicates, Driving Table Choice & MySQL 8 Hash Joins [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply `INNER JOIN` Mechanics — Explicit `ON` Predicates, Driving Table Choice & MySQL 8 Hash Joins in real MYSQL systems — specifically: Matching rows a...

5.2 `LEFT JOIN` — Preserving Unmatched Rows, `ON` vs `WHERE` Placement Trap & Anti-Joins (`IS NULL`) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply `LEFT JOIN` — Preserving Unmatched Rows, `ON` vs `WHERE` Placement Trap & Anti-Joins (`IS NULL`) in real MYSQL systems — specifically: Preserving...

5.3 `SELF JOIN` Hierarchies, `RIGHT JOIN` Rewrites & Intentional `CROSS JOIN` Matrix Generation [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply `SELF JOIN` Hierarchies, `RIGHT JOIN` Rewrites & Intentional `CROSS JOIN` Matrix Generation in real MYSQL systems — specifically: Joining a table...

5.4 Set Operations — `UNION ALL` (Zero-Sort Streaming) vs `UNION` (Deduplication) & MySQL 8 `INTERSECT`/`EXCEPT` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Set Operations — `UNION ALL` (Zero-Sort Streaming) vs `UNION` (Deduplication) & MySQL 8 `INTERSECT`/`EXCEPT` in real MYSQL systems — specifically:...

Module 5 Hands-On Capstone Project (100 Marks): Multi-Table Student Progress, Unattempted Gap Finder (`LEFT JOIN` Anti-Join) & Org Hierarchy SQL — Write normalized multi-table reporting queries joining `users`, `groups`, `modules`, `topics`, and `mock_attempts` — including a `LEFT JOIN ... WHERE a.id IS NULL` anti-join finding students...

MODULE 6 OF 16 • 12 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 6: Subqueries, `EXISTS` vs `NOT IN`, Recursive CTEs (`WITH`) & Updatable Views

Write clean modular SQL with correlated `EXISTS` semi-joins, avoid the deadly `NOT IN (NULL)` trap, traverse arbitrary-depth trees with MySQL 8 `WITH RECURSIVE` CTEs, and encapsulate security views.

Objectives: ✓ Know why `NOT IN (subquery)` returns ZERO rows if a single `NULL` appears in the subquery (use `NOT EXISTS`!) • ✓ Refactor deeply nested derived tables into readable, reusable `WITH` Common Table Expressions and `WITH RECURSIVE` hierarchies • ✓ Understand `ALGORITHM=MERGE` vs `TEMPTABLE` views and `WITH CHECK OPTION` security

6.1 Subqueries in MySQL — Scalar, Derived Tables, `EXISTS` Semi-Joins & The Deadly `NOT IN (NULL)` Trap [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Subqueries in MySQL — Scalar, Derived Tables, `EXISTS` Semi-Joins & The Deadly `NOT IN (NULL)` Trap in real MYSQL systems — specifically: Uncorrel...

6.2 Common Table Expressions (`WITH` & `WITH RECURSIVE`) — Multi-Step Pipelines & Hierarchical Trees [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Common Table Expressions (`WITH` & `WITH RECURSIVE`) — Multi-Step Pipelines & Hierarchical Trees in real MYSQL systems — specifically: Introduced...

6.3 MySQL Views (`CREATE OR REPLACE VIEW`) — `ALGORITHM=MERGE` vs `TEMPTABLE`, Security & `WITH CHECK OPTION` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply MySQL Views (`CREATE OR REPLACE VIEW`) — `ALGORITHM=MERGE` vs `TEMPTABLE`, Security & `WITH CHECK OPTION` in real MYSQL systems — specifically: Vi...

Module 6 Hands-On Capstone Project (100 Marks): Recursive Syllabus Prerequisite Tree (`WITH RECURSIVE`) & Executive Security Views — Model an arbitrary-depth topic prerequisite graph traversed via `WITH RECURSIVE` (with cycle/depth guard `cte_max_recursion_depth`), identify uncompleted dependencies using `NOT EXISTS`, and...

MODULE 7 OF 16 • 12 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 7: Analytical Window Functions (`OVER`, `PARTITION BY`, Ranking & `LEAD`/`LAG`)

Perform advanced OLAP analytics directly in MySQL 8 without collapsing detail rows: running totals (`SUM() OVER`), Top-N per group (`ROW_NUMBER`, `DENSE_RANK`), and retention deltas (`LAG`/`LEAD`).

Objectives: ✓ Understand why Window Functions preserve individual rows unlike `GROUP BY` row-collapsing • ✓ Solve "Top 3 Scorers per Course" and "Nth Highest Salary" cleanly with `DENSE_RANK()` / `ROW_NUMBER()` • ✓ Compute month-over-month growth and session streaks using `LAG()`, `LEAD()`, and explicit window frames

7.1 Window Functions (`OVER`, `PARTITION BY`, `ORDER BY`) & Frame Clauses (`ROWS` vs `RANGE`) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Window Functions (`OVER`, `PARTITION BY`, `ORDER BY`) & Frame Clauses (`ROWS` vs `RANGE`) in real MYSQL systems — specifically: How MySQL 8 evalua...

7.2 MySQL Ranking Functions — `ROW_NUMBER()`, `RANK()`, `DENSE_RANK()`, `NTILE()` & Nth Highest Record [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply MySQL Ranking Functions — `ROW_NUMBER()`, `RANK()`, `DENSE_RANK()`, `NTILE()` & Nth Highest Record in real MYSQL systems — specifically: Exact tie...

7.3 Offset Window Functions — `LAG()`, `LEAD()`, `FIRST_VALUE()` & `LAST_VALUE()` Frame Gotcha [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Offset Window Functions — `LAG()`, `LEAD()`, `FIRST_VALUE()` & `LAST_VALUE()` Frame Gotcha in real MYSQL systems — specifically: Peeking backward...

Module 7 Hands-On Capstone Project (100 Marks): Cohort Leaderboard, Percentile Rank & Attempt-over-Attempt Improvement Engine — Write MySQL 8 Window Function analytics computing each student's tie-safe rank per batch (`DENSE_RANK()`), Top-3 podium winners per module via CTE + `ROW_NUMBER() <= 3`, cumulative XP runnin...

MODULE 8 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 8: B+Tree Indexing Internals, InnoDB Clustered Keys & `EXPLAIN` Performance Tuning

Slash slow 5-second queries down to 0.4 milliseconds by mastering InnoDB Clustered vs Secondary B+Trees, Composite Leftmost-Prefix design, SARGable predicates, Covering Indexes, and `EXPLAIN ANALYZE`.

Objectives: ✓ Design composite indexes `(a, b, c)` respecting the Leftmost-Prefix rule and equality-before-range ordering • ✓ Explain why random `CHAR(36)` UUID primary keys fragment InnoDB 16 KB leaf pages (prefer `BIGINT` or sequential UUIDv7 `BINARY(16)`) • ✓ Diagnose `type: ALL`, `Using temporary`, and `Using filesort` using `EXPLAIN` and `EXPLAIN ANALYZE`

8.1 Creating B+Tree Indexes — Composite Leftmost-Prefix Rule, SARGable Queries & Covering Indexes [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Creating B+Tree Indexes — Composite Leftmost-Prefix Rule, SARGable Queries & Covering Indexes in real MYSQL systems — specifically: How B+Tree lea...

8.2 InnoDB Clustered Index vs Secondary (Non-Clustered) Indexes — Bookmark Lookups & UUIDv4 Page Splits [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply InnoDB Clustered Index vs Secondary (Non-Clustered) Indexes — Bookmark Lookups & UUIDv4 Page Splits in real MYSQL systems — specifically: In InnoD...

