AeroAspire - SDE Intern
Gokul Krishna S

Week 4 – Day 1 (October 13)

Questions/Reflections :

1. What is the normalized form? Why avoid data redundancy?

   - A normalized form is a rule-based level of organization applied to database tables to reduce repetition and ensure data integrity. Think of normalization as cleaning up your tables step-by-step — each normal form (1NF, 2NF, 3NF, BCNF, etc.) makes your schema more consistent and less redundant.

     For example:
   - 1NF (First Normal Form): No repeating groups or multiple values in a single column — every field holds an atomic (single) value.
   - 2NF (Second Normal Form): Every non-key attribute fully depends on the whole primary key (not just part of it).
   - 3NF (Third Normal Form): Remove transitive dependencies — non-key attributes should depend only on the key.

     Avoid data redundancy (storing the same information in multiple places) because it can lead to:
   - Inconsistency: One record gets updated, but duplicates don't.
   - Wasted storage: Space consumed unnecessarily by duplicate data.
   - Difficult maintenance: Updating one record means checking and fixing multiple locations.
   - Normalization ensures a database is lean, consistent, and easy to maintain.

2. How would you decide types (VARCHAR, DATE, INT etc.)?

- INT → For whole numbers like user IDs, age, counts, or status codes.
- FLOAT or DECIMAL → For numbers with decimals such as prices or measurements.
- VARCHAR(length) → For variable-length text, like names, emails, and titles. Use only as large as you need (e.g., VARCHAR(50) for names).
- CHAR(length) → For fixed-length text (like country codes, e.g., CHAR(2)).
- DATE / DATETIME → For time-based data such as creation_date, due_date, or timestamps of updates.
- BOOLEAN or TINYINT(1) → For true/false flags, such as completed = true.
- TEXT → For long-form user input, such as comments or descriptions.

3. Describe schema changes you might make if tasks need tagging / user assignment.

- Tasks Table → Already holds core details (task_id, title, description, created_at, due_date, etc.).
- Users Table → Holds user details (user_id, name, email).
- Tags Table → Stores unique tags (tag_id, tag_name).
- Task_Tags Table (many-to-many link) → Connects tasks and tags:
- Columns: (task_id, tag_id). Each task can have multiple tags.
- Task_Assignments Table → Connects tasks to assigned users:
- Columns: (task_id, user_id, assigned_at).