Building Web-based Applications with MERNStack

Milestone: Implementation in MySQL

Group 13 Harini Prasad Vasisht Sushmitha Sudharsan

+1(984)374-4836 (Harini Prasad Vasisht) +1(857)565-8800 (Sushmitha Sudharsan)

> <u>vasisht.h@northeastern.edu</u> <u>sudharsan.s@northeastern.edu</u>

Percentage of Effort Contributed by Student1:50%	
Percentage of Effort Contributed by Student2:50%	
Signature of Student 1:	havini
Signature of Student 2:	Jushnitha
Submission Date: 10	0/11/2024

CREATE TABLES FOR ENTITITES

```
1)User Table
CREATE TABLE User (
 user_id INT PRIMARY KEY AUTO_INCREMENT,
 username VARCHAR(50) NOT NULL,
 email_id VARCHAR(100) UNIQUE NOT NULL,
 password VARCHAR(50) NOT NULL
);
2)Admin Table
CREATE TABLE Admin (
 admin_id INT PRIMARY KEY AUTO_INCREMENT,
 user_id INT NOT NULL,
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE
);
3)Regular User Table
CREATE TABLE Regular (
 regular_id INT PRIMARY KEY AUTO_INCREMENT,
 user_id INT NOT NULL,
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE
);
4)Project Table
CREATE TABLE Project (
 project_id INT PRIMARY KEY AUTO_INCREMENT,
 project_name VARCHAR(100) NOT NULL,
 start_date DATE,
 end_date DATE,
```

```
description TEXT
);
5) Task Table
CREATE TABLE Task (
 task_id INT PRIMARY KEY AUTO_INCREMENT,
 title VARCHAR(100) NOT NULL,
 status ENUM('Pending', 'In Progress', 'Completed') NOT NULL,
 due_date DATE,
 user_id INT,
 project_id INT,
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE SET NULL,
 FOREIGN KEY (project_id) REFERENCES Project(project_id) ON DELETE SET NULL
);
6) Team Table
CREATE TABLE Team (
 team_id INT PRIMARY KEY AUTO_INCREMENT,
 team_name VARCHAR(100),
 project_id INT,
 FOREIGN KEY (project_id) REFERENCES Project(project_id) ON DELETE SET NULL
);
7)Note Table
CREATE TABLE Note (
 note_id INT PRIMARY KEY AUTO_INCREMENT,
 content TEXT NOT NULL,
 creation_date DATE NOT NULL,
 last_modified DATE,
 task_id INT NOT NULL,
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE CASCADE
```

```
);
8) Comment Table
CREATE TABLE Comment (
  comment_id INT PRIMARY KEY AUTO_INCREMENT,
  content TEXT NOT NULL,
  creation_date DATE NOT NULL,
 note_id INT NOT NULL,
 user_id INT NOT NULL,
 task_id INT,
 FOREIGN KEY (note_id) REFERENCES Note(note_id) ON DELETE CASCADE,
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE SET NULL
);
9) Discussion Table
CREATE TABLE Discussion (
 note_id INT NOT NULL,
 comment_id INT NOT NULL,
 task_id INT NOT NULL,
 PRIMARY KEY (note_id, comment_id, task_id),
 FOREIGN KEY (note_id) REFERENCES Note(note_id),
 FOREIGN KEY (comment_id) REFERENCES Comment(comment_id),
 FOREIGN KEY (task_id) REFERENCES Task(task_id)
);
10) Category Table
CREATE TABLE Category (
  category_id INT PRIMARY KEY AUTO_INCREMENT,
  category_name VARCHAR(50) NOT NULL,
  project_id INT,
```

```
FOREIGN KEY (project_id) REFERENCES Project(project_id) ON DELETE SET NULL
);
11) Log Table
CREATE TABLE Log (
 log_id INT PRIMARY KEY AUTO_INCREMENT,
  action VARCHAR(100) NOT NULL,
 timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
 user_id INT,
 task_id INT,
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE SET NULL,
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE SET NULL
);
12) Attachment Table
CREATE TABLE Attachment (
  attachment_id INT PRIMARY KEY AUTO_INCREMENT,
 type VARCHAR(50) NOT NULL,
 sent_time DATETIME DEFAULT CURRENT_TIMESTAMP,
 user_id INT,
 task_id INT,
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE SET NULL,
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE SET NULL
);
13) Event Table
CREATE TABLE Event (
  event_id INT PRIMARY KEY AUTO_INCREMENT,
  event_name VARCHAR(100) NOT NULL,
 date DATE,
  user_id INT,
```

```
task_id INT,
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE SET NULL,
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE SET NULL
);
14)Remainder Table
CREATE TABLE Remainder (
  remainder_id INT PRIMARY KEY AUTO_INCREMENT,
  remainder_time DATETIME NOT NULL,
 task_id INT NOT NULL,
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE CASCADE
);
15)Collaboration Table
CREATE TABLE Collaboration (
  collaboration_id INT PRIMARY KEY AUTO_INCREMENT,
 collaboration_req VARCHAR(100),
 user_id INT,
 task_id INT,
       FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE SET NULL,
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE SET NULL
);
16)Tag Table
CREATE TABLE Tag (
 tag_id INT PRIMARY KEY AUTO_INCREMENT,
 tag_name VARCHAR(50) NOT NULL
);
```

