BUAN 6320 DATABASE FOUNDATIONS FOR BUSINESS ANALYTICS

EQUINOX LEARNING – AN ONLINE LEARNING PLATFO

(GROUP – 3)

Group Members

Moksha Rajanna – MXR230035

Amulya Reddy – SXR230104

Ramya Javvadi – KXJ230012

Tarun Raghu – TXR23002

Siva Srinivas Narra – SXN230069

PROJECT CHARTER: ONLINE LEARNING PLATFORM

OVERVIEW:

An online learning platform is an information system designed to deliver educational content and resources via the Internet. Unlike traditional classroom learning, it offers an accessible, inclusive, diverse, and technology-driven approach to meet the learning needs of users while enhancing their skill sets.

OBJECTIVES:

Diversity: Present learners with a broad selection of courses across a range of subjects, encompassing academic topics such as mathematics, vocational skills like stitching, and life skills such as cooking.

Accessibility: We offer high-quality course content that can be downloaded on multiple devices, including tablets, laptops, and smartphones, with optimized data usage for seamless access, even in low-bandwidth environments.

Inclusivity: The content is designed to meet the needs of learners of all ages and skill levels, whether they are beginners, novice programmers, or experienced professionals seeking to refresh their skills. The platform offers cost-effective solutions with flexible subscription plans.

Technology-driven: Innovative features like an AI chatbot, live expert sessions for resolving doubts, online forum discussion boards, prerecorded videos, integrated coding infrastructure, and social media controls.

STAKEHOLDERS:

Learners/Students: Individuals who access and utilize the platform to acquire new knowledge, develop skills, and further their education

Course Instructors/ Content Creators: Professionals who develop and publish courses on the platform, providing educational content and expertise to learners

Platform Administrators: Personnel responsible for managing the technical aspects of the platform, including maintenance, updates, user support, and ensuring smooth operation

Educational Institutions: Schools, colleges, and universities that may collaborate with the platform to offer courses or integrate online learning into their curriculum

FUNCTIONS:

User Profile Management: Manage user accounts, which encompasses a range of tasks such as registration, login, subscription, and profile management. Effective management of user accounts is essential for ensuring a seamless and personalized learning experience. These profiles can also be linked to the student profiles created by the respective educational institutions.

User Registration: It involves maintaining a database of user information for individuals who sign up on the platform. These users can vary in roles, including students, content creators, and IT support specialists. The platform collects and stores various details about each user, such as their first name, last name, password, address, designation, work experience and others. This information is essential for providing personalized experiences and ensuring the security and integrity of the platform.

Course Management: It involves organizing and categorizing courses based on numerous factors such as the level of the course and the technology used. With the help of advanced search algorithms, users can easily find free and paid courses and access content that aligns with their interests and needs. Additionally, the platform records vital details such as course ratings and the number of users registered for each course, providing valuable insights into the popularity and quality of the content.

Course Assessment and Grading: The online learning platform assesses student learning through quizzes, tests, and others. It stores questions, answers, grades, and other relevant data for each assessment. Grading reports provide feedback on student performance and recommend improvement areas with relevant resources. Furthermore, the platform allows learners to track their progress by displaying the number of learning hours spent on each course, completion status in percentages, and course feedback.

Course Advertising and Promotions: This involves various strategies to attract and engage learners. The platform integrates social media icons to promote courses, share success stories, and engage with learners through social media channels to reach a wider audience. Platforms promote the courses through blog posts and webinars related to their courses. It also records the details like coupon codes and discounts on the course for the courses registered through affiliate links.

User Support Tickets: Users encountering issues with the platform can easily access customer IT support. They submit a ticket promptly received by support agents who prioritize based on urgency. Clear communication and regular updates ensure users are informed throughout the resolution process. Ticket status can be tracked anytime from the website.

Payroll System: Employee payroll data such as pay slips, tax information, and other information is stored on a separate table, where monthly transactions are recorded. Incentives are given based on course popularity and course feedback. Employee bonuses will be given to employees annually based on course ratings, number of enrolled users, and course feedback.

DELIVERABLES:

To create an online application featuring a repository of courses and pertinent course content crafted by professionals, accessible to learners via the Internet for skill-building purposes.

TIMELINE:

Weeks 1-2: Initial Project Planning

- Define project scope, functions and objectives
- Prepare a project charter for client approval
- Identify target customers who will utilize the learning platform

Weeks 3-4: Initial Database Design

- Create tables based on functionalities offered
- Assign proper data types and constraints to the tables designed
- Create Entity Relationship diagrams to visualize the structure

Weeks 5-8 Database design

- Populate mock data in the database objects
- Create major queries and store procedures relevant to the project
- Modify database schema based on testing feedback on mock data

Weeks 9-10 Security and Optimization

- Develop functions for custom operations and performance improvement
- Develop triggers for setting alerts and audits
- Fine-tune queries for tables, major queries, stored procedures, and triggers

Weeks 11-12 Documentation and Deployment

- Document the database schema and deploy the project in a staging environment
- Prepare training material and support needed to understand the platform
- Plan roll-out in the production environment after testing is complete

Post Deployment

- Collect feedback from end users to identify issues seen post-deployment
- Monitor system performance and ensure compliance with relevant policies
- Perform upgrades based on evolved requirements and advanced infrastructure availability

APPROVAL:

This project charter is approved by:

Prof. Kannan Srikanth

Professor at The University of Texas at Dallas

Date: 7th May 2024

Meeting Log:

Meeting Date & Time: 02/12/2024, 8:00 PM – 8:30 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- What is an online learning platform
- Which are the prominent learning platforms
- How to document the project charter

Meeting Date & Time: 02/14/2024, 4:00 PM – 5:30 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Overview, scope, and objectives
- Assigned tasks to individual members to document the charter

Meeting Date & Time: 02/16/2024, 5 PM – 6:15 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Reviewed the project charter documents
- Discussed in detail the features of the online learning platform

Meeting Date & Time: 02/19/2024, 5 PM – 6:15 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Reviewed the functions of the online learning platform
- Reviewed the timeline of the project

