A Chetan Varma_EOD_24-06-2025

Takeaway From Live Session:

Three-Tier Architecture :

1. Client Tier - Presentation Layer

- User interface: Web or mobile app.
- Sends input requests, receives processed data.
- **Example**: Mobile app to book Tirupati Darshan tickets.

2. DB Engine - Application Layer

- API or backend service that interprets client requests.
- Controls logic like virtual queuing, FIFO management.
- Translates inputs into SQL queries or service calls.

3. Data Tier - Storage Layer

- Database management system-DBMS.
- Stores and retrieves persistent data.
- In demo: **AWS RDS hosting MySQL DB**.

Demo: End-to-End Flow

- **Using MySQL Workbench + AWS RDS to create a database and retrieve it from AWS RDS**.
- Flow:

  Client -Laptop → DB Engine -RDS MySQL → Data Storage -Cloud
- Real-time queries executed on cloud DB confirmed tiered architecture.

Key concepts in Data Engineering:

- Relation Database, NoSQL Database
- Data warehouse
- Bigdata
- ETL
- ELT
- Data Lake
- Data Mart
- Data Lakehouse
- Delta Lake
- Delta Live Table

One-Tier vs. Three-Tier Architecture

Feature	One-Tier Architecture	Three-Tier Architecture
Layers	All combined	Separated: UI, Logic, DB
Scalability	Limited	Highly scalable
Security	Weak -everything on one machine	Strong -isolated tiers
Example	Local MySQL + UI	Web UI + API + RDS MySQL
Pros	Simple, fast to test	Modular, production-grade
Cons	Not suitable for real-world apps	Initial setup complexity

CloudGuru/Udemy/YouTube/Support links Studied:

3 Tier architecture: https://www.geeksforgeeks.org/dbms/introduction-of-3-tier-architecture-in-dbms-set-2/

Different normalization forms: https://www.geeksforgeeks.org/dbms/normal-forms-in-dbms/

Data Engineering tools: https://www.secoda.co/blog/the-top-20-most-commonly-used-data-engineering-tools

Hands-on Practice:

1. Sample Denormalized Dataset

StudentID	StudentName	Age	Course1	Course1Instructor	Course2	Course2Instructor
101	Alice	20	Math	Dr Smith	History	Dr Jones
102	Bob	19	Math	Dr Smith	Science	Dr Lee
101	Alice	20	Art	Dr Brown	Math	Dr Smith

Issues Identified

- Multiple courses stored in one row
- Redundant data repeated across rows
- Violation of atomicity and normalization principles

2. First Normal Form - 1NF

Steps Applied

- Removed multivalued columns
- Unpivoted repeating groups into separate rows
- Ensured each column holds only one value per row

Transformed Table - Enrollment1NF

StudentID	StudentName	Age	Course	Instructor
101	Alice	20	Math	Dr Smith
101	Alice	20	History	Dr Jones
102	Bob	19	Math	Dr Smith
102	Bob	19	Science	Dr Lee
101	Alice	20	Art	Dr Brown

Now each row has atomic values and no repeating groups. This satisfies First Normal Form.

3. Second Normal Form - 2NF

Identified Key

- Composite Primary Key: StudentID and Course

Identified Partial Dependencies

- StudentName and Age depend only on StudentID
- Instructor depends only on Course

Normalization Actions

- Split data into three separate tables

Student Table

StudentID	StudentName	Age
101	Alice	20
102	Bob	19

Course Table

Course	Instructor
Math	Dr Smith
History	Dr Jones
Science	Dr Lee
Art	Dr Brown

Enrollment Table

StudentID	Course
101	Math
101	History
101	Art
102	Math
102	Science

All non-key attributes fully depend on the entire key. This satisfies Second Normal Form.

4. 3NF and BCNF

Third Normal Form - 3NF

- No transitive dependencies
- All non-key attributes must depend only on the primary key

Assume we now want to store **Instructor Department** or **Instructor Email**. This leads to:

Course	Instructor	InstructorEmail
Math	Dr Smith	smith@email.com
History	Dr Jones	jones@email.com

Now **InstructorEmail depends on Instructor, not Course**, so we have a **transitive dependency**.

This violates 3NF.

Fix: Split the Course table

Revised Course Table

Course	Instructor
Math	Dr Smith
History	Dr Jones

New Instructor Table

Instructor	InstructorEmail
Dr Smith	smith@email.com
Dr Jones	jones@email.com

Boyce-Codd Normal Form - BCNF

- A stricter form of 3NF
- Every determinant must be a candidate key

Student

StudentID	StudentName	Age
101	Alice	20
102	Bob	19

Course

Course	Instructor
Math	Dr Smith
History	Dr Jones
Science	Dr Lee
Art	Dr Brown

Instructor

Instructor	InstructorEmail
Dr Smith	smith@email.com
Dr Jones	jones@email.com
Dr Lee	lee@email.com
Dr Brown	brown@email.com

Enrollment

StudentID	Course
101	Math
101	History
101	Art
102	Math
102	Science

Final Schema:

Student Table

Primary Key: StudentID

Course Table

Primary Key: Course

Enrollment Table

Primary Key: Composite of StudentID and Course

Foreign Keys: StudentID references Student, Course references Course

Progress in Self-Learning - SQL, Python, Agile, Jira, & Confluence:

Revised and practiced sql concepts on different types of key such as primary key, foreign key, composite key etc.