

- ① Agile Methodology Overview:**
- Focuses on:
 - 1. Iterative development
 - 2. Collaboration with customers
 - 3. Responding to change quickly
 - 4. Delivering working software frequently
 - Agile teams are:
 - 1. Self-organizing
 - 2. Cross-functional
 - 3. Responsible for continuous delivery.

- ② Scrum Framework in Web Development:**
- Scrum includes:
 - ① Sprints
 - ② Daily Standups
 - ③ Scrum Roles
 - ④ Backlogs
 - Product Owner, Stakeholder, team, Product + Sprint

- ① Structured Sprint Planning:** During the Sprint Planning
- The Team:
1. Reviews user stories involving DB changes
 2. Identifies required tables, fields, constraints
 3. Ensures DB remains normalized
 4. Avoids unnecessary nulls or duplication

- ② Defining Database Requirements:** Before Development:
1. Check if the story requires more tables
 2. Identify needed foreign keys
 3. Ensure integer IDs are used
 4. Decide if EAV is required for flexible attributes.

- ③ Prioritizing Database Tasks:**
1. Creating new tables
 2. Adding constraints
 3. Updating relations
 4. Avoiding duplicate data
 5. Applying multi-role patterns
 6. Supporting multi-lingual content.

- ④ Estimating Database Tasks:**
- Consider:
 - 1. Complexity → joins, indexing, many-to-many relationships.
 - 2. EAV model use
 - 3. Multi-lingual or multi-role systems
 - 4. Query performance
 - 5. Normalization effort.
- All these affect Story point estimation

- ① Avoiding Null Values:**
- Problems with Nulls:
1. Confusing logic
 2. Extra conditions in queries
 3. Harder to validate
 4. Can break application logic
- Solutions:
1. Use Not Null Constraints
 2. Use empty arrays / collections instead
 3. Provide default values.

- ② Working with Integers:**
- Use integers for:
1. Primary Keys
 2. Foreign Keys
 3. Stale fields
- Benefits:
1. Faster processing
 2. Less storage
 3. Better indexing
 4. More efficient queries

- ③ Preventing Duplicate Values:**
- Techniques:
1. Unique Constraints → prevent inserting same value twice
 2. Split By ID Strategy → use IDs instead of repeating strings
 3. Validation before insertion → verify data before saving.
- * Avoid String IDs → Slower + inconsistent

① Scrum In Web Development: Scrum helps teams deliver small, functional parts of the system every sprint.

① Optimizing Data by ID:

Normalization Levels:

1. Avoid repeating data
2. Move repeated data to separate tables
3. Link with integer foreign keys.

Benefits:

1. Better performance
2. Less storage
3. Cleaner design
4. Easier to maintain.

Use lookup tables and foreign keys.

② Preventing String Repetition:

Never Repeat String Like:

1. Department names
2. Colours
3. Categories
4. Status names

Benefits:

1. Better performance
2. Less storage
3. Cleaner design
4. Easier to maintain.

Use lookup tables and foreign keys.

③ Problems with Comma-Separated Fields:

Cause:

1. Bad query performance
2. Hard to filter

3. Breaking Normalization.

4. Data inconsistency

Correct approach:

Use junction tables for many-to-many relationships.

⑤ Entity-Attribute-Value (EAV) Model:

EAV Definition:

- Entity = Objects
Attribute = Properties
Values = Actual attribute values

Used when attributes change frequently

EAV Structure:

- Tables:
1. Entities Table → Main items
 2. Attributes Table → list of possible attributes
 3. Values Table → Stores values for each attribute.

When to Use EAV:

1. Medical Records
2. Metadata
3. Dynamic forms
4. Product Catalogs
5. Systems where each entity has different properties

Drawbacks of EAV:

1. Complex queries
2. Requires pivoting
3. Harder to validate
4. May impact performance
5. Needs indexing and caching.

⑥ Multi-Role and Multi-Lingual Patterns:

Uses:

1. Roles Table
2. Junction Table (many-to-many)
3. Never store roles as comma-separated strings

Benefits:

1. Flexible
2. Clean permission structure
3. Avoid duplication

Uses:

1. Base Table (language neutral data)
2. Language Table
3. Translation Table

Benefits:

1. No repetition
2. Easy to add new languages
3. Consistent content management

EAV can also support changing permissions.

② Data Normalization Principles:

Optimizing Data by ID:

Normalized Levels:

1. Avoid repeating data
2. Move repeated data to separate tables
3. Link with integer foreign keys.

Benefits:

1. Better performance
2. Less storage
3. Cleaner design
4. Easier to maintain.

Use lookup tables and foreign keys.

Preventing String Repetition:

Never Repeat String Like:

1. Department names
2. Colours
3. Categories
4. Status names

Benefits:

1. Better performance
2. Less storage
3. Cleaner design
4. Easier to maintain.

Use lookup tables and foreign keys.

Problems with Comma-Separated Fields:

Cause:

1. Bad query performance
2. Hard to filter

3. Breaking Normalization.

4. Data inconsistency

Correct approach:

Use junction tables for many-to-many relationships.

Performance Optimization:

Key Techniques:

1. Use integers
2. Avoid duplicates
3. Avoid CSV fields
4. Avoid NULLs
5. Use Caching for heavy queries
6. Index foreign keys

Entity-Attribute-Value (EAV) Model:

Definition:

- Entity = Objects
Attribute = Properties
Values = Actual attribute values

When to Use EAV:

1. Medical Records
2. Metadata
3. Dynamic forms
4. Product Catalogs
5. Systems where each entity has different properties

Drawbacks of EAV:

1. Complex queries
2. Requires pivoting
3. Harder to validate
4. May impact performance
5. Needs indexing and caching.

Multi-Role and Multi-Lingual Patterns:

Uses:

1. Roles Table
2. Junction Table (many-to-many)
3. Never store roles as comma-separated strings

Benefits:

1. Flexible
2. Clean permission structure
3. Avoid duplication

Uses:

1. Base Table (language neutral data)
2. Language Table
3. Translation Table

Benefits:

1. No repetition
2. Easy to add new languages
3. Consistent content management

Advanced Database Patterns:

Hashing:

1. Used for:
 - 1. Data Integrity
 - 2. Secure Passwords
 - 3. File Validation

Secure File Storage:

Best Practices:

1. Never store files in DB directly
 - Store paths
2. Use normalized metadata tables
3. Hash files for integrity

Summary:

1. Scrum DB work must follow best practices every sprint.
2. Avoid NULLs, Duplicates, CSV fields.
3. Always use integer keys.
4. Normalize using ID-based separation.
5. Use EAV for dynamic attributes.
6. Multi-Role = Junction Tables
7. Multi-Language = Separate translation tables.
8. Use hashing for security.
9. Optimize for performance with indexing + normalization.