EE3350

2025 JUNE

Q1

i) Define the term CRUD in database management. (1 marks)
Key Points:

- Definition: CRUD stands for Create, Read, Update, Delete - the four basic operations performed on database records.
- Create: Adding new records to the database.
- Read: Retrieving existing data from the database.
- Update: Modifying existing records in the database.
- Delete: Removing records from the database.

Instructions for Checking:

- 0.5 mark for correct definition of CRUD.
- 0.5 mark for explaining the four basic operations.

ii) Explain the role of primary keys in relational databases. (1 marks)
Key Points:

- Definition: A primary key is a unique identifier for each record in a database table.
- Uniqueness: Ensures no duplicate records exist in the table.
- Non-null: Primary key values cannot be empty or null.
- Indexing: Automatically creates an index for faster data retrieval.
- Referential integrity: Serves as a reference point for foreign keys in other tables.

Instructions for Checking:

- 0.5 mark for defining primary key as unique identifier.
- 0.5 mark for explaining its role in maintaining data integrity and uniqueness.

iii) Define foreign key and its purpose in database tables. (1.5 marks)
Key Points:

- Definition: A foreign key is a field or combination of fields that references the primary key of another table.
- Purpose: Establishes relationships between tables and ensures **referential integrity**, meaning only valid data is stored.

Instructions for Checking:

- 0.5 mark for defining foreign key.
- 1 mark for explaining its purpose in establishing table relationships and mentioning referential integrity.
- Partial marks (0.5 marks) can be awarded for incomplete but correct explanations.

iv) List five common types of database relationships. (2.5 marks)
Key Points (Any five of the following):

1. One-to-One
2. One-to-Many
3. Many-to-One
4. Many-to-Many
5. Self-referencing / Recursive relationship

Instructions for Checking:

- 0.5 mark for each correctly listed relationship.

v) Discuss the importance and best practices of database schema design in large-scale projects like device management systems. (4 marks)
Key Points:

- **Importance:**
    - Scalability: Proper schema design ensures system can handle growing data volumes.

    - Performance: Well-designed schemas optimize query execution and reduce response times.
    - Data integrity: Maintains consistency and accuracy across large datasets.
    - Maintainability: Makes system easier to modify and extend over time.

- **Best Practices:**
    - Normalization: Reduce data redundancy and improve data integrity.
    - Indexing strategy: Create appropriate indexes for frequently queried fields.
    - Naming conventions: Use consistent, meaningful names for tables and columns.
    - Documentation: Maintain clear documentation of schema structure and relationships.
    - Security considerations: Implement proper access controls and data protection measures.

Instructions for Checking:

- 2 marks for explaining importance. (at least two clearly explained).
- 2 marks for explaining best practices (at least two clearly explained).
- Partial marks can be awarded: 0.5, 1, 1.5 marks for incomplete but correct explanations in each category.

Q2

i) What is Sequelize in the context of Node.js? (1 marks)

Key Points:

- Definition: Sequelize is a promise-based Object-Relational Mapping (ORM) library for Node.js.
- Database abstraction: Provides JavaScript objects to interact with database tables without writing raw SQL queries.
- Features: Supports migrations, validations, associations, and transactions.
- Benefits: Simplifies database operations and provides database-agnostic code.

Instructions for Checking:

- 1 mark for defining Sequelize as an ORM for Node.js or explaining its purpose in database abstraction or mentioning features and benefits.
- Partial marks (0.5 marks) can be awarded for incomplete but correct explanations.

ii) Explain JWT and its use in authentication. (1.5 marks)

Key Points:

- Definition: JWT (JSON Web Token) is a compact, URL-safe token format for securely transmitting information between parties.
- Stateless authentication: Used to verify user identity after login without storing session data server-side.
- Token structure: Contains encoded user information and expiration time.
- Security benefit: Eliminates need for server-side session storage, improving scalability.

Instructions for Checking:

- 0.5 mark for correct definition of JWT.
- 1 mark for explaining its use in authentication (stateless authentication or token-based authentication).
- Partial marks (0.5 marks) can be awarded for incomplete but correct explanations.

iii) What is the purpose of middleware in an Express.js application? (1 marks)

Key Points:

- Definition: Middleware functions are functions that execute during the request-response cycle in Express.js.
- Access: Have access to request object (req), response object (res), and next middleware function.
- Purpose: Perform operations like authentication, logging, parsing, error handling, and data validation.

Instructions for Checking:

- 0.5 mark for defining middleware as functions in request-response cycle.
- 0.5 mark for explaining their purpose (authentication, logging, parsing, etc.).

iv) List the HTTP methods used in RESTful APIs and their typical database operation equivalents. (2.5 marks)

Key Points:

- GET: Retrieve data from server (equivalent to SELECT/READ operations).
- POST: Create new resources (equivalent to INSERT/CREATE operations).
- PUT: Update/replace entire resource (equivalent to UPDATE operations).
- PATCH: Partially update resource (equivalent to UPDATE operations for specific fields).
- DELETE: Remove resources (equivalent to DELETE operations).

Instructions for Checking:

- 0.5 marks for each correctly paired HTTP method with database operation.
- Must include at least GET, POST, PUT, DELETE with correct database equivalents for full marks.
- Partial marks can be awarded for correct method names without database equivalents.

v) Explain how authorization and authentication are implemented in web applications, with reference to JWT and role-based access control. (4 marks)

Key Points:

- **Authentication:**
    - User identity verification (who you are).
    - User provides credentials (username/password), server validates and issues JWT token.
    - JWT token stored client-side and included in subsequent request headers.
    - Server validates JWT signature and expiration for protected routes.
- **Authorization:**
    - Permission verification (what you can access/do).
    - Role-based access control assigns users specific roles with defined permissions.
    - Server checks user roles from JWT payload against required permissions.
    - Different endpoints require different role levels (admin, user, guest).

Instructions for Checking:

- 2 marks for explaining JWT implementation in authentication process.
- 2 marks for explaining role-based access control in authorization.
- Partial marks can be awarded: 0.5, 1, 1.5 marks for incomplete but correct explanations in each category.