Meeting Date & Time: 03/04/2024, 6:00 PM – 6:30 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Discussed the tables needed for the learning platform
- Created metadata for 15 tables for the EQUINOX learning platform

Meeting Date & Time: 03/13/2024, 11:00 AM – 12:30 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Populated sample data for 15 tables created
- Reviewed sample data and tables created for the platform

Meeting Date & Time: 03/25/2024, 12:40 PM – 01:40 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Designed scenarios for major queries
- Wrote 5 major queries based on the scenarios discussed

Meeting Date & Time: 04/01/2024, 10:30 AM – 12:00 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Discussed about the project video story to be made
- Completed the remaining major queries based on the scenarios discussed

Meeting Date & Time: 04/08/2024, 4:00 PM – 5:30 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Discussed about stored procedures needed for the platform
- Repopulated data in the tables to write meaningful stored procedures

Meeting Date & Time: 04/16/2024, 5:00 PM – 8:00 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Recorded project video
- Created a dummy website to feature in the project video

Meeting Date & Time: 04/23/2024, 1:00 PM – 3:00 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Discussed stored procedures needed
- Created 5 stored procedures as the scenarios discussed

Meeting Date & Time: 04/29/2024, 6:00 PM to 10:00 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Discussed about the functions and triggers needed
- Created 5 functions and 5 stored procedures as per the scenarios discussed

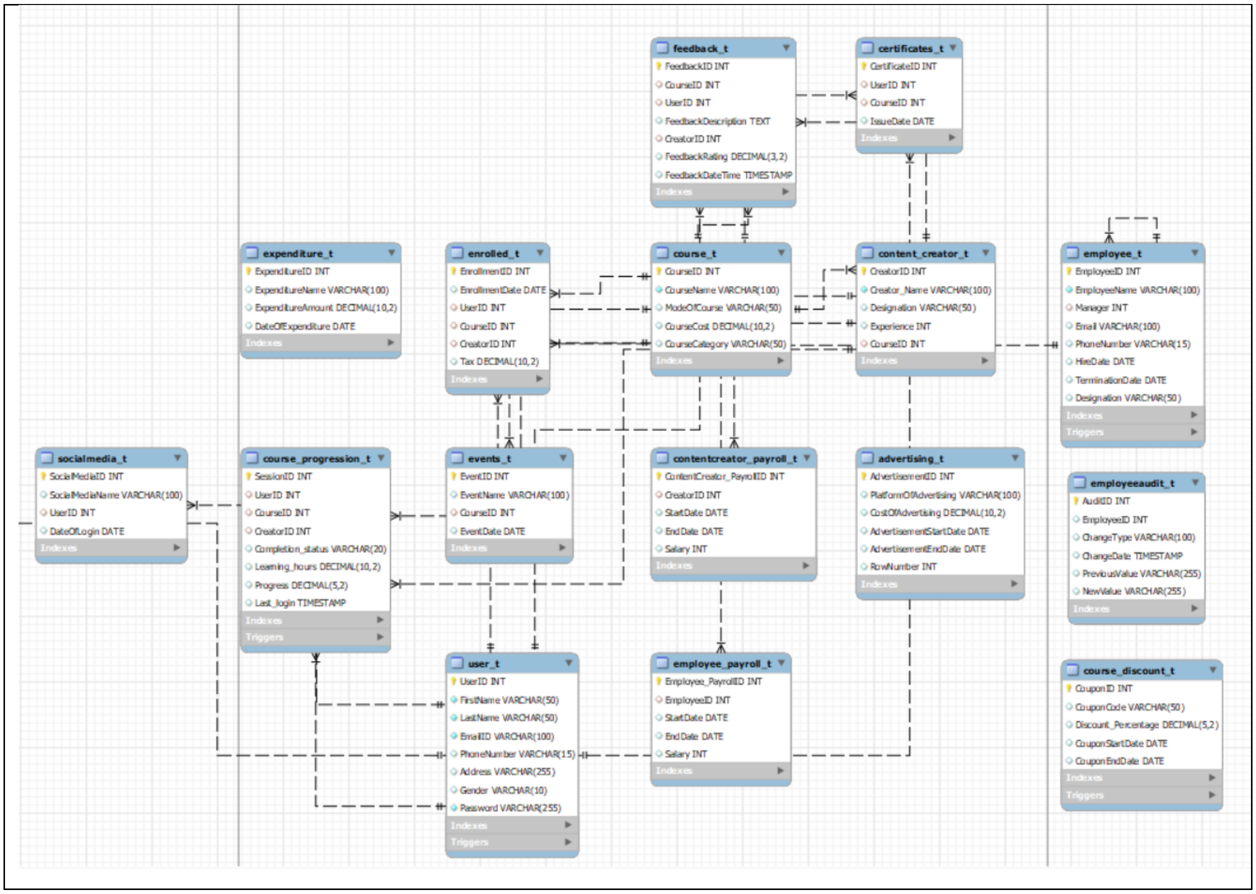
Meeting Date & Time: 04/30/2024, 4:00 PM to 6:30 PM

Attendees: Amulya, Tarun, Ramya, Srinivas, Moksha

Minutes of the meeting:

- Discussed about the project presentation
- Rehearsed for the project presentation

E-R Diagram:

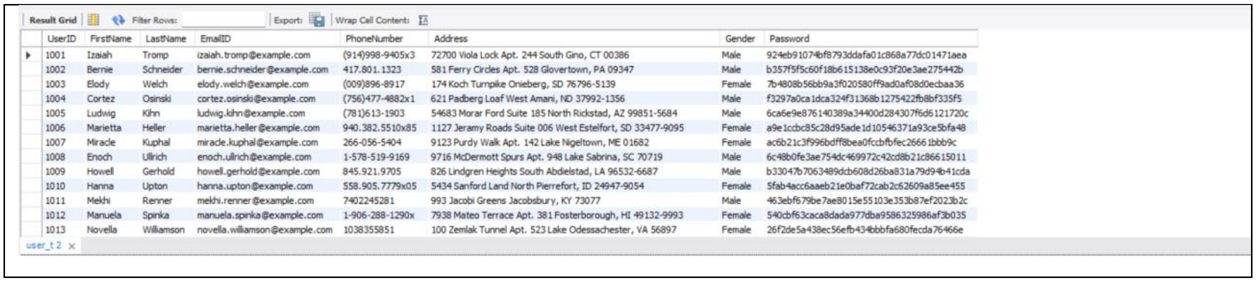


Complex Queries:

1) Query Description:

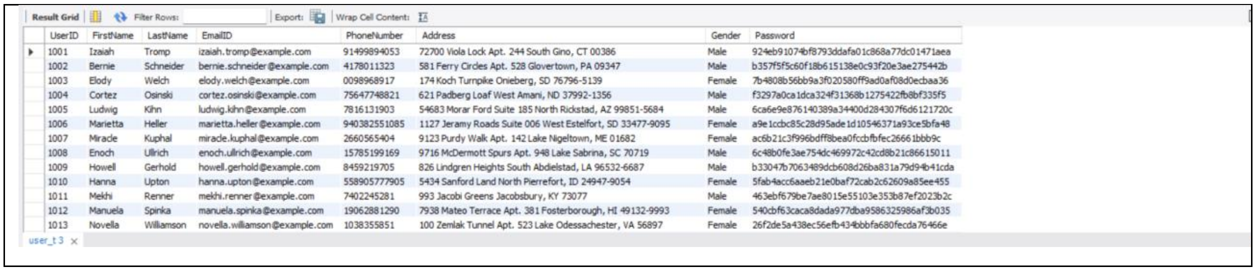
Using regular expressions, utilize a pattern-matching technique to filter special characters and text characters in the PhoneNumber field of the user_t table. In this scenario, we aim to extract only entries containing valid phone numbers, excluding any non-numeric characters or alphabetic characters.

Data before execution:



UserID	FirstName	LastName	EmailID	PhoneNumber	Address	Gender	Password
1001	Izaiah	Tromp	izaiah.tromp@example.com	(914)998-9405x3	72700 Viola Lock Apt. 244 South Gino, CT 00386	Male	924eb91074bf8793ddafa01c868a77dc01471aea
1002	Bernie	Schneider	bernie.schneider@example.com	417.801.1323	581 Ferry Circles Apt. 528 Glovertown, PA 09347	Male	b357f5f5c60f18b615138e0c93f20e3ae275442b
1003	Elody	Welch	elody.welch@example.com	(009)896-8917	174 Koch Turnpike Onieberg, SD 76796-5139	Female	7b4808b56bb9a3f020580ff9ad0af08d0ecbaa36
1004	Cortez	Osinski	cortez.osinski@example.com	(756)477-4882x1	621 Padberg Loaf West Amani, ND 37992-1356	Male	f3297a0ca1dca324f31368b1275422fb8bf335f5
1005	Ludwig	Kihn	ludwig.kihn@example.com	(781)613-1903	54683 Morar Ford Suite 185 North Rickstad, AZ 99851-5684	Male	6ca6e9e876140389a34400d284307f6d6121720c
1006	Marietta	Heller	marietta.heller@example.com	940.382.5510x85	1127 Jeramy Roads Suite 006 West Estelfort, SD 33477-9095	Female	a9e1ccbc85c28d95ade1d10546371a93ce5bfa48
1007	Miracle	Kuphal	miracle.kuphal@example.com	266-056-5404	9123 Purdy Walk Apt. 142 Lake Nigeltown, ME 01682	Female	ac6b21c3f996bdff8bea0fccbfbfec26661bbb9c
1008	Enoch	Ullrich	enoch.ullrich@example.com	1-578-519-9169	9716 McDermott Spurs Apt. 948 Lake Sabrina, SC 70719	Male	6c48b0fe3ae754dc469972c42cd8b21c86615011
1009	Howell	Gerhold	howell.gerhold@example.com	845.921.9705	826 Lindgren Heights South Abdielstad, LA 96532-6687	Male	b33047b7063489dcb608d26ba831a79d94b41cda
1010	Hanna	Upton	hanna.upton@example.com	558.905.7779x05	5434 Sanford Land North Pierrefort, ID 24947-9054	Female	5fab4acc6aaeb21e0baf72cab2c62609a85ee455
1011	Mekhi	Renner	mekhi.renner@example.com	7402245281	993 Jacobi Greens Jacobsbury, KY 73077	Male	463ebf679be7ae8015e55103e353b87ef2023b2c
1012	Manuela	Spinka	manuela.spinka@example.com	1-906-288-1290x	7938 Mateo Terrace Apt. 381 Fosterborough, HI 49132-9993	Female	540cbf63caca8dada977dba9586325986af3b035
1013	Novella	Williamson	novella.williamson@example.com	1038355851	100 Zemlak Tunnel Apt. 523 Lake Odessachester, VA 56897	Female	26f2de5a438ec56efb434bbbfa680fecda76466e

Result after execution:

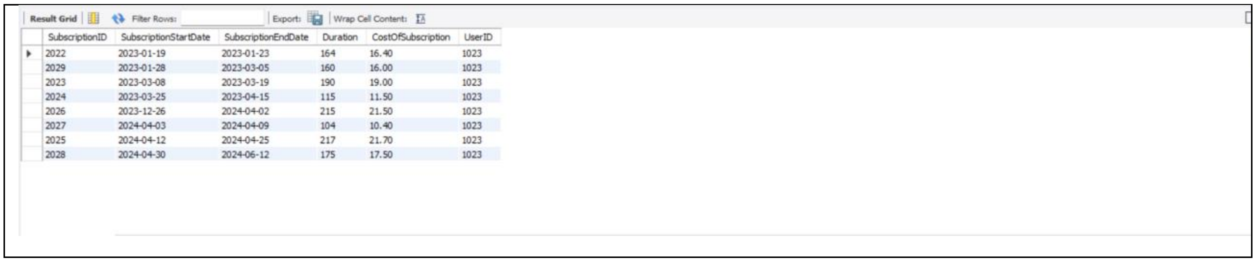


UserID	FirstName	LastName	EmailID	PhoneNumber	Address	Gender	Password
1001	Izaiah	Tromp	izaiah.tromp@example.com	91499894053	72700 Viola Lock Apt. 244 South Gino, CT 00386	Male	924eb91074bf8793ddafa01c868a77dc01471aea
1002	Bernie	Schneider	bernie.schneider@example.com	4178011323	581 Ferry Circles Apt. 528 Glovertown, PA 09347	Male	b357f5f5c60f18b615138e0c93f20e3ae275442b
1003	Elody	Welch	elody.welch@example.com	0098968917	174 Koch Turnpike Onieberg, SD 76796-5139	Female	7b4808b56bb9a3f020580ff9ad0af08d0ecbaa36
1004	Cortez	Osinski	cortez.osinski@example.com	75647748821	621 Padberg Loaf West Amani, ND 37992-1356	Male	f3297a0ca1dca324f31368b1275422fb8bf335f5
1005	Ludwig	Kihn	ludwig.kihn@example.com	7816131903	54683 Morar Ford Suite 185 North Rickstad, AZ 99851-5684	Male	6ca6e9e876140389a34400d284307f6d6121720c
1006	Marietta	Heller	marietta.heller@example.com	940382551085	1127 Jeramy Roads Suite 006 West Estelfort, SD 33477-9095	Female	a9e1ccbc85c28d95ade1d10546371a93ce5bfa48
1007	Miracle	Kuphal	miracle.kuphal@example.com	2660565404	9123 Purdy Walk Apt. 142 Lake Nigeltown, ME 01682	Female	ac6b21c3f996bdff8bea0fccbfbfec26661bbb9c
1008	Enoch	Ullrich	enoch.ullrich@example.com	15785199169	9716 McDermott Spurs Apt. 948 Lake Sabrina, SC 70719	Male	6c48b0fe3ae754dc469972c42cd8b21c86615011
1009	Howell	Gerhold	howell.gerhold@example.com	8459219705	826 Lindgren Heights South Abdielstad, LA 96532-6687	Male	b33047b7063489dcb608d26ba831a79d94b41cda
1010	Hanna	Upton	hanna.upton@example.com	558905777905	5434 Sanford Land North Pierrefort, ID 24947-9054	Female	5fab4acc6aaeb21e0baf72cab2c62609a85ee455
1011	Mekhi	Renner	mekhi.renner@example.com	7402245281	993 Jacobi Greens Jacobsbury, KY 73077	Male	463ebf679be7ae8015e55103e353b87ef2023b2c
1012	Manuela	Spinka	manuela.spinka@example.com	19062881290	7938 Mateo Terrace Apt. 381 Fosterborough, HI 49132-9993	Female	540cbf63caca8dada977dba9586325986af3b035
1013	Novella	Williamson	novella.williamson@example.com	1038355851	100 Zemlak Tunnel Apt. 523 Lake Odessachester, VA 56897	Female	26f2de5a438ec56efb434bbbfa680fecda76466e

2) Query Description:

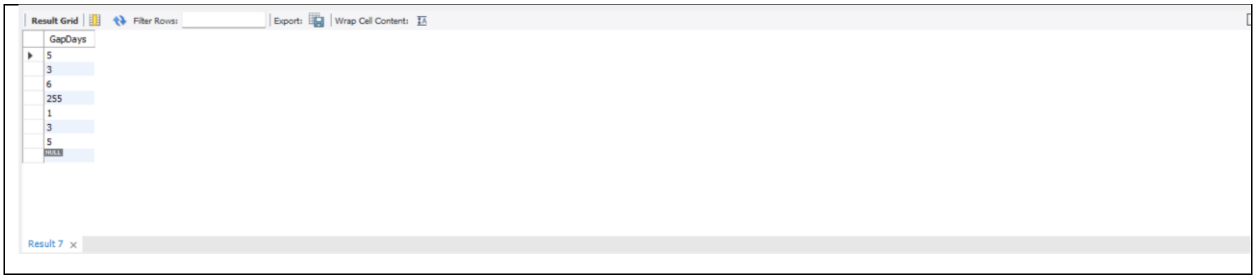
We utilize SQL to calculate the gap days in subscription for a particular user using the LEAD window function and DATEDIFF function. This method allows us to analyze the time intervals between consecutive subscription dates for a given user. The LEAD function is used to get the next subscription date for each row within the same user partition, ordered by the subscription date. DATEDIFF calculates the difference in days between the current subscription date and the next one. PARTITION BY user_id ensures the LEAD function operates within each user's subscription history. The WHERE clause filters the results for a particular user based on the user_id.

Data before execution:



SubscriptionID	SubscriptionStartDate	SubscriptionEndDate	Duration	CostOfSubscription	UserID
2022	2023-01-19	2023-01-23	164	16.40	1023
2029	2023-01-28	2023-03-05	160	16.00	1023
2023	2023-03-08	2023-03-19	190	19.00	1023
2024	2023-03-25	2023-04-15	115	11.50	1023
2026	2023-12-26	2024-04-02	215	21.50	1023
2027	2024-04-03	2024-04-09	104	10.40	1023
2025	2024-04-12	2024-04-25	217	21.70	1023
2028	2024-04-30	2024-06-12	175	17.50	1023

Result after execution:

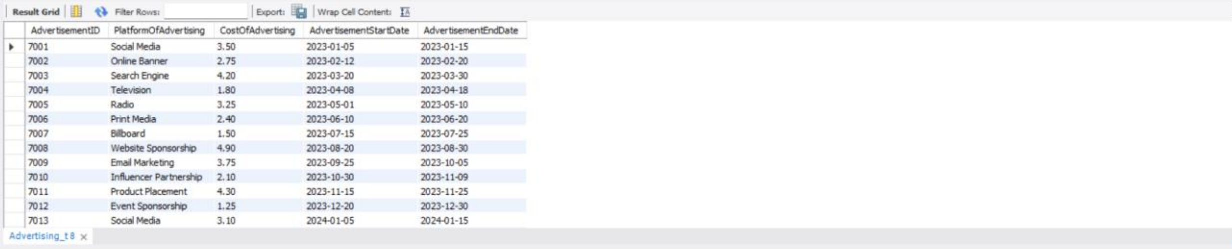


GapDays
5
3
6
255
1
3
5
NULL

3) Query Description:

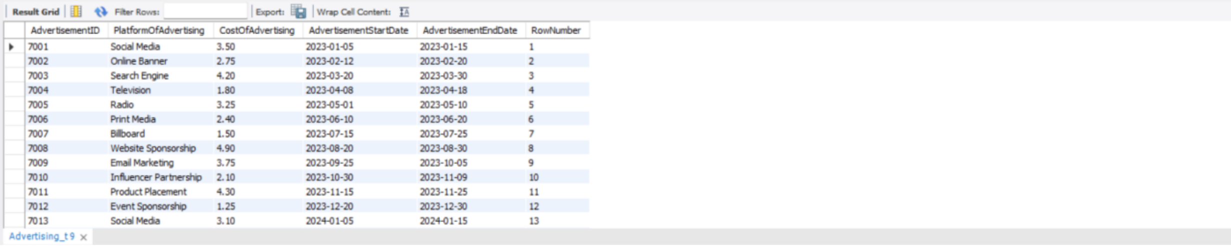
Add a row number sequence to the Advertising_t table in SQL; we use the ROW_NUMBER() window function. This function assigns a unique incremental number to each row, ordered by a specified column. The resulting table includes a new " RowNumber " column containing the generated sequence.

Data before execution:



AdvertisementID	PlatformOfAdvertising	CostOfAdvertising	AdvertisementStartDate	AdvertisementEndDate
7001	Social Media	3.50	2023-01-05	2023-01-15
7002	Online Banner	2.75	2023-02-12	2023-02-20
7003	Search Engine	4.20	2023-03-20	2023-03-30
7004	Television	1.80	2023-04-08	2023-04-18
7005	Radio	3.25	2023-05-01	2023-05-10
7006	Print Media	2.40	2023-06-10	2023-06-20
7007	Billboard	1.50	2023-07-15	2023-07-25
7008	Website Sponsorship	4.90	2023-08-20	2023-08-30
7009	Email Marketing	3.75	2023-09-25	2023-10-05
7010	Influencer Partnership	2.10	2023-10-30	2023-11-09
7011	Product Placement	4.30	2023-11-15	2023-11-25
7012	Event Sponsorship	1.25	2023-12-20	2023-12-30
7013	Social Media	3.10	2024-01-05	2024-01-15

Result after execution:

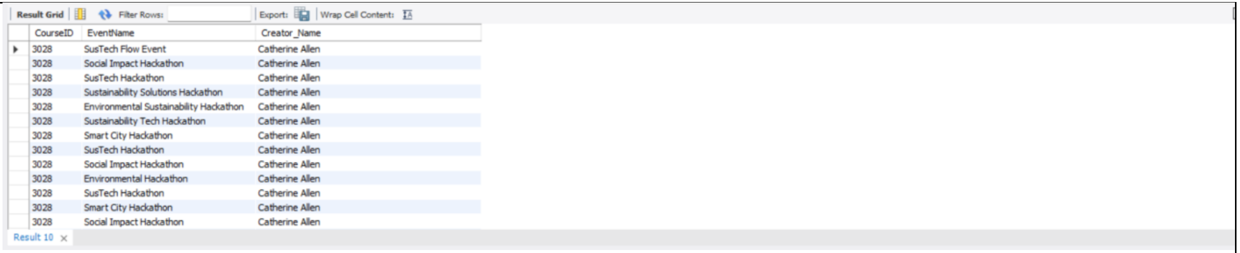


AdvertisementID	PlatformOfAdvertising	CostOfAdvertising	AdvertisementStartDate	AdvertisementEndDate	RowNumber
7001	Social Media	3.50	2023-01-05	2023-01-15	1
7002	Online Banner	2.75	2023-02-12	2023-02-20	2
7003	Search Engine	4.20	2023-03-20	2023-03-30	3
7004	Television	1.80	2023-04-08	2023-04-18	4
7005	Radio	3.25	2023-05-01	2023-05-10	5
7006	Print Media	2.40	2023-06-10	2023-06-20	6
7007	Billboard	1.50	2023-07-15	2023-07-25	7
7008	Website Sponsorship	4.90	2023-08-20	2023-08-30	8
7009	Email Marketing	3.75	2023-09-25	2023-10-05	9
7010	Influencer Partnership	2.10	2023-10-30	2023-11-09	10
7011	Product Placement	4.30	2023-11-15	2023-11-25	11
7012	Event Sponsorship	1.25	2023-12-20	2023-12-30	12
7013	Social Media	3.10	2024-01-05	2024-01-15	13

4) Query Description:

Retrieve details about the course, event, and content creator with the highest number of events; we utilize SQL joins, GROUP BY, and LIMIT clauses, along with the concept of subqueries. This involves connecting the Content_Creator_t and Events_t tables through appropriate join conditions. We group the results by course, event, and content creator, counting the number of events associated with each combination. Subsequently, we order the results by the count of events in descending order and use the LIMIT clause to retrieve only the top entry. This enables us to obtain the details of the course, event, and content creator with the maximum number of events.

Result after execution:

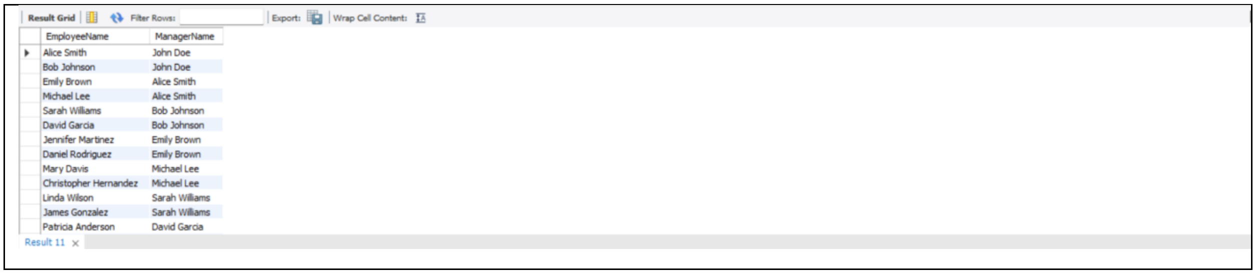


CourseID	EventName	Creator_Name
3028	SusTech Flow Event	Catherine Allen
3028	Social Impact Hackathon	Catherine Allen
3028	SusTech Hackathon	Catherine Allen
3028	Sustainability Solutions Hackathon	Catherine Allen
3028	Environmental Sustainability Hackathon	Catherine Allen
3028	Sustainability Tech Hackathon	Catherine Allen
3028	Smart City Hackathon	Catherine Allen
3028	SusTech Hackathon	Catherine Allen
3028	Social Impact Hackathon	Catherine Allen
3028	Environmental Hackathon	Catherine Allen
3028	SusTech Hackathon	Catherine Allen
3028	Smart City Hackathon	Catherine Allen
3028	Social Impact Hackathon	Catherine Allen

5) Query Description:

Retrieve employee details and their respective managers using SQL's self-join concept; we create a connection within the same table, employee_t. By joining the table to itself, we establish relationships between employees and their managers based on matching IDs. The resulting query selects the employee_id, employee_name, and manager_id columns, and the manager's name is retrieved through the self-join. Executing this query provides information about each employee alongside their manager's details.

Result after execution:

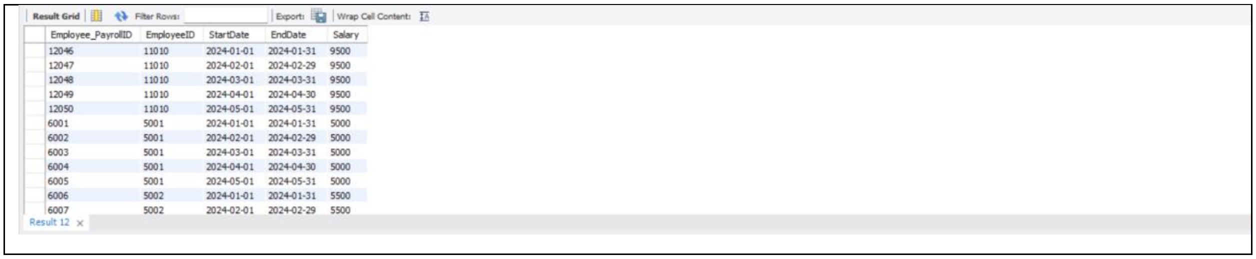


EmployeeName	ManagerName
Alice Smith	John Doe
Bob Johnson	John Doe
Emily Brown	Alice Smith
Michael Lee	Alice Smith
Sarah Williams	Bob Johnson
David Garcia	Bob Johnson
Jennifer Martinez	Emily Brown
Daniel Rodriguez	Emily Brown
Mary Davis	Michael Lee
Christopher Hernandez	Michael Lee
Linda Wilson	Sarah Williams
James Gonzalez	Sarah Williams
Patricia Anderson	David Garcia

6) Query Description:

Merge data from two tables using the UNION concept in SQL, we combine the records from Employee_payroll_t and Content_Creator_Payroll_t tables into a single result set. This involves selecting columns with similar data types from both tables and ensuring they appear in the same order. The UNION operator combines the results while removing duplicates, if any. Executing this operation provides a unified dataset containing payroll information from both tables.

Result after execution:

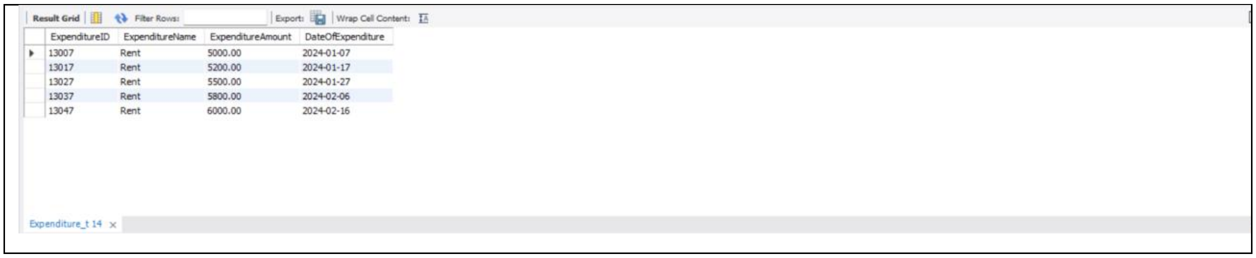


Employee_PayrollID	EmployeeID	StartDate	EndDate	Salary
12046	11010	2024-01-01	2024-01-31	9500
12047	11010	2024-02-01	2024-02-29	9500
12048	11010	2024-03-01	2024-03-31	9500
12049	11010	2024-04-01	2024-04-30	9500
12050	11010	2024-05-01	2024-05-31	9500
6001	5001	2024-01-01	2024-01-31	5000
6002	5001	2024-02-01	2024-02-29	5000
6003	5001	2024-03-01	2024-03-31	5000
6004	5001	2024-04-01	2024-04-30	5000
6005	5001	2024-05-01	2024-05-31	5000
6006	5002	2024-01-01	2024-01-31	5500
6007	5002	2024-02-01	2024-02-29	5500

7) Query Description:

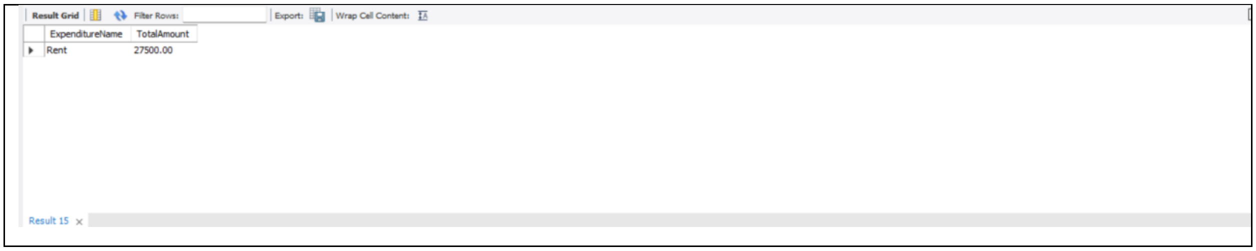
Calculate the total sum of rent expenditure using SQL's GROUP BY, SUM, and HAVING concepts; we aggregate the rent values from the expenditure_t table. First, we group the records by the rent category. Then, we apply the SUM function to compute the total rent expenditure within each group. Executing this query provides the aggregated sum value of rent expenditure.

Data before execution:



ExpenditureID	ExpenditureName	ExpenditureAmount	DateOfExpenditure
13007	Rent	5000.00	2024-01-07
13017	Rent	5200.00	2024-01-17
13027	Rent	5500.00	2024-01-27
13037	Rent	5800.00	2024-02-06
13047	Rent	6000.00	2024-02-16

Result after execution:

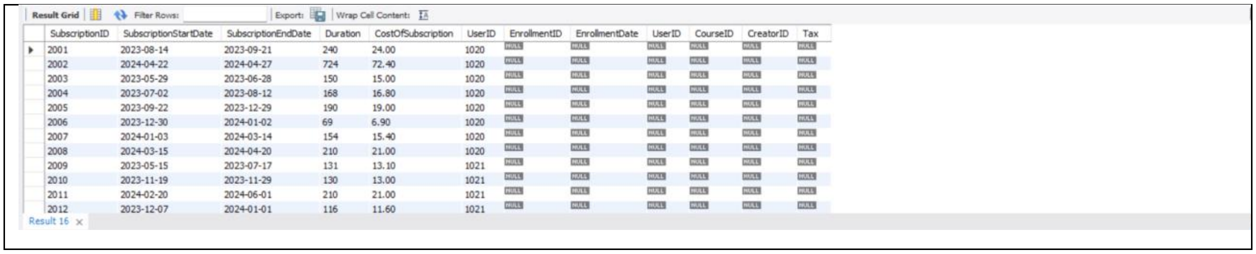


ExpenditureName	TotalAmount
Rent	27500.00

8) Query Description:

To identify users subscribed to but not enrolled in courses using SQL's LEFT JOIN and WHERE clauses, we connect the Subscription_t and Enrolled_t tables based on their user IDs. The LEFT JOIN retains all subscription records, while the WHERE clause filters out those without corresponding enrollment records. Executing this query returns the user IDs of individuals who are subscribed to courses but not enrolled.

Result after execution:

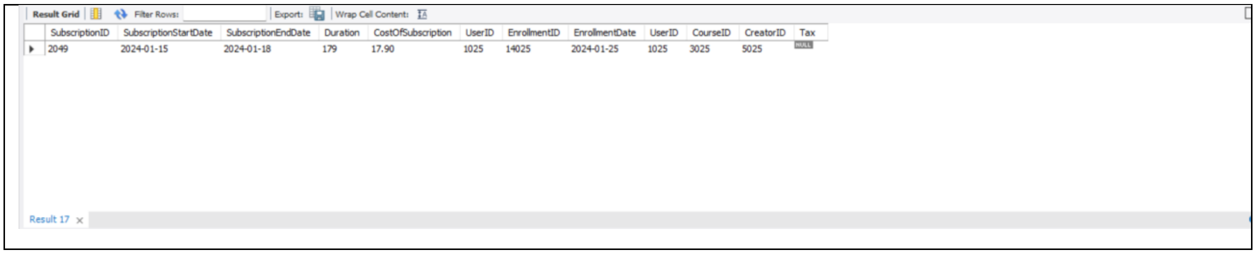


SubscriptionID	SubscriptionStartDate	SubscriptionEndDate	Duration	CostOfSubscription	UserID	EnrollmentID	EnrollmentDate	UserID	CourseID	CreatorID	Tax
2001	2023-08-14	2023-09-21	240	24.00	1020	NULL	NULL	NULL	NULL	NULL	NULL
2002	2024-04-22	2024-04-27	724	72.40	1020	NULL	NULL	NULL	NULL	NULL	NULL
2003	2023-05-29	2023-06-28	150	15.00	1020	NULL	NULL	NULL	NULL	NULL	NULL
2004	2023-07-02	2023-08-12	168	16.80	1020	NULL	NULL	NULL	NULL	NULL	NULL
2005	2023-09-22	2023-12-29	190	19.00	1020	NULL	NULL	NULL	NULL	NULL	NULL
2006	2023-12-30	2024-01-02	69	6.90	1020	NULL	NULL	NULL	NULL	NULL	NULL
2007	2024-01-03	2024-03-14	154	15.40	1020	NULL	NULL	NULL	NULL	NULL	NULL
2008	2024-03-15	2024-04-20	210	21.00	1020	NULL	NULL	NULL	NULL	NULL	NULL
2009	2023-05-15	2023-07-17	131	13.10	1021	NULL	NULL	NULL	NULL	NULL	NULL
2010	2023-11-19	2023-11-29	130	13.00	1021	NULL	NULL	NULL	NULL	NULL	NULL
2011	2024-02-20	2024-06-01	210	21.00	1021	NULL	NULL	NULL	NULL	NULL	NULL
2012	2023-12-07	2024-01-01	116	11.60	1021	NULL	NULL	NULL	NULL	NULL	NULL

9) Query Description:

Identify individuals who concurrently subscribed to a service and enrolled in a program during January. Leveraging the INNER JOIN operation and the MONTH function, it correlates data from two tables, Subscription_t and Enrolled_t. The query selects users' IDs who have both subscribed and enrolled, filtering by the month of January. Scrutinizing the subscription and enrollment dates ensures that only users meeting these criteria are included in the result set.

Result after execution:

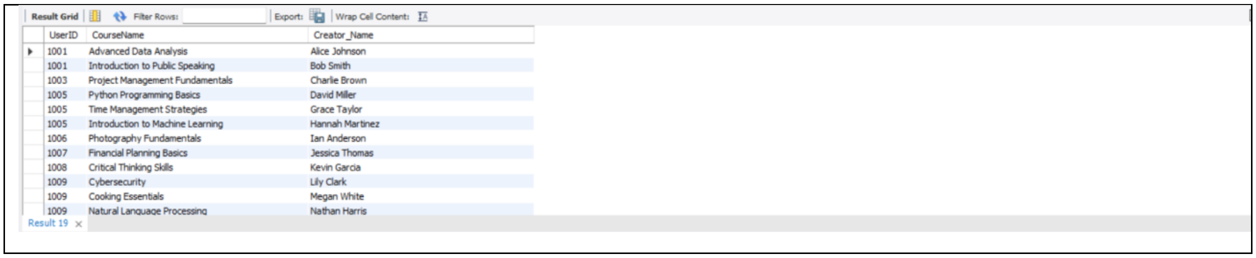


SubscriptionID	SubscriptionStartDate	SubscriptionEndDate	Duration	CostOfSubscription	UserID	EnrollmentID	EnrollmentDate	UserID	CourseID	CreatorID	Tax
2049	2024-01-15	2024-01-18	179	17.90	1025	14025	2024-01-25	1025	3025	5025	NULL

10) Query Description:

Retrieves details of users' enrollment, their enrolled courses, and information about the content creators associated with those courses. Employing JOINs, it combines data from three tables: Enrolled_t, Course_t, and Content_Creator_t. The query ensures that corresponding details from the course and content creator tables are fetched for each enrollment record. This way, the output provides a comprehensive view of user enrollments, including course specifics and the creators behind them.

Result after execution:



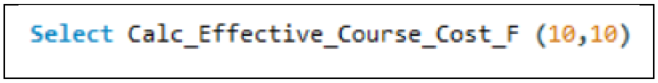
UserID	CourseName	Creator_Name
1001	Advanced Data Analysis	Alice Johnson
1001	Introduction to Public Speaking	Bob Smith
1003	Project Management Fundamentals	Charlie Brown
1005	Python Programming Basics	David Miller
1005	Time Management Strategies	Grace Taylor
1005	Introduction to Machine Learning	Hannah Martinez
1006	Photography Fundamentals	Ian Anderson
1007	Financial Planning Basics	Jessica Thomas
1008	Critical Thinking Skills	Kevin Garcia
1009	Cybersecurity	Lily Clark
1009	Cooking Essentials	Megan White
1009	Natural Language Processing	Nathan Harris

Functions:

1) Function Description:

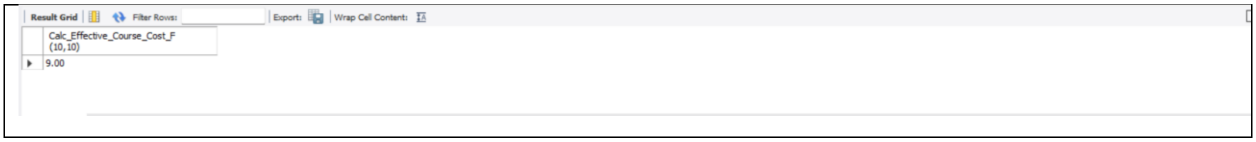
Calculates the effective cost of a course after applying a discount. It takes two input parameters: the original course cost (CourseCost) and the discount percentage (Discount_Percentage). Deducting the discount amount from the original cost determines the final price learners would pay for the course.

Use function:



```
Select Calc_Effective_Course_Cost_F (10,10)
```

Result after execution:

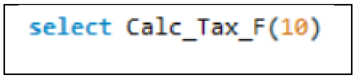


```
Calc_Effective_Course_Cost_F
(10,10)
9.00
```

2) Function Description:

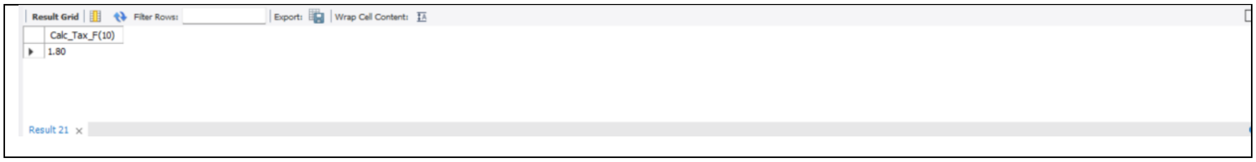
Computes the tax amount based on the course cost provided as an input parameter. It applies a tax rate of 18% to the course cost, determining the additional amount payable as tax for the course.

Use function:



```
select Calc_Tax_F(10)
```

Result after execution:

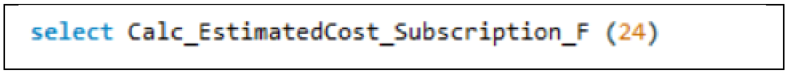


```
Calc_Tax_F(10)
1.80
```

3) Function Description:

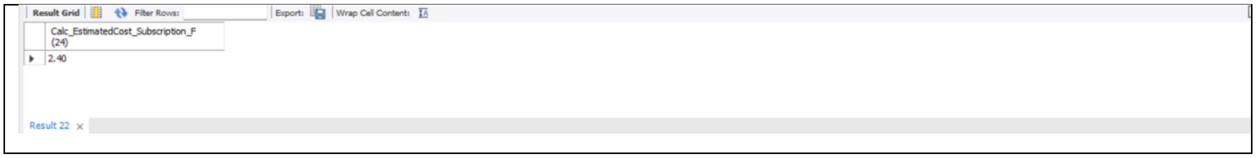
Computes the estimated cost based on a subscription's duration, with the duration specified as an input parameter. It calculates the estimated cost by multiplying the duration by 10 cents, providing an approximate total cost for the subscription period.

Use function:



```
select Calc_EstimatedCost_Subscription_F (24)
```

Result after execution:

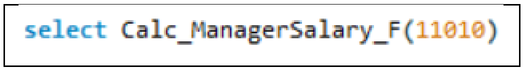


```
Calc_EstimatedCost_Subscription_F
(24)
2.40
```

4) Function Description:

