



Lab Report

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	25
Problem Understanding & Analysis	7					
Implementation	8					
Report Writing	10					
Total obtained mark						
Comments						

Semester: Spring 2024

Student Name: Istewak Hassan Tewak

Student ID: 0242220005341228

Batch: 39

Section: C2

Course Code: SE224

Course Name: Database System Lab

Course Teacher Name: Mr. Kazi Rifat Ahmed

Designation: Lecturer

Submission Date: 11 /05/2024

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```
-- Create the coachingcenter database CREATE DATABASE IF NOT EXISTS coachingcenter;
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

⚠ Note: #1007 Can't create database 'coachingcenter'; database exists

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

```
USE coachingcenter;
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

⚠ Error: #1046 No database selected

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.3266 seconds.)

```
-- Create the Class table CREATE TABLE IF NOT EXISTS Class ( class_id INT AUTO_INCREMENT PRIMARY KEY, class_name VARCHAR(255), class_description TEXT );
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2158 seconds.)

```
-- Create the CoachingTutor table CREATE TABLE IF NOT EXISTS CoachingTutor ( tutor_id INT AUTO_INCREMENT PRIMARY KEY, tutor_name VARCHAR(255), subject_taught VARCHAR(255), contact_info VARCHAR(255) );
```

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2332 seconds.)

```
-- Create the Student table CREATE TABLE IF NOT EXISTS Student ( student_id INT AUTO_INCREMENT PRIMARY KEY, student_name VARCHAR(255), contact_info VARCHAR(255), address TEXT );
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2012 seconds.)

```
-- Create the Assignment table CREATE TABLE IF NOT EXISTS Assignment ( assignment_id INT AUTO_INCREMENT PRIMARY KEY, class_id INT, tutor_id INT, title VARCHAR(255), description TEXT, deadline DATE, FOREIGN KEY (class_id) REFERENCES Class(class_id), FOREIGN KEY (tutor_id) REFERENCES CoachingTutor(tutor_id) );
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2452 seconds.)

```
-- Create the Attendance table CREATE TABLE IF NOT EXISTS Attendance ( attendance_id INT AUTO_INCREMENT PRIMARY KEY, class_id INT, student_id INT, date DATE, status ENUM('Present', 'Absent'), FOREIGN KEY (class_id) REFERENCES Class(class_id), FOREIGN KEY (student_id) REFERENCES Student(student_id) );
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2468 seconds.)

```
-- Create the ClassBatch table CREATE TABLE IF NOT EXISTS ClassBatch ( batch_id INT AUTO_INCREMENT PRIMARY KEY, class_id INT, batch_name VARCHAR(255), start_date DATE, end_date DATE, FOREIGN KEY (class_id) REFERENCES Class(class_id) );
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2189 seconds.)

```
-- Create the ClassEnrollment table CREATE TABLE IF NOT EXISTS ClassEnrollment ( enrollment_id INT AUTO_INCREMENT PRIMARY KEY, batch_id INT, student_id INT, enrollment_date DATE, FOREIGN KEY (batch_id) REFERENCES ClassBatch(batch_id), FOREIGN KEY (student_id) REFERENCES Student(student_id) );
```

[[Edit inline](#)][[Edit](#)][[Create PHP code](#)]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1444 seconds.)

```
-- Create the UES (User-Entity-State) table CREATE TABLE IF NOT EXISTS UES ( user_id INT AUTO_INCREMENT PRIMARY KEY, username VARCHAR(255), password VARCHAR(255), role ENUM('Admin', 'Student', 'Tutor') );
```

[[Edit inline](#)][[Edit](#)][[Create PHP code](#)]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.2890 seconds.)

```
-- Create the Grade table CREATE TABLE IF NOT EXISTS Grade ( grade_id INT AUTO_INCREMENT PRIMARY KEY, assignment_id INT, student_id INT, grade_value DECIMAL(5,2), FOREIGN KEY (assignment_id) REFERENCES Assignment(assignment_id), FOREIGN KEY (student_id) REFERENCES Student(student_id) );
```

[[Edit inline](#)][[Edit](#)][[Create PHP code](#)]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.3158 seconds.)

```
-- Create the Transaction table CREATE TABLE IF NOT EXISTS Transaction ( transaction_id INT AUTO_INCREMENT PRIMARY KEY, student_id INT, amount DECIMAL(10,2), transaction_date DATE, transaction_type ENUM('Payment', 'Fee'), FOREIGN KEY (student_id) REFERENCES Student(student_id) );
```

[[Edit inline](#)][[Edit](#)][[Create PHP code](#)]