Relationship Tables

```
1) User-Task Table
CREATE TABLE User_Task (
 user_id INT NOT NULL,
 task_id INT NOT NULL,
 PRIMARY KEY (user_id, task_id),
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE CASCADE
);
2) User-Category Table
CREATE TABLE User_Category (
 user_id INT NOT NULL,
 category_id INT NOT NULL,
 PRIMARY KEY (user_id, category_id),
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
 FOREIGN KEY (category_id) REFERENCES Category(category_id) ON DELETE CASCADE
);
3) User-Collaborate Table
CREATE TABLE User_Collaborate (
 user_id INT NOT NULL,
 collaboration_id INT NOT NULL,
  PRIMARY KEY (user_id, collaboration_id),
 FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
 FOREIGN KEY (collaboration_id) REFERENCES Collaboration(collaboration_id) ON DELETE
CASCADE
);
```

```
4)Task-Tag Table
CREATE TABLE Task_Tag (
 task_id INT NOT NULL,
 tag_id INT NOT NULL,
 PRIMARY KEY (task_id, tag_id),
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE CASCADE,
 FOREIGN KEY (tag_id) REFERENCES Tag(tag_id) ON DELETE CASCADE
);
5)Task-Note Table
CREATE TABLE Task_Note (
 task_id INT NOT NULL,
 note_id INT NOT NULL,
 PRIMARY KEY (task_id, note_id),
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE CASCADE,
 FOREIGN KEY (note_id) REFERENCES Note(note_id) ON DELETE CASCADE
);
6)Task-Event Table
CREATE TABLE Task_Event (
 task_id INT NOT NULL,
 event_id INT NOT NULL,
 PRIMARY KEY (task_id, event_id),
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE CASCADE,
 FOREIGN KEY (event_id) REFERENCES Event(event_id) ON DELETE CASCADE
);
7)Task-Log Table
CREATE TABLE Task_Log(
 task_id INT NOT NULL,
 log_id INT NOT NULL,
```

```
PRIMARY KEY (task_id, log_id),
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE CASCADE,
 FOREIGN KEY (log_id) REFERENCES Log(log_id) ON DELETE CASCADE
);
8) Task-Attachment Table
CREATE TABLE Task_Attachment (
 task_id INT NOT NULL,
 attachment_id INT NOT NULL,
 PRIMARY KEY (task_id, attachment_id),
 FOREIGN KEY (task_id) REFERENCES Task(task_id) ON DELETE CASCADE,
 FOREIGN KEY (attachment_id) REFERENCES Attachment(attachment_id) ON DELETE
CASCADE
);
9)Team-Project Table
CREATE TABLE Team_Project (
 team_id INT NOT NULL,
 project_id INT NOT NULL,
 PRIMARY KEY (team_id, project_id),
 FOREIGN KEY (team_id) REFERENCES Team(team_id) ON DELETE CASCADE,
 FOREIGN KEY (project_id) REFERENCES Project(project_id) ON DELETE CASCADE
);
```