8.3 Mastering `EXPLAIN` & `EXPLAIN ANALYZE` — Access Types (`const` > `eq_ref` > `ref` > `range` > `ALL`) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Mastering `EXPLAIN` & `EXPLAIN ANALYZE` — Access Types (`const` > `eq_ref` > `ref` > `range` > `ALL`) in real MYSQL systems — specifically: Decodi...

Module 8 Hands-On Capstone Project (100 Marks): Slow-Query Forensics & Sub-Millisecond Composite Covering Index Redesign — Take 5 deliberately unoptimized OLTP queries (non-SARGable `WHERE DATE(created_at) = ...`, wrong composite column order, `OFFSET 100000` deep pagination, `SELECT *` bookmark-lookup storm) an...

MODULE 9 OF 16 • 12 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 9: Stored Procedures, Triggers & Indexed Native Binary `JSON` Columns

Encapsulate database routines safely with Stored Procedures (`IN`/`OUT`, `SIGNAL SQLSTATE`, Prepared Statements), automate row-level audit trails with Triggers (`NEW`/`OLD`), and index semi-structured MySQL 8 `JSON` fields.

Objectives: ✓ Distinguish Stored Procedures (`CALL`, transactions allowed, multiple result sets) from Stored Functions (used in `SELECT`, deterministic/no transactions) • ✓ Write lightweight `BEFORE`/`AFTER` audit & invariant Triggers without hidden recursive latency • ✓ Query binary `JSON` columns (`->>`) at B+Tree speed using Virtual Generated Columns + Functional Indexes

9.1 Stored Procedures & Functions — `DELIMITER`, `IN`/`OUT` Params, `SIGNAL SQLSTATE` & Prepared SQL [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Stored Procedures & Functions — `DELIMITER`, `IN`/`OUT` Params, `SIGNAL SQLSTATE` & Prepared SQL in real MYSQL systems — specifically: Creating ro...

9.2 MySQL Triggers (`BEFORE` / `AFTER` `INSERT`, `UPDATE`, `DELETE`) — `NEW` vs `OLD` & Audit Logs [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply MySQL Triggers (`BEFORE` / `AFTER` `INSERT`, `UPDATE`, `DELETE`) — `NEW` vs `OLD` & Audit Logs in real MYSQL systems — specifically: Row-level `FO...

9.3 Native `JSON` in MySQL 8 — Binary DOM Storage, `->` vs `->>`, `JSON_TABLE` & Virtual Functional Indexes [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Native `JSON` in MySQL 8 — Binary DOM Storage, `->` vs `->>`, `JSON_TABLE` & Virtual Functional Indexes in real MYSQL systems — specifically: How...

Module 9 Hands-On Capstone Project (100 Marks): Immutable Financial Audit Trigger Trail & Indexed JSON Metadata Search Engine — Build an automated `AFTER UPDATE` fee-change audit trigger logging `OLD.discount_price` -> `NEW.discount_price` into `price_audit_logs`, a transactional `sp_issue_certificate` Stored Procedu...

MODULE 10 OF 16 • 16 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 10: ACID Transactions, InnoDB MVCC Isolation/Locking, DCL Security & Online Backups

Guarantee zero lost money or corrupted seats under high concurrency using ACID Transactions, InnoDB MVCC (`REPEATABLE READ`), `SELECT ... FOR UPDATE`, Least-Privilege `GRANT` roles, and zero-lock `mysqldump --single-transaction` backups.

Objectives: ✓ Master `START TRANSACTION`, `COMMIT`, `ROLLBACK`, `SAVEPOINT`, and why DDL causes an implicit commit • ✓ Compare `READ COMMITTED` vs MySQL's default `REPEATABLE READ` MVCC snapshots, Next-Key Locks, and deadlock avoidance • ✓ Harden database users with host-scoped

Least-Privilege `GRANT`s and perform hot backups + Binlog Point-in-Time Recovery

10.1 ACID Transactions, InnoDB MVCC (`REPEATABLE READ`), `SELECT ... FOR UPDATE` & Deadlock Prevention [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply ACID Transactions, InnoDB MVCC (`REPEATABLE READ`), `SELECT ... FOR UPDATE` & Deadlock Prevention in real MYSQL systems — specifically: Atomicity...

10.2 Database Security (`DCL`) — `CREATE USER`, `caching_sha2_password`, Least-Privilege `GRANT`/`REVOKE` & Roles [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Database Security (`DCL`) — `CREATE USER`, `caching_sha2_password`, Least-Privilege `GRANT`/`REVOKE` & Roles in real MYSQL systems — specifically:...

10.3 Production Backups, Point-in-Time Recovery (PITR) & Replication — `mysqldump --single-transaction` & Binlogs [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Production Backups, Point-in-Time Recovery (PITR) & Replication — `mysqldump --single-transaction` & Binlogs in real MYSQL systems — specifically:...

Module 10 Hands-On Capstone Project (100 Marks): Concurrency-Safe Wallet & Seat-Booking Transaction Engine + Hot Backup Runbook — Implement a deadlock-free seat booking & wallet debit transaction acquiring `SELECT seats_left FROM batches WHERE id = ? FOR UPDATE` in deterministic ascending ID order with `SAVEPOINT` part...

MODULE 11 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

InnoDB Storage Engine Internals, B+Tree Clustering & EXPLAIN Tuning

Production-grade MYSQL Training engineering module covering InnoDB Storage Engine Internals, B+Tree Clustering & EXPLAIN Tuning with hands-on labs and capstone build.

Objectives: ✓ InnoDB Clustered Primary Keys, Secondary Indexes & Covering Scans — Purpose: Design compact BIGINT/UUIDv7 keys and composite indexes that avoid random disk seeks. • ✓ Reading EXPLAIN ANALYZE (type, key, rows, Extra: Using filesort) — Purpose: Eliminate full table scans and temporary disk sorts on slow reporting endpoints. • ✓ Sargable Predicates, Prefix Indexes & Functional Indexes in MySQL 8 — Purpose: Rewrite non-sargable YEAR(created_at) filters so MySQL uses B+Tree indexes at full speed.

11.1 InnoDB Clustered Primary Keys, Secondary Indexes & Covering Scans [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply InnoDB Clustered Primary Keys, Secondary Indexes & Covering Scans in real MYSQL systems — specifically: Design compact BIGINT/UUIDv7 keys and comp...

11.2 Reading EXPLAIN ANALYZE (type, key, rows, Extra: Using filesort) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Reading EXPLAIN ANALYZE (type, key, rows, Extra: Using filesort) in real MYSQL systems — specifically: Eliminate full table scans and temporary di...

11.3 Sargable Predicates, Prefix Indexes & Functional Indexes in MySQL 8 [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Sargable Predicates, Prefix Indexes & Functional Indexes in MySQL 8 in real MYSQL systems — specifically: Rewrite non-sargable YEAR(created_at) fi...

Module 11 Hands-On Capstone Project (100 Marks): Sub-10ms Query Optimization Audit on a 5M-Row E-Commerce Schema — Design, implement, test, and document 'Sub-10ms Query Optimization Audit on a 5M-Row E-Commerce Schema' applying every production concept from Module 11 (InnoDB Storage Engine Internals, B+T...

MODULE 12 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

ACID Transactions, MVCC Isolation Levels, Gap Locks & Deadlocks

Production-grade MYSQL Training engineering module covering ACID Transactions, MVCC Isolation Levels, Gap Locks & Deadlocks with hands-on labs and capstone build.

Objectives: ✓ InnoDB MVCC Undo Logs & Repeatable Read vs Read Committed — Purpose: Understand phantom prevention, consistent snapshots, and locking read behavior. • ✓ Record Locks, Gap Locks, Next-Key Locks & SELECT ... FOR UPDATE — Purpose: Prevent overselling stock and diagnose InnoDB deadlock graphs (SHOW ENGINE INNODB STATUS). • ✓ Idempotent Upserts (INSERT ... ON DUPLICATE KEY UPDATE) & Batching — Purpose: Ingest high-velocity telemetry and webhook syncs atomically without race errors.

12.1 InnoDB MVCC Undo Logs & Repeatable Read vs Read Committed [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply InnoDB MVCC Undo Logs & Repeatable Read vs Read Committed in real MYSQL systems — specifically: Understand phantom prevention, consistent snapshot...

12.2 Record Locks, Gap Locks, Next-Key Locks & SELECT ... FOR UPDATE [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Record Locks, Gap Locks, Next-Key Locks & SELECT ... FOR UPDATE in real MYSQL systems — specifically: Prevent overselling stock and diagnose InnoD...

12.3 Idempotent Upserts (INSERT ... ON DUPLICATE KEY UPDATE) & Batching [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Idempotent Upserts (INSERT ... ON DUPLICATE KEY UPDATE) & Batching in real MYSQL systems — specifically: Ingest high-velocity telemetry and webhoo...

Module 12 Hands-On Capstone Project (100 Marks): Race-Free Wallet Ledger & Flash-Sale Inventory Transaction Engine — Design, implement, test, and document 'Race-Free Wallet Ledger & Flash-Sale Inventory Transaction Engine' applying every production concept from Module 12 (ACID Transactions, MVCC Isolation...

MODULE 13 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

MySQL 8 Window Functions, Recursive CTEs, JSON & Table Partitioning

Production-grade MYSQL Training engineering module covering MySQL 8 Window Functions, Recursive CTEs, JSON & Table Partitioning with hands-on labs and capstone build.

Objectives: ✓ RANK, DENSE_RANK, LAG, LEAD & Moving-Window Aggregations — Purpose: Compute running balances, top-N per category, and MoM growth in pure MySQL 8 SQL. • ✓ Recursive CTE Hierarchies & Generated Indexed JSON Columns — Purpose: Query org/category trees and index semi-structured JSON attributes cleanly. • ✓ RANGE / LIST Table Partitioning & Instant Partition Drop Archival — Purpose: Purge expired 90-day logs in milliseconds without locking active InnoDB tables.

13.1 RANK, DENSE_RANK, LAG, LEAD & Moving-Window Aggregations [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply RANK, DENSE_RANK, LAG, LEAD & Moving-Window Aggregations in real MYSQL systems — specifically: Compute running balances, top-N per category, and M...

13.2 Recursive CTE Hierarchies & Generated Indexed JSON Columns [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Recursive CTE Hierarchies & Generated Indexed JSON Columns in real MYSQL systems — specifically: Query org/category trees and index semi-structure...

13.3 RANGE / LIST Table Partitioning & Instant Partition Drop Archival [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply RANGE / LIST Table Partitioning & Instant Partition Drop Archival in real MYSQL systems — specifically: Purge expired 90-day logs in milliseconds...

Module 13 Hands-On Capstone Project (100 Marks): Executive Cohort Analytics & Range-Partitioned Audit Warehouse — Design, implement, test, and document 'Executive Cohort Analytics & Range-Partitioned Audit Warehouse' applying every production concept from Module 13 (MySQL 8 Window Functions, Recursive C...

MODULE 14 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Replication, Point-in-Time Recovery (PITR), Security & Zero-Downtime DDL

Production-grade MYSQL Training engineering module covering Replication, Point-in-Time Recovery (PITR), Security & Zero-Downtime DDL with hands-on labs and capstone build.

Objectives: ✓ GTID Binary Log Replication & Read/Write Workload Splitting — Purpose: Scale read-heavy traffic across replicas and promote standbys safely during maintenance. • ✓ Logical/Physical Backups + Binlog Point-in-Time Recovery (PITR) — Purpose: Restore accidentally deleted rows to the exact second before human error occurred. • ✓ Least-Privilege MySQL Roles, Slow Query Log & ALGORITHM=INSTANT DDL — Purpose: Alter multi-million-row production tables online without blocking live customer writes.

14.1 GTID Binary Log Replication & Read/Write Workload Splitting [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply GTID Binary Log Replication & Read/Write Workload Splitting in real MYSQL systems — specifically: Scale read-heavy traffic across replicas and pro...

14.2 Logical/Physical Backups + Binlog Point-in-Time Recovery (PITR) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Logical/Physical Backups + Binlog Point-in-Time Recovery (PITR) in real MYSQL systems — specifically: Restore accidentally deleted rows to the exa...

14.3 Least-Privilege MySQL Roles, Slow Query Log & ALGORITHM=INSTANT DDL [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Least-Privilege MySQL Roles, Slow Query Log & ALGORITHM=INSTANT DDL in real MYSQL systems — specifically: Alter multi-million-row production table...

Module 14 Hands-On Capstone Project (100 Marks): High-Availability Primary–Replica & PITR Disaster Recovery Setup — Design, implement, test, and document 'High-Availability Primary–Replica & PITR Disaster Recovery Setup' applying every production concept from Module 14 (Replication, Point-in-Time Recovery...

MODULE 15 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

High-Concurrency InnoDB Locking, Deadlock Diagnosis & Partitioning

Self-explanatory list-and-bullet production mastery module covering High-Concurrency InnoDB Locking, Deadlock Diagnosis & Partitioning in MYSQL Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ InnoDB Row Locks, Gap Locks, Next-Key Locks & Deadlock Log Diagnosis — Purpose: Understand why concurrent transactions block or deadlock and order SQL updates to prevent them. • ✓ Zero-Downtime Online DDL Schema Migrations & Safe Column Evolution — Purpose: Add indexes and columns to multi-million-row production MySQL tables without locking live traffic. • ✓ Range/List Table Partitioning, Archival Strategies & Slow Query Log Tuning — Purpose: Keep hot operational tables compact and sub-10ms fast as historical data grows. • ✓ Build, verify, and document 'Deadlock-Free Wallet Ledger & Range-Partitioned Audit Schema in MySQL 8' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + Google Gemini AI Technical Interviews.

15.1 InnoDB Row Locks, Gap Locks, Next-Key Locks & Deadlock Log Diagnosis [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply InnoDB Row Locks, Gap Locks, Next-Key Locks & Deadlock Log Diagnosis in real MYSQL systems — specifically: Understand why concurrent transactions...

15.2 Zero-Downtime Online DDL Schema Migrations & Safe Column Evolution [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Zero-Downtime Online DDL Schema Migrations & Safe Column Evolution in real MYSQL systems — specifically: Add indexes and columns to multi-million-...

15.3 Range/List Table Partitioning, Archival Strategies & Slow Query Log Tuning [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Range/List Table Partitioning, Archival Strategies & Slow Query Log Tuning in real MYSQL systems — specifically: Keep hot operational tables compa...

Module 15 Hands-On Capstone Project (100 Marks): Deadlock-Free Wallet Ledger & Range-Partitioned Audit Schema in MySQL 8 — Architect, implement, test, and document 'Deadlock-Free Wallet Ledger & Range-Partitioned Audit Schema in MySQL 8' applying every bullet-wise engineering concept from Module 15 (High-Concurr...

MODULE 16 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

MySQL Replication, Point-in-Time Recovery (PITR) & SQL Interview Mastery

Self-explanatory list-and-bullet production mastery module covering MySQL Replication, Point-in-Time Recovery (PITR) & SQL Interview Mastery in MYSQL Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ GTID Replication, Read/Write Splitting & Replication Lag Safeguards — Purpose: Scale read-heavy workloads across replicas while handling read-your-own-writes consistency. • ✓ Logical/Physical Backups, Binary Log Point-in-Time Recovery & Least-Privilege Grants — Purpose: Restore accidental table drops to the exact second before failure and enforce strict DB privileges. • ✓ Complex Analytical SQL (Window Functions, Recursive CTEs & Top-N Per Group) — Purpose: Write high-scoring SQL queries for data engineering and backend product-company interviews. • ✓ Build, verify, and document 'Read-Replica Routing Blueprint, Binary-Log PITR Drill & Analytical SQL Suite' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + Google Gemini AI Technical Interviews.

16.1 GTID Replication, Read/Write Splitting & Replication Lag Safeguards [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply GTID Replication, Read/Write Splitting & Replication Lag Safeguards in real MYSQL systems — specifically: Scale read-heavy workloads across replic...

16.2 Logical/Physical Backups, Binary Log Point-in-Time Recovery & Least-Privilege Grants [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Logical/Physical Backups, Binary Log Point-in-Time Recovery & Least-Privilege Grants in real MYSQL systems — specifically: Restore accidental tabl...

16.3 Complex Analytical SQL (Window Functions, Recursive CTEs & Top-N Per Group) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Complex Analytical SQL (Window Functions, Recursive CTEs & Top-N Per Group) in real MYSQL systems — specifically: Write high-scoring SQL queries f...

Module 16 Hands-On Capstone Project (100 Marks): Read-Replica Routing Blueprint, Binary-Log PITR Drill & Analytical SQL Suite — Architect, implement, test, and document 'Read-Replica Routing Blueprint, Binary-Log PITR Drill & Analytical SQL Suite' applying every bullet-wise engineering concept from Module 16 (MySQL R...

3. LIVE CLIENT PRODUCTION PROJECT, GLOBAL CERTIFICATIONS & CAREER SCOPE

End-of-Course Live Client Production Project + Official Internship Confirmation Letter

Project: OrderVault High-Concurrency E-Commerce Relational Schema & Query Tuning Suite • **Confirmed Role:** MySQL 8 Database Design & Performance Tuning Intern

Domain & Production Infra: Multi-Vendor Retail & Inventory Analytics Client (MySQL 8.0 InnoDB + Read Replicas + Composite Indexing + Automated Backup & Slow-Log Audit)

- Normalized 3NF transactional e-commerce schemas with foreign-key constraints, ACID InnoDB transactions, and deadlock guards.
- Eliminated full-table scans using EXPLAIN ANALYZE, covering composite B-Tree indexes, and window-function analytics.
- Authored auditable stored procedures, triggers, views, and role-based database privilege policies.
- Executed zero-data-loss schema migrations and cut p95 checkout query latency by 78% in live client production.

Aligned Global Vendor Certifications: **1Z0-909** (Oracle Certified Professional: MySQL 8.0 Database Developer) • **1Z0-908** (Oracle Certified Professional: MySQL 8.0 Database Administrator)

Target Job Roles: SQL & Database Developer, Junior MySQL DBA / Data Engineer, Backend Database Specialist, BI & SQL Analytics Associate

Realistic Annual CTC Scope: **Fresher / Entry-Level (0–1 Yr):** ₹4.2 LPA – ₹7.8 LPA **Mid-Level Database / Data Engineer (1–3 Yrs):** ₹8.5 LPA – ₹16.5 LPA **Senior Data Architect / Backend Lead (3–6+ Yrs):** ₹17.0 LPA – ₹32.0+ LPA

MockAttempt Academy Pvt. Ltd. • Email: support@mockattempt.com • Phone/WhatsApp: +91 98765 43210 • Web: https://mockattempt.com/courses/mysql-training

Try 2 Free Topic-Wise 50-MCQ Mock Tests & Spoken Gemini AI Interviews daily on every topic above before enrolling.