✓ 50 rows inserted. (Query took 0.0724 seconds.)

```
INSERT INTO class (class_id, class_name, class_description) VALUES (1, 'Class 1', 'Description for Class 1'), (2, 'Class 2', 'Description for Class 2'), (3, 'Class 3', 'Description for Class 3'), (4, 'Class 4', 'Description for Class 4'), (5, 'Class 5', 'Description for Class 5'), (6, 'Class 6', 'Description for Class 6'), (7, 'Class 7', 'Description for Class 7'), (8, 'Class 8', 'Description for Class 8'), (9, 'Class 9', 'Description for Class 9'), (10, 'Class 10', 'Description for Class 10'), (11, 'Class 11', 'Description for Class 11'), (12, 'Class 12', 'Description for Class 12'), (13, 'Class 13', 'Description for Class 13'), (14, 'Class 14', 'Description for Class 14'), (15, 'Class 15', 'Description for Class 15'), (16, 'Class 16', 'Description for Class 16'), (17, 'Class 17', 'Description for Class 17'), (18, 'Class 18', 'Description for Class 18'), (19, 'Class 19', 'Description for Class 19'), (20, 'Class 20', 'Description for Class 20');
```

[[Edit](#)]

✓ 50 rows inserted. (Query took 0.0005 seconds.)

```
-- Assuming the transaction table already exists -- Inserting data for student_id from 1 to 50 INSERT INTO transaction (transaction_id, student_id, amount, transaction_date, transaction_type) VALUES (1, 1, 100.50, '2024-05-11', 'Purchase'), (2, 2, 75.25, '2024-05-10', 'Refund'), (3, 3, 150.00, '2024-05-09', 'Purchase'), (4, 4, 200.75, '2024-05-08', 'Purchase'), (5, 5, 50.50, '2024-05-07', 'Refund'), (6, 6, 80.20, '2024-05-06', 'Purchase'), (7, 7, 95.75, '2024-05-05', 'Purchase'), (8, 8, 120.30, '2024-05-04', 'Refund'), (9, 9, 180.90, '2024-05-03', 'Purchase'), (10, 10, 70.40, '2024-05-02', 'Purchase'), (11, 11, 110.60, '2024-05-01', 'Refund'), (12, 12, 130.20, '2024-04-30', 'Purchase'), (13, 13, 90.10, '2024-04-29', 'Purchase'), (14, 14, 85.50, '2024-04-28', 'Refund'), (15, 15, 40.75, '2024-04-27', 'Purchase'), (16, 16, 55.80, '2024-04-26', 'Purchase'), (17, 17, 70.90, '2024-04-25', 'Purchase'), (18, 18, 60.30, '2024-04-24', 'Refund'), (19, 19, 100.10, '2024-04-23', 'Purchase'), (20, 20, 120.40, '2024-04-22', 'Refund'), (21, 21, 90.80, '2024-04-21', 'Purchase'), (22, 22, 110.50, '2024-04-20', 'Refund'), (23, 23, 130.70, '2024-04-19', 'Purchase'), (24, 24, 150.20, '2024-04-18', 'Refund'), (25, 25, 170.60, '2024-04-17', 'Purchase'), (26, 26, 190.90, '2024-04-16', 'Refund'), (27, 27, 210.30, '2024-04-15', 'Purchase'), (28, 28, 230.75, '2024-04-14', 'Refund'), (29, 29, 250.10, '2024-04-13', 'Purchase'), (30, 30, 270.50, '2024-04-12', 'Refund'), (31, 31, 290.80, '2024-04-11', 'Purchase'), (32, 32, 310.20, '2024-04-10', 'Refund'), (33, 33, 330.60, '2024-04-09', 'Purchase'), (34, 34, 350.90, '2024-04-08', 'Refund'), (35, 35, 370.30, '2024-04-07', 'Purchase'), (36, 36, 390.75, '2024-04-06', 'Refund'), (37, 37, 410.10, '2024-04-05', 'Purchase'), (38, 38, 430.50, '2024-04-04', 'Refund'), (39, 39, 450.80, '2024-04-03', 'Purchase'), (40, 40, 470.20, '2024-04-02', 'Refund'), (41, 41, 490.60, '2024-04-01', 'Purchase'), (42, 42, 510.90, '2024-03-31', 'Refund'), (43, 43, 530.30, '2024-03-30', 'Purchase'), (44, 44, 550.75, '2024-03-29', 'Refund'), (45, 45, 570.10, '2024-03-28', 'Purchase'), (46, 46, 590.50, '2024-03-27', 'Refund'), (47, 47, 610.80, '2024-03-26', 'Purchase'), (48, 48, 630.20, '2024-03-25', 'Refund'), (49, 49, 650.60, '2024-03-24', 'Purchase'), (50, 50, 670.90, '2024-03-23', 'Refund');
```

[[Edit](#)]

✓ 50 rows inserted. (Query took 0.0573 seconds.)

```
INSERT INTO ues (user_id, username, password, role) VALUES (1, 'user1', 'password1', 'role1'), (2, 'user2', 'password2', 'role2'), (3, 'user3', 'password3', 'role3'), (4, 'user4', 'password4', 'role4'), (5, 'user5', 'password5', 'role5'), (6, 'user6', 'password6', 'role6'), (7, 'user7', 'password7', 'role7'), (8, 'user8', 'password8', 'role8'), (9, 'user9', 'password9', 'role9'), (10, 'user10', 'password10', 'role10'), (11, 'user11', 'password11', 'role11'), (12, 'user12', 'password12', 'role12'), (13, 'user13', 'password13', 'role13'), (14, 'user14', 'password14', 'role14'), (15, 'user15', 'password15', 'role15'), (16, 'user16', 'password16', 'role16'), (17, 'user17', 'password17', 'role17'), (18, 'user18', 'password18', 'role18'), (19, 'user19', 'password19', 'role19'), (20, 'user20', 'password20', 'role20'), (21, 'user21', 'password21', 'role21'), (22, 'user22', 'password22', 'role22'), (23, 'user23', 'password23', 'role23'), (24, 'user24', 'password24', 'role24'), (25, 'user25', 'password25', 'role25'), (26, 'user26', 'password26', 'role26'), (27, 'user27', 'password27', 'role27'), (28, 'user28', 'password28', 'role28'), (29, 'user29', 'password29', 'role29'), (30, 'user30', 'password30', 'role30'), (31, 'user31', 'password31', 'role31'), (32, 'user32', 'password32', 'role32'), (33, 'user33', 'password33', 'role33'), (34, 'user34', 'password34', 'role34'), (35, 'user35', 'password35', 'role35'), (36, 'user36', 'password36', 'role36'), (37, 'user37', 'password37', 'role37'), (38, 'user38', 'password38', 'role38'), (39, 'user39', 'password39', 'role39'), (40, 'user40', 'password40', 'role40'), (41, 'user41', 'password41', 'role41'), (42, 'user42', 'password42', 'role42'), (43, 'user43', 'password43', 'role43'), (44, 'user44', 'password44', 'role44'), (45, 'user45', 'password45', 'role45'), (46, 'user46', 'password46', 'role46'), (47, 'user47', 'password47', 'role47'), (48, 'user48', 'password48', 'role48'), (49, 'user49', 'password49', 'role49'), (50, 'user50', 'password50', 'role50');
```

[[Edit](#)]

✓ 50 rows inserted. (Query took 0.0939 seconds.)

```
INSERT INTO coachingtutor (tutor_id, tutor_name, subject_taught, contact_info) VALUES (1, 'Tutor 1', 'Mathematics', '123-456-7890'), (2, 'Tutor 2', 'Physics', '234-567-8901'), (3, 'Tutor 3', 'Chemistry', '345-678-9012'), (4, 'Tutor 4', 'Biology', '456-789-0123'), (5, 'Tutor 5', 'English', '567-890-1234'), (6, 'Tutor 6', 'History', '678-901-2345'), (7, 'Tutor 7', 'Geography', '789-012-3456'), (8, 'Tutor 8', 'Computer Science', '890-123-4567'), (9, 'Tutor 9', 'Economics', '901-234-5678'), (10, 'Tutor 10', 'Psychology', '012-345-6789'), (11, 'Tutor 11', 'Sociology', '111-222-3333'), (12, 'Tutor 12', 'Political Science', '222-333-4444'), (13, 'Tutor 13', 'Anthropology', '333-444-5555'), (14, 'Tutor 14', 'Philosophy', '444-555-6666'), (15, 'Tutor 15', 'Art', '555-666-7777'), (16, 'Tutor 16', 'Music', '666-777-8888'), (17, 'Tutor 17', 'Drama', '777-888-9999'), (18, 'Tutor 18', 'Physical Education', '888-999-0000'), (19, 'Tutor 19', 'Languages', '999-000-1111'), (20, 'Tutor 20', 'Mathematics', '100-111-2222'), (21, 'Tutor 21', 'Physics', '211-222-3333'), (22, 'Tutor 22', 'Chemistry', '322-333-4444'), (23, 'Tutor 23', 'Biology', '433-444-5555'), (24, 'Tutor 24', 'English', '544-555-6666'), (25, 'Tutor 25', 'History', '655-666-7777'), (26, 'Tutor 26', 'Geography', '766-777-8888'), (27, 'Tutor 27', 'Computer Science', '877-888-9999'), (28, 'Tutor 28', 'Economics', '988-999-0000'), (29, 'Tutor 29', 'Psychology', '099-000-1111'), (30, 'Tutor 30', 'Sociology', '100-111-2222'), (31, 'Tutor 31', 'Political Science', '211-222-3333'), (32, 'Tutor 32', 'Anthropology', '322-333-4444'), (33, 'Tutor 33', 'Philosophy', '433-444-5555'), (34, 'Tutor 34', 'Art', '544-555-6666'), (35, 'Tutor 35', 'Music', '655-666-7777'), (36, 'Tutor 36', 'Drama', '766-777-8888'), (37, 'Tutor 37', 'Physical Education', '877-888-9999'), (38, 'Tutor 38', 'Languages', '988-999-0000'), (39, 'Tutor 39', 'Mathematics', '099-000-1111'), (40, 'Tutor 40', 'Physics', '100-111-2222'), (41, 'Tutor 41', 'Chemistry', '211-222-3333'), (42, 'Tutor 42', 'Biology', '322-333-4444'), (43, 'Tutor 43', 'English', '433-444-5555'), (44, 'Tutor 44', 'History', '544-555-6666'), (45, 'Tutor 45', 'Geography', '655-666-7777'), (46, 'Tutor 46', 'Computer Science', '766-777-8888'), (47, 'Tutor 47', 'Economics', '877-888-9999'), (48, 'Tutor 48', 'Psychology', '988-999-0000'), (49, 'Tutor 49', 'Sociology', '099-000-1111'), (50, 'Tutor 50', 'Political Science', '100-111-2222');
```

[[Edit](#)]

✓ 50 rows inserted. (Query took 0.0586 seconds.)

```
INSERT INTO student (student_id, student_name, contact_info, address) VALUES (1, 'John Doe', 'john.doe@example.com', '123 Main Street, CityA'), (2, 'Jane Smith', 'jane.smith@example.com', '456 Elm Street, CityB'), (3, 'Michael Johnson', 'michael.johnson@example.com', '789 Oak Street, CityC'), (4, 'Emily Brown', 'emily.brown@example.com', '321 Pine Street, CityD'), (5, 'Christopher Lee', 'christopher.lee@example.com', '654 Maple Street, CityE'), (6, 'Amanda Wilson', 'amanda.wilson@example.com', '987 Birch Street, CityF'), (7, 'David Martinez', 'david.martinez@example.com', '159 Cedar Street, CityG'), (8, 'Jessica Taylor', 'jessica.taylor@example.com', '753 Walnut Street, CityH'), (9, 'Daniel Anderson', 'daniel.anderson@example.com', '246 Spruce Street, CityI'), (10, 'Sarah Thomas', 'sarah.thomas@example.com', '864 Cherry Street, CityJ'), (11, 'Matthew Hernandez', 'matthew.hernandez@example.com', '357 Pine Street, CityK'), (12, 'Olivia Garcia', 'olivia.garcia@example.com', '468 Ash Street, CityL'), (13, 'Noah White', 'noah.white@example.com', '579 Birch Street, CityM'), (14, 'Sophia Green', 'sophia.green@example.com', '680 Elm Street, CityN'), (15, 'Liam Black', 'liam.black@example.com', '791 Oak Street, CityO'), (16, 'Isabella Gray', 'isabella.gray@example.com', '802 Pine Street, CityP'), (17, 'Mason Blue', 'mason.blue@example.com', '913 Maple Street, CityQ'), (18, 'Charlotte Brown', 'charlotte.brown@example.com', '024 Cedar Street, CityR'), (19, 'Ethan Red', 'ethan.red@example.com', '135 Birch Street, CityS'), (20, 'Amelia Gold', 'amelia.gold@example.com', '246 Walnut Street, CityT'), (21, 'Jacob Silver', 'jacob.silver@example.com', '357 Spruce Street, CityU'), (22, 'Mia Bronze', 'mia.bronze@example.com', '468 Cherry Street, CityV'), (23, 'William Platinum', 'william.platinum@example.com', '579 Ash Street, CityW'), (24, 'Harper Diamond', 'harper.diamond@example.com', '680 Birch Street, CityX'), (25, 'Benjamin Ruby', 'benjamin.ruby@example.com', '791 Elm Street, CityY'), (26, 'Evelyn Sapphire', 'evelyn.sapphire@example.com', '802 Oak Street, CityZ'), (27, 'Alexander Emerald', 'alexander.emerald@example.com', '913 Pine Street, CityAA'), (28, 'Sofia Jade', 'sofia.jade@example.com', '024 Maple Street, CityAB'), (29, 'Nathan Opal', 'nathan.opal@example.com', '135 Cedar Street, CityAC'), (30, 'Avery Pearl', 'avery.pearl@example.com', '246 Birch Street, CityAD'), (31, 'Caleb Garnet', 'caleb.garnet@example.com', '357 Walnut Street, CityAE'), (32, 'Madison Amethyst', 'madison.amethyst@example.com', '468 Spruce Street, CityAF'), (33, 'Julian Topaz', 'julian.topaz@example.com', '579 Cherry Street, CityAG'), (34, 'Lillian Citrine', 'lillian.citrine@example.com', '680 Ash Street, CityAH'), (35, 'Isaac Ruby', 'isaac.ruby@example.com', '791 Birch Street, CityAI'), (36, 'Grace Sapphire', 'grace.sapphire@example.com', '802 Elm Street, CityAJ'), (37, 'Carter Emerald', 'carter.emerald@example.com', '913 Oak Street, CityAK'), (38, 'Natalie Jade', 'natalie.jade@example.com', '024 Pine Street, CityAL'), (39, 'Dylan Opal', 'dylan.opal@example.com', '135 Maple Street, CityAM'), (40, 'Zoe Pearl', 'zoe.pearl@example.com', '246 Cedar Street, CityAN'), (41, 'Nolan Garnet', 'nolan.garnet@example.com', '357 Birch Street, CityAO'), (42, 'Hannah Amethyst', 'hannah.amethyst@example.com', '468 Walnut Street, CityAP'), (43, 'Christopher Topaz', 'christopher.topaz@example.com', '579 Spruce Street, CityAQ'), (44, 'Victoria Citrine', 'victoria.citrine@example.com', '680 Cherry Street, CityAR'), (45, 'Ethan Ruby', 'ethan.ruby@example.com', '791 Ash Street, CityAS'), (46, 'Madison Sapphire', 'madison.sapphire@example.com', '802 Birch Street, CityAT'), (47, 'Julian Emerald', 'julian.emerald@example.com', '913 Elm Street, CityAU'), (48, 'Sofia Jade', 'sofia.jade@example.com', '024 Oak Street, CityAV'), (49, 'Nathan Opal', 'nathan.opal@example.com', '135 Pine Street, CityAW'), (50, 'Avery Pearl', 'avery.pearl@example.com', '246 Maple Street, CityAX');
```