INSERTING ROWS INTO TABLES

Insert 20 unique records into User Table
 INSERT INTO User (username, email_id, password)
 VALUES

```
('john_doe1', 'john.doe1@example.com', 'password123'),
('jane_smith1', 'jane.smith1@example.com', 'securepass1'),
('alex_jones1', 'alex.jones1@example.com', 'alex123'),
('emily_davis1', 'emily.davis1@example.com', 'pass456'),
('michael_brown1', 'michael.brown1@example.com', 'mbrown789'),
('sam_harris', 'sam.harris@example.com', 'password789'),
('linda_white', 'linda.white@example.com', 'passabc'),
('chris_evans', 'chris.evans@example.com', 'evans123'),
('kim_lee', 'kim.lee@example.com', 'kim789'),
('nina_brown', 'nina.brown@example.com', 'nina456'),
('daniel_king', 'daniel.king@example.com', 'king123'),
('lisa_clark', 'lisa.clark@example.com', 'lisa999'),
('george_hall', 'george.hall@example.com', 'hall123'),
('jessica_turner', 'jessica.turner@example.com', 'turn456'),
('tom_young', 'tom.young@example.com', 'young123'),
('peter_white', 'peter.white@example.com', 'pwhite789'),
('susan_scott', 'susan.scott@example.com', 'sscott123'),
('rachel_green', 'rachel.green@example.com', 'greenpass'),
('david_patel', 'david.patel@example.com', 'dp123'),
('anna_martin', 'anna.martin@example.com', 'am123');
```

Insert 10 unique records into Admin Table
 INSERT INTO Admin (user_id)
 VALUES

```
(1), (2), (6), (8), (10), (12), (14), (16), (18), (20);
```

3) Insert 10 unique records into Regular Table
INSERT INTO Regular (user_id)
VALUES
(3), (4), (5), (7), (9), (11), (13), (15), (17), (19);

4) Insert 20 unique records into Project Table

INSERT INTO Project (project_name, start_date, end_date, description)
VALUES

```
('Project Alpha', '2024-01-01', '2024-12-31', 'A year-long development project.'),
('Project Beta', '2024-02-15', '2024-11-30', 'A rapid sprint project.'),
('Project Gamma', '2024-03-10', '2024-09-30', 'Customer-centric improvements.'),
('Project Delta', '2024-04-01', '2024-10-15', 'E-commerce backend optimization.'),
('Project Epsilon', '2024-05-05', '2024-11-05', 'User interface enhancements.'),
('Project Zeta', '2024-06-10', '2024-12-10', 'Cloud migration.'),
('Project Eta', '2024-07-15', '2024-12-15', 'Data analytics setup.'),
('Project Theta', '2024-08-20', '2024-12-20', 'System performance tuning.'),
('Project lota', '2024-09-01', '2024-12-31', 'Security upgrades.'),
('Project Kappa', '2024-10-01', '2024-12-15', 'Database optimization.'),
('Project Lambda', '2024-01-20', '2024-06-30', 'Machine learning integration.'),
('Project Mu', '2024-02-28', '2024-07-30', 'Automated testing deployment.'),
('Project Nu', '2024-03-15', '2024-08-15', 'Mobile platform support.'),
('Project Xi', '2024-04-12', '2024-09-12', 'API development.'),
('Project Omicron', '2024-05-18', '2024-10-18', 'Enterprise resource planning.'),
('Project Pi', '2024-06-25', '2024-11-25', 'Project management software.'),
('Project Rho', '2024-07-30', '2024-12-30', 'Customer data management.'),
('Project Sigma', '2024-08-15', '2024-11-15', 'Real-time data processing.'),
('Project Tau', '2024-09-10', '2024-12-10', 'Predictive analytics project.'),
('Project Upsilon', '2024-10-20', '2024-12-25', 'Inventory management.');
```

5) Insert 20 unique records into Task Table

```
INSERT INTO Task (title, status, due_date, user_id, project_id)
VALUES
  ('Module 1 Development', 'In Progress', '2024-06-15', 1, 1),
  ('Backend Setup', 'Pending', '2024-08-10', 3, 2),
  ('Frontend Testing', 'Completed', '2024-09-20', 4, 3),
  ('Deployment Planning', 'In Progress', '2024-12-01', 5, 4),
  ('User Testing', 'Pending', '2024-10-10', 6, 5),
  ('Data Migration', 'Completed', '2024-11-30', 8, 6),
  ('Database Setup', 'Pending', '2024-12-05', 10, 7),
  ('Performance Optimization', 'In Progress', '2024-07-20', 11, 8),
  ('Security Review', 'Completed', '2024-09-01', 12, 9),
  ('Bug Fixes', 'Pending', '2024-12-20', 14, 10),
  ('API Integration', 'In Progress', '2024-08-01', 15, 11),
  ('Automated Testing', 'Completed', '2024-11-01', 16, 12),
  ('Mobile Optimization', 'In Progress', '2024-07-25', 17, 13),
  ('Code Review', 'Pending', '2024-09-15', 18, 14),
  ('Documentation', 'Completed', '2024-06-10', 19, 15),
  ('User Training', 'In Progress', '2024-12-30', 20, 16),
  ('System Monitoring', 'Pending', '2024-12-15', 2, 17),
  ('Real-Time Data', 'Completed', '2024-07-15', 7, 18),
  ('Predictive Models', 'In Progress', '2024-11-20', 9, 19),
  ('Inventory Sync', 'Pending', '2024-11-25', 13, 20);
    6) Insert 20 unique records into Team Table
INSERT INTO Team (team_name, project_id)
VALUES
  ('Dev Team Alpha', 1), ('Dev Team Beta', 2), ('QA Team Gamma', 3),
  ('Ops Team Delta', 4), ('UX Team Epsilon', 5), ('Cloud Team Zeta', 6),
  ('Data Team Eta', 7), ('Perf Team Theta', 8), ('Security Team Iota', 9),
  ('DB Team Kappa', 10), ('Al Team Lambda', 11), ('Test Team Mu', 12),
  ('Mobile Team Nu', 13), ('API Team Xi', 14), ('ERP Team Omicron', 15),
```

```
('PM Team Pi', 16), ('Customer Team Rho', 17), ('Processing Team Sigma', 18), ('Analytics Team Tau', 19), ('Inventory Team Upsilon', 20);
```

7) Insert 20 unique records into Note TableINSERT INTO Note (content, creation_date, last_modified, task_id)VALUES

```
('Design layout for module 1', '2024-01-05', '2024-01-06', 1),
('API specifications', '2024-02-15', '2024-02-17', 2),
('Test plan for UI components', '2024-03-01', '2024-03-05', 3),
('User feedback summary', '2024-04-12', '2024-04-15', 4),
('Initial database schema', '2024-01-25', '2024-01-27', 5),
('Data migration checklist', '2024-05-20', '2024-05-21', 6),
('Database optimization notes', '2024-06-15', '2024-06-16', 7),
('Performance testing report', '2024-07-10', '2024-07-11', 8),
('Security assessment', '2024-08-12', '2024-08-14', 9),
('Bug triage list', '2024-09-01', '2024-09-03', 10),
('API error log', '2024-10-01', '2024-10-02', 11),
('Testing framework setup', '2024-11-10', '2024-11-12', 12),
('Mobile optimization checklist', '2024-08-25', '2024-08-26', 13),
('Code review summary', '2024-06-15', '2024-06-16', 14),
('Documentation outline', '2024-05-05', '2024-05-07', 15),
('User training materials', '2024-06-30', '2024-07-01', 16),
('System monitoring setup', '2024-07-25', '2024-07-26', 17),
('Real-time data processing log', '2024-09-15', '2024-09-16', 18),
('Predictive model results', '2024-10-20', '2024-10-22', 19),
('Inventory sync report', '2024-11-20', '2024-11-21', 20);
```

8) Insert 20 unique records into Comment Table
INSERT INTO Comment (content, creation_date, note_id, user_id, task_id)
VALUES

('Reviewed layout, looks good', '2024-01-06', 1, 2, 1),

```
('UI components passed all tests', '2024-03-05', 3, 4, 3),
  ('Feedback incorporated', '2024-04-15', 4, 5, 4),
  ('Schema reviewed, minor changes needed', '2024-01-27', 5, 6, 5),
  ('Data migration checklist approved', '2024-05-21', 6, 8, 6),
  ('Optimization steps noted', '2024-06-16', 7, 10, 7),
  ('Performance testing on schedule', '2024-07-11', 8, 11, 8),
  ('Security assessment approved', '2024-08-14', 9, 12, 9),
  ('Bug list updated', '2024-09-03', 10, 14, 10),
  ('API errors fixed', '2024-10-02', 11, 15, 11),
  ('Testing framework fully set up', '2024-11-12', 12, 16, 12),
  ('Mobile optimizations approved', '2024-08-26', 13, 17, 13),
  ('Code review complete', '2024-06-16', 14, 18, 14),
  ('Documentation outline complete', '2024-05-07', 15, 19, 15),
  ('Training materials prepared', '2024-07-01', 16, 20, 16),
  ('Monitoring tools deployed', '2024-07-26', 17, 2, 17),
  ('Real-time data processing verified', '2024-09-16', 18, 7, 18),
  ('Predictive models functioning well', '2024-10-22', 19, 9, 19),
  ('Inventory sync successful', '2024-11-21', 20, 13, 20);
    9) Insert 20 unique records into Category Table
INSERT INTO Category (category_name, project_id)
VALUES
  ('Design', 1), ('Backend', 2), ('Testing', 3), ('Deployment', 4),
  ('User Experience', 5), ('Database', 6), ('Data Migration', 7),
  ('Optimization', 8), ('Security', 9), ('Bug Fixes', 10),
  ('API', 11), ('Automated Testing', 12), ('Mobile', 13),
  ('Code Review', 14), ('Documentation', 15), ('Training', 16),
  ('Monitoring', 17), ('Real-Time Data', 18), ('Predictive Modeling', 19),
  ('Inventory', 20);
```

('Clarify API authentication steps', '2024-02-17', 2, 3, 2),

```
10) Insert 20 unique records into Log TableINSERT INTO Log (action, timestamp, user_id, task_id)VALUES
```

```
('Created Task', '2024-01-05 09:00:00', 1, 1),
('Updated Task', '2024-02-15 10:30:00', 2, 2),
('Completed Testing', '2024-03-01 11:45:00', 3, 3),
('Reviewed Feedback', '2024-04-12 12:15:00', 4, 4),
('Schema Approved', '2024-01-25 13:25:00', 5, 5),
('Migrated Data', '2024-05-20 14:30:00', 6, 6),
('Optimized Database', '2024-06-15 15:45:00', 7, 7),
('Performance Checked', '2024-07-10 16:50:00', 8, 8),
('Security Review Done', '2024-08-12 17:00:00', 9, 9),
('Bug Fixes Applied', '2024-09-01 18:10:00', 10, 10),
('API Errors Resolved', '2024-10-01 19:15:00', 11, 11),
('Testing Completed', '2024-11-10 20:30:00', 12, 12),
('Mobile Checked', '2024-08-25 21:45:00', 13, 13),
('Code Reviewed', '2024-06-15 22:55:00', 14, 14),
('Documentation Done', '2024-05-05 09:00:00', 15, 15),
('Training Scheduled', '2024-06-30 10:00:00', 16, 16),
('Monitoring Active', '2024-07-25 11:15:00', 17, 17),
('Data Processing Verified', '2024-09-15 12:25:00', 18, 18),
('Model Testing Done', '2024-10-20 13:35:00', 19, 19),
('Inventory Synced', '2024-11-20 14:45:00', 20, 20);
```

11) Insert 20 unique records into Attachment TableINSERT INTO Attachment (type, sent_time, user_id, task_id)VALUES

```
('PDF', '2024-02-15 12:00:00', 1, 1),

('Image', '2024-03-10 14:30:00', 2, 2),

('Doc', '2024-07-20 09:00:00', 3, 3),

('Spreadsheet', '2024-09-15 16:00:00', 4, 4),
```

```
('Presentation', '2024-11-20 10:15:00', 5, 5),
  ('Video', '2024-04-10 08:00:00', 6, 6),
  ('Text File', '2024-06-05 11:30:00', 7, 7),
  ('Audio', '2024-05-12 17:45:00', 8, 8),
  ('Zip Archive', '2024-09-20 12:00:00', 9, 9),
  ('Database Export', '2024-12-10 13:30:00', 10, 10),
  ('Image', '2024-08-05 14:15:00', 11, 11),
  ('Text Doc', '2024-10-20 15:00:00', 12, 12),
  ('PDF', '2024-07-10 09:00:00', 13, 13),
  ('Code File', '2024-06-15 10:00:00', 14, 14),
  ('Spreadsheet', '2024-12-05 11:45:00', 15, 15),
  ('Presentation', '2024-09-01 12:30:00', 16, 16),
  ('Audio Clip', '2024-11-25 13:45:00', 17, 17),
  ('Video', '2024-07-25 14:15:00', 18, 18),
  ('Project Docs', '2024-10-12 10:25:00', 19, 19),
  ('Project Summary', '2024-12-15 16:30:00', 20, 20);
    12) Insert 20 unique records into Event Table
INSERT INTO Event (event_name, date, user_id, task_id)
VALUES
  ('Kickoff Meeting', '2024-01-05', 1, 1),
  ('Planning Session', '2024-02-10', 2, 2),
  ('User Review', '2024-03-15', 3, 3),
  ('Project Closure', '2024-04-20', 4, 4),
  ('Client Feedback', '2024-05-25', 5, 5),
  ('Team Sync', '2024-06-30', 6, 6),
  ('Stakeholder Meeting', '2024-07-05', 7, 7),
  ('Code Review', '2024-08-10', 8, 8),
  ('Test Results Discussion', '2024-09-15', 9, 9),
  ('Bug Fixing Session', '2024-10-20', 10, 10),
  ('Feature Demo', '2024-11-25', 11, 11),
```

```
('Testing Summary', '2024-12-01', 12, 12),
  ('Deployment Discussion', '2024-06-05', 13, 13),
  ('System Handover', '2024-08-20', 14, 14),
  ('Documentation Review', '2024-09-10', 15, 15),
 ('QA Training', '2024-12-15', 16, 16),
  ('Database Review', '2024-01-15', 17, 17),
  ('Real-Time Testing', '2024-02-25', 18, 18),
  ('Model Validation', '2024-03-05', 19, 19),
  ('Inventory Meeting', '2024-04-15', 20, 20);
    13) Insert 20 unique records into Remainder Table
INSERT INTO Remainder (remainder_time, task_id)
VALUES
 ('2024-01-10 10:00:00', 1),
 ('2024-02-14 11:00:00', 2),
 ('2024-03-05 12:00:00', 3),
 ('2024-04-10 09:30:00', 4),
 ('2024-05-20 15:00:00', 5),
 ('2024-06-25 14:30:00', 6),
 ('2024-07-15 13:45:00', 7),
 ('2024-08-12 16:00:00', 8),
 ('2024-09-25 17:15:00', 9),
 ('2024-10-01 08:30:00', 10),
 ('2024-11-10 10:15:00', 11),
 ('2024-12-15 09:45:00', 12),
 ('2024-01-20 11:00:00', 13),
 ('2024-02-28 14:00:00', 14),
 ('2024-03-15 12:30:00', 15),
 ('2024-04-22 15:45:00', 16),
 ('2024-05-18 13:20:00', 17),
  ('2024-06-30 11:50:00', 18),
```

```
('2024-07-28 09:00:00', 19),
  ('2024-08-10 12:10:00', 20);
    14) Insert 20 unique records into Collaboration Table
INSERT INTO Collaboration (collaboration_req, user_id, task_id)
VALUES
  ('Backend collaboration', 1, 1),
  ('UI collaboration', 2, 2),
  ('Database assistance', 3, 3),
  ('Testing support', 4, 4),
  ('Optimization help', 5, 5),
  ('Security review', 6, 6),
  ('Deployment plan', 7, 7),
  ('Documentation update', 8, 8),
  ('User feedback session', 9, 9),
  ('Code review', 10, 10),
  ('API testing', 11, 11),
  ('Data migration', 12, 12),
  ('Performance tuning', 13, 13),
  ('Feature discussion', 14, 14),
  ('Bug fix review', 15, 15),
  ('Training session', 16, 16),
  ('Real-time sync', 17, 17),
  ('Predictive model analysis', 18, 18),
  ('Inventory management', 19, 19),
  ('Task prioritization', 20, 20);
    15) Insert 20 unique records into Tag Table
INSERT INTO Tag (tag_name)
VALUES
  ('Urgent'), ('High Priority'), ('Low Priority'), ('In Progress'), ('Completed'),
```

('Pending'), ('Needs Review'), ('Bug'), ('Feature'), ('Enhancement'), ('Testing'), ('Optimization'), ('Security'), ('Database'), ('User Feedback'), ('Performance'), ('Deployment'), ('Documentation'), ('Collaboration'), ('Finalized');

16) Insert 20 unique records into User_Task Table

INSERT INTO User_Task (user_id, task_id)

VALUES

17) Insert 20 unique records into User_Category Table

INSERT INTO User_Category (user_id, category_id)

VALUES

18) Insert 20 unique records into User_Collaborate Table

INSERT INTO User_Collaborate (user_id, collaboration_id)

VALUES

19) Insert 20 unique records into Task_Tag Table

INSERT INTO Task_Tag (task_id, tag_id)

VALUES

20) Insert 20 unique records into Task_Note Table

INSERT INTO Task_Note (task_id, note_id)

VALUES

21) Insert 20 unique records into Team_Project Table

INSERT INTO Team_Project (team_id, project_id)

VALUES

22) Insert unique records into User_Collaborate Table

INSERT INTO User_Collaborate (user_id, collaboration_id)

VALUES

23) Insert unique records into Task_Event Table

INSERT INTO Task_Event (task_id, event_id)

VALUES

$$(1, 1), (2, 1), (3, 2), (4, 2), (5, 3),$$

$$(6, 3), (7, 4), (8, 4), (9, 5), (10, 5),$$

$$(11, 6), (12, 6), (13, 7), (14, 7), (15, 8),$$

(16, 8), (17, 9), (18, 9), (19, 10), (20, 10);

24) Insert refined records into Task_Log Table

INSERT INTO Task_Log (task_id, log_id)

VALUES

25) Insert unique records into Task_Attachment Table

INSERT INTO Task_Attachment (task_id, attachment_id)

VALUES

$$(1, 1), (2, 1), (3, 2), (4, 2), (5, 3),$$

$$(6, 3), (7, 4), (8, 4), (9, 5), (10, 5),$$

$$(11, 6), (12, 6), (13, 7), (14, 7), (15, 8),$$

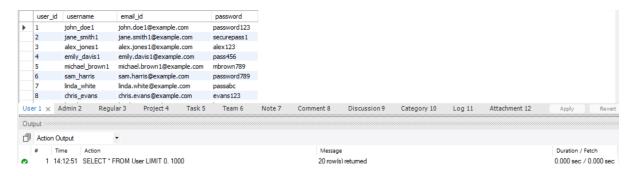
26) Inserting rows into the Discussion table

INSERT INTO Discussion (note_id, comment_id, task_id) VALUES

TABLE OUTPUTS:

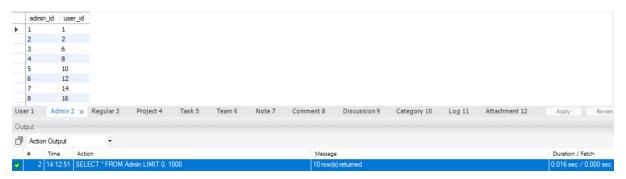
1) Select all records from the User table

SELECT * FROM User;



2) Select all records from the Admin table

SELECT * FROM Admin;



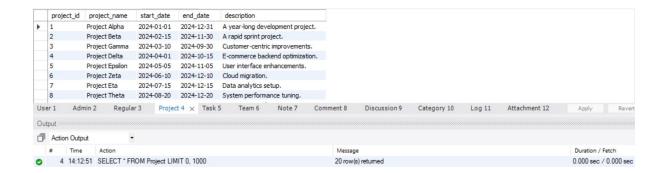
3) Select all records from the Regular table

SELECT * FROM Regular;



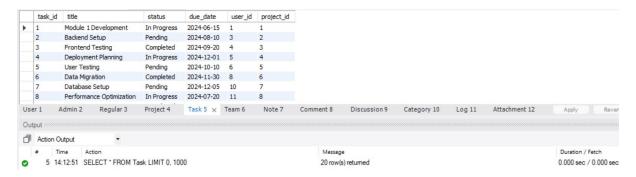
4) Select all records from the Project table

SELECT * FROM Project;



5) Select all records from the Task table

SELECT * FROM Task;



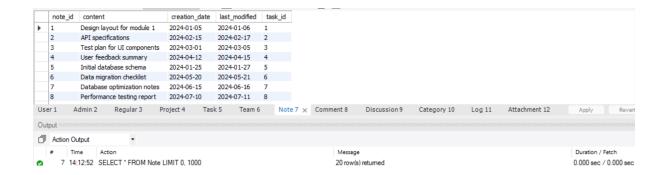
6) Select all records from the Team table

SELECT * FROM Team;



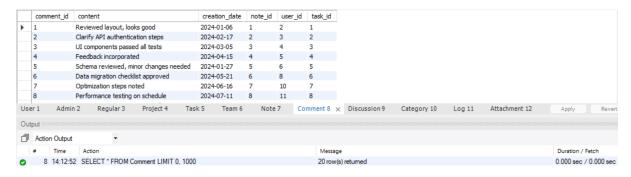
7) Select all records from the Note table

SELECT * FROM Note:



8) Select all records from the Comment table

SELECT * FROM Comment;



9) Select all records from the Discussion table

SELECT * FROM Discussion;



10) Select all records from the Category table

SELECT * FROM Category;



11) Select all records from the Log table

SELECT * FROM Log;



12) Select all records from the Attachment table

SELECT * FROM Attachment;



13) Select all records from the Event table

SELECT * FROM Event;



14) Select all records from the Remainder table

SELECT * FROM Remainder;



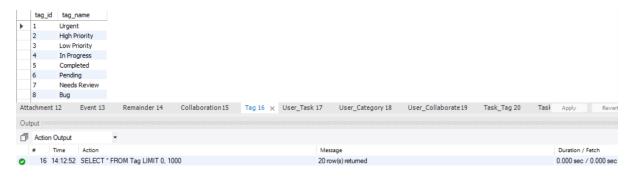
15) Select all records from the Collaboration table

SELECT * FROM Collaboration;



16) Select all records from the Tag table

SELECT * FROM Tag;



17) Select all records from the User_Task table

SELECT * FROM User_Task;



18) Select all records from the User_Category table

SELECT * FROM User_Category;



19) Select all records from the User_Collaborate table

SELECT * FROM User_Collaborate;



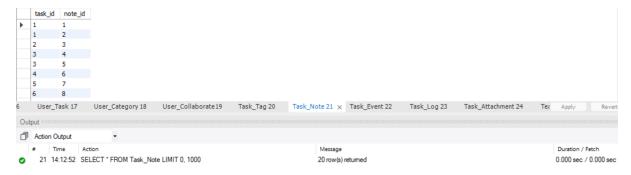
20) Select all records from the Task_Tag table

SELECT * FROM Task_Tag;



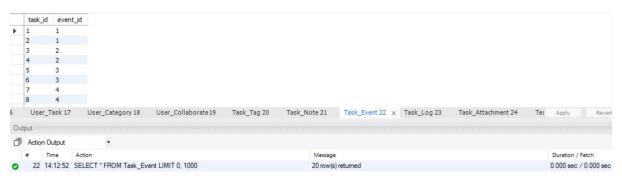
21) Select all records from the Task_Note table

SELECT * FROM Task_Note;



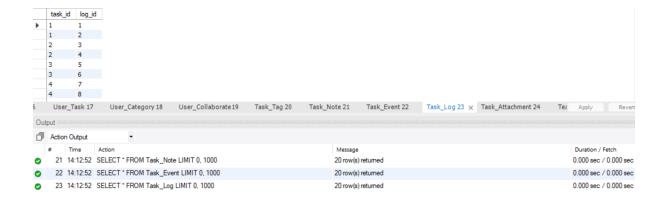
22) Select all records from the Task_Event table

SELECT * FROM Task_Event;



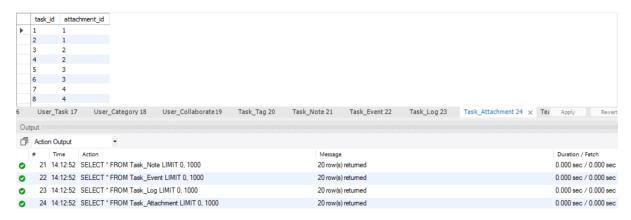
23) Select all records from the Task_Log table

SELECT * FROM Task_Log;



24) Select all records from the Task_Attachment table

SELECT * FROM Task_Attachment;



25) Select all records from the Team_Project table

SELECT * FROM Team_Project;