[[Edit](#)]

✔ 50 rows inserted. (Query took 0.0005 seconds.)

```
INSERT INTO assignment (assignment_id, class_id, tutor_id, title, description, deadline) VALUES (1, 1, 1, 'Assignment 1', 'Description for Assignment 1', '2024-05-15'), (2, 2, 2, 'Assignment 2', 'Description for Assignment 2', '2024-05-16'), (3, 3, 3, 'Assignment 3', 'Description for Assignment 3', '2024-05-17'), (4, 4, 4, 'Assignment 4', 'Description for Assignment 4', '2024-05-18'), (5, 5, 5, 'Assignment 5', 'Description for Assignment 5', '2024-05-19'), (6, 6, 6, 'Assignment 6', 'Description for Assignment 6', '2024-05-20'), (7, 7, 7, 'Assignment 7', 'Description for Assignment 7', '2024-05-21'), (8, 8, 8, 'Assignment 8', 'Description for Assignment 8', '2024-05-22'), (9, 9, 9, 'Assignment 9', 'Description for Assignment 9', '2024-05-23'), (10, 10, 10, 'Assignment 10', 'Description for Assignment 10', '2024-05-24'), (11, 11, 11, 'Assignment 11', 'Description for Assignment 11', '2024-05-25'), (12, 12, 12,
```

[Edit]

✔ 50 rows inserted. (Query took 0.0007 seconds.)

```
INSERT INTO attendance (attendance_id, class_id, student_id, date, status) VALUES (1, 1, 1, '2024-05-01', 'Present'), (2, 2, 2, '2024-05-01', 'Absent'), (3, 3, 3, '2024-05-01', 'Present'), (4, 4, 4, '2024-05-01', 'Present'), (5, 5, 5, '2024-05-01', 'Absent'), (6, 6, 6, '2024-05-01', 'Present'), (7, 7, 7, '2024-05-01', 'Present'), (8, 8, 8, '2024-05-01', 'Absent'), (9, 9, 9, '2024-05-01', 'Present'), (10, 10, 10, '2024-05-01', 'Present'), (11, 11, 11, '2024-05-02', 'Present'), (12, 12, 12, '2024-05-02', 'Absent'), (13, 13, 13, '2024-05-02', 'Present'), (14, 14, 14, '2024-05-02', 'Absent'), (15, 15, 15, '2024-05-02', 'Present'), (16, 16, 16, '2024-05-02', 'Present'), (17, 17, 17, '2024-05-02', 'Absent'), (18, 18, 18, '2024-05-02', 'Present'), (19, 19, 19, '2024-05-02', 'Absent'), (20, 20, 20, '2024-05-02', 'Present'), (21, 21, 21, '2024-05-03', 'Present'), (22, 22, 22, '2024-05-03', 'Absent'), (23, 23, 23, '2024-05-03',
```

[Edit]

✔ 50 rows inserted. (Query took 0.0006 seconds.)

```
-- Assuming the classbatch table already exists -- Add 50 non-empty rows of data INSERT INTO classbatch (batch_id, class_id, batch_name, start_date, end_date) VALUES (1, 1, 'Batch 1', '2024-01-01', '2024-02-01'), (2, 2, 'Batch 2', '2024-01-02', '2024-02-02'), (3, 3, 'Batch 3', '2024-01-03', '2024-02-03'), (4, 4, 'Batch 4', '2024-01-04', '2024-02-04'), (5, 5, 'Batch 5', '2024-01-05', '2024-02-05'), (6, 6, 'Batch 6', '2024-01-06', '2024-02-06'), (7, 7, 'Batch 7', '2024-01-07', '2024-02-07'), (8, 8, 'Batch 8', '2024-01-08', '2024-02-08'), (9, 9, 'Batch 9', '2024-01-09', '2024-02-09'), (10, 10, 'Batch 10', '2024-01-10', '2024-02-10'), (11, 11, 'Batch 11', '2024-01-11', '2024-02-11'), (12, 12, 'Batch 12', '2024-01-12', '2024-02-12'), (13, 13, 'Batch 13', '2024-01-13', '2024-02-13'), (14, 14, 'Batch 14', '2024-01-14', '2024-02-14'), (15, 15, 'Batch 15', '2024-01-15', '2024-02-15'), (16, 16, 'Batch 16', '2024-01-16', '2024-02-16'), (17, 17, 'Batch 17', '2024-01-17', '2024-02-17'), (18, 18, 'Batch 18', '2024-01-18', '2024-02-18'), (19, 19, 'Batch 19', '2024-01-19', '2024-02-19'), (20, 20, 'Batch 20', '2024-01-20', '2024-02-20'), (21, 21, 'Batch 21', '2024-01-21', '2024-02-21'), (22, 22, 'Batch 22', '2024-01-22', '2024-02-22'), (23, 23, 'Batch 23', '2024-01-23', '2024-02-23'), (24, 24, 'Batch 24', '2024-01-24', '2024-02-24'), (25, 25, 'Batch 25', '2024-01-25', '2024-02-25'), (26, 26, 'Batch 26', '2024-01-26', '2024-02-26'), (27, 27, 'Batch 27', '2024-01-27', '2024-02-27'), (28, 28, 'Batch 28', '2024-01-28', '2024-02-28'), (29, 29, 'Batch 29', '2024-01-29', '2024-02-29'), (30, 30, 'Batch 30', '2024-01-30', '2024-02-30'), (31, 31, 'Batch 31', '2024-02-01', '2024-02-31'), (32, 32, 'Batch 32', '2024-02-02', '2024-03-01'), (33, 33, 'Batch 33', '2024-02-03', '2024-03-02'), (34, 34, 'Batch 34', '2024-02-04', '2024-03-03'), (35, 35, 'Batch 35', '2024-02-05', '2024-03-04'), (36, 36, 'Batch 36', '2024-02-06', '2024-03-05'), (37, 37, 'Batch 37', '2024-02-07', '2024-03-06'), (38, 38, 'Batch 38', '2024-02-08', '2024-03-07'), (39, 39, 'Batch 39', '2024-02-09', '2024-03-08'), (40, 40, 'Batch 40', '2024-02-10', '2024-03-09'), (41, 41, 'Batch 41', '2024-02-11', '2024-03-10'), (42, 42, 'Batch 42', '2024-02-12', '2024-03-11'), (43, 43, 'Batch 43', '2024-02-13', '2024-03-12'), (44, 44, 'Batch 44', '2024-02-14', '2024-03-13'), (45, 45, 'Batch 45', '2024-02-15', '2024-03-14'), (46, 46, 'Batch 46', '2024-02-16', '2024-03-15'), (47, 47, 'Batch 47', '2024-02-17', '2024-03-16'), (48, 48, 'Batch 48', '2024-02-18', '2024-03-17'), (49, 49, 'Batch 49', '2024-02-19', '2024-03-18'), (50, 50, 'Batch 50', '2024-02-20', '2024-03-19');
```

[Edit]

✔ 50 rows inserted. (Query took 0.0009 seconds.)

```
-- Assuming the classenrollment table already exists -- Add 50 nonempty rows of data INSERT INTO classenrollment (enrollment_id, batch_id, student_id, enrollment_date) VALUES (1, 1, 1, '2024-01-01'), (2, 2, 2, '2024-01-02'), (3, 3, 3, '2024-01-03'), (4, 4, 4, '2024-01-04'), (5, 5, 5, '2024-01-05'), (6, 6, 6, '2024-01-06'), (7, 7, 7, '2024-01-07'), (8, 8, 8, '2024-01-08'), (9, 9, 9, '2024-01-09'), (10, 10, 10, '2024-01-10'), (11, 11, 11, '2024-01-11'), (12, 12, 12, '2024-01-12'), (13, 13, 13, '2024-01-13'), (14, 14, 14, '2024-01-14'), (15, 15, 15, '2024-01-15'), (16, 16, 16, '2024-01-16'), (17, 17, 17, '2024-01-17'), (18, 18, 18, '2024-01-18'), (19, 19, 19, '2024-01-19'), (20, 20, 20, '2024-01-20'), (21, 21, 21, '2024-01-21'), (22, 22, 22, '2024-01-22'), (23, 23, 23, '2024-01-23'), (24, 24, 24, '2024-01-24'), (25, 25, 25, '2024-01-25'), (26, 26, 26, '2024-01-26'), (27, 27, 27, '2024-01-27'), (28, 28, 28, '2024-01-28'), (29, 29, 29, '2024-01-29'), (30, 30, 30, '2024-01-30'), (31, 31, 31, '2024-02-01'), (32, 32, 32, '2024-02-02'), (33, 33, 33, '2024-02-03'), (34, 34, 34, '2024-02-04'), (35, 35, 35, '2024-02-05'), (36, 36, 36, '2024-02-06'), (37, 37, 37, '2024-02-07'), (38, 38, 38, '2024-02-08'), (39, 39, 39, '2024-02-09'), (40, 40, 40, '2024-02-10'), (41, 41, 41, '2024-02-11'), (42, 42, 42, '2024-02-12'), (43, 43, 43, '2024-02-13'), (44, 44, 44, '2024-02-14'), (45, 45, 45, '2024-02-15'), (46, 46, 46, '2024-02-16'), (47, 47, 47, '2024-02-17'), (48, 48, 48, '2024-02-18'), (49, 49, 49, '2024-02-19'), (50, 50, 50, '2024-02-20');
```

[Edit]

✔ 50 rows inserted. (Query took 0.0006 seconds.)

```
-- Assuming the grade table already exists -- Add 50 nonempty rows of data INSERT INTO grade (grade_id, assignment_id, student_id, grade_value) VALUES (1, 1, 1, 85), (2, 2, 2, 90), (3, 3, 3, 75), (4, 4, 4, 88), (5, 5, 5, 92), (6, 6, 6, 80), (7, 7, 7, 87), (8, 8, 8, 78), (9, 9, 9, 85), (10, 10, 10, 91), (11, 11, 11, 83), (12, 12, 12, 89), (13, 13, 13, 86), (14, 14, 14, 79), (15, 15, 15, 94), (16, 16, 16, 82), (17, 17, 17, 88), (18, 18, 18, 75), (19, 19, 19, 90), (20, 20, 20, 84), (21, 21, 21, 91), (22, 22, 22, 77), (23, 23, 23, 85), (24, 24, 24, 92), (25, 25, 25, 81), (26, 26, 26, 87), (27, 27, 27, 76), (28, 28, 28, 88), (29, 29, 29, 93), (30, 30, 30, 80), (31, 31, 31, 85), (32, 32, 32, 91), (33, 33, 33, 79), (34, 34, 34, 89), (35, 35, 35, 84), (36, 36, 36, 90), (37, 37, 37, 77), (38, 38, 38, 91), (39, 39, 39, 86), (40, 40, 40, 83), (41, 41, 41, 78), (42, 42, 42, 95), (43, 43, 43, 80), (44, 44, 44, 87), (45, 45, 45, 74), (46, 46, 46, 92), (47, 47, 47, 89), (48, 48, 48, 76), (49, 49, 49, 94), (50, 50, 50, 82);
```

[Edit]

1. How many transactions are there in the transaction table?

Your SQL query has been executed successfully.

```
SELECT COUNT(*) AS transaction_count FROM transaction;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

[Extra options](#)

transaction_count
50

[Query results operations](#)

2. What is the average amount of transactions in the transaction table?

Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT AVG(amount) AS average_amount FROM transaction;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows:

[Extra options](#)

average_amount
301.197000

☐ Show all | Number of rows: 25 | Filter rows:

3. Which student has the highest transaction amount in the transaction table?

Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT student_id, MAX(amount) AS highest_amount FROM transaction;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows:

[Extra options](#)

student_id	highest_amount
1	720.25

☐ Edit ☐ Copy ☐ Delete

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all | Number of rows: 25 | Filter rows:

[Query results operations](#)

4. How many batches are there in the classbatch table?

Your SQL query has been executed successfully.

```
SELECT COUNT(*) AS batch_count FROM classbatch;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

[Extra options](#)

batch_count
50

[Query results operations](#)

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

5. What is the earliest start date of batches in the classbatch table?

Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT MIN(start_date) AS earliest_start_date FROM classbatch;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

earliest_start_date
2024-01-01

☐ Show all | Number of rows: 25 | Filter rows:

Query results operations

6. Which student has enrolled in the maximum number of batches in the classenrollment table?

Your SQL query has been executed successfully.

```
SELECT student_id, COUNT(*) AS enrollment_count FROM classenrollment GROUP BY student_id ORDER BY enrollment_count DESC LIMIT 1;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Extra options

student_id	enrollment_count
9	1

Query results operations

7. How many unique assignments are there in the grade table?

Your SQL query has been executed successfully.

```
SELECT COUNT(DISTINCT assignment_id) AS unique_assignment_count FROM grade;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Extra options

unique_assignment_count
50

Query results operations

8. What is the average grade value across all assignments in the grade table?

Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT AVG(grade_value) AS average_grade_value FROM grade;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

average_grade_value
85.140000

☐ Show all | Number of rows: 25 | Filter rows:

9. Which student has the highest average grade value in the grade table?

Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)

```
SELECT student_id, AVG(grade_value) AS average_grade FROM grade GROUP BY student_id ORDER BY average_grade DESC LIMIT 1;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Extra options

	student_id	average_grade
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	94.000000

☐ Check all With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

Query results operations

☐ Print ☐ Copy to clipboard ☐ Export ☐ Display chart ☐ Create view

☐ Bookmark this SQL query

10. How many students have a grade value greater than 90 in the grade table?

Your SQL query has been executed successfully.

```
SELECT COUNT(*) AS high_grades_count FROM grade WHERE grade_value > 90;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Extra options

high_grades_count
10

Query results operations

☐ Print ☐ Copy to clipboard ☐ Export ☐ Display chart ☐ Create view

11. What is the total amount spent in transactions in the transaction table?

Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT SUM(amount) AS total_spent FROM transaction;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

total_spent
15059.85

☐ Show all | Number of rows: 25 | Filter rows: Search this table

12. Which batch has the longest duration in the classbatch table?

✓ Showing rows 0 - 0 (1 total, Query took 0.0008 seconds.)

```
SELECT batch_id, DATEDIFF(end_date, start_date) AS duration_days FROM classbatch ORDER BY duration_days DESC LIMIT 1;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Extra options

	batch_id	duration_days
<input type="checkbox"/> Edit Copy Delete	2	31

[↑](#) ☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

Query results operations

13. How many students have enrolled in all batches in the classenrollment table?

Your SQL query has been executed successfully.

```
SELECT COUNT(DISTINCT student_id) AS students_with_all_enrollments FROM classenrollment;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Extra options

students_with_all_enrollments

50

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

[Bookmark this SQL query](#)

14. What is the average enrollment count per student in the classenrollment table?

✓ Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT AVG(enrollment_count) AS average_enrollments_per_student FROM ( SELECT student_id, COUNT(*) AS enrollment_count FROM classenrollment GROUP BY student_id ) AS enrollment_counts;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

average_enrollments_per_student

1.0000

☐ Show all | Number of rows: 25 | Filter rows: Search this table

15. What is the total number of assignments submitted in the grade table?

⚠️ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

Your SQL query has been executed successfully.

```
SELECT COUNT(*) AS total_assignments FROM grade;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Extra options

total_assignments
50

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

[Bookmark this SQL query](#)

16. Which batch has the highest number of enrolled students in the classenrollment table?

Your SQL query has been executed successfully.

```
SELECT batch_id, COUNT(DISTINCT student_id) AS enrolled_students FROM classenrollment GROUP BY batch_id ORDER BY enrolled_students DESC LIMIT 1;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Extra options

batch_id	enrolled_students
16	1

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

[Bookmark this SQL query](#)

17. What is the maximum grade value in the grade table?

✓ Showing rows 0 - 0 (1 total, Query took 0.0013 seconds.)

```
SELECT MAX(grade_value) AS max_grade_value FROM grade;
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

max_grade_value
94.00

☐ Show all | Number of rows: 25 | Filter rows:

Query results operations

18. What is the average grade value for a specific assignment in the grade table?

Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT assignment_id, AVG(grade_value) AS average_grade_value FROM grade WHERE assignment_id = 'specific_assignment_id';
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	assignment_id	average_grade_value
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	NULL	NULL

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table

19. How many transactions were made on a specific date in the transaction table?

Your SQL query has been executed successfully.

```
SELECT COUNT(*) AS transaction_count FROM transaction WHERE transaction_date = 'YYYY-MM-DD';
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Extra options

transaction_count
0

Query results operations

☐ Print ☐ Copy to clipboard ☐ Export ☐ Display chart ☐ Create view

20. What is the earliest enrollment date for a specific student in the classenrollment table?

Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT MIN(enrollment_date) AS earliest_enrollment_date FROM classenrollment WHERE student_id = 'specific_student_id';
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

earliest_enrollment_date
NULL

☐ Show all | Number of rows: 25 | Filter rows: Search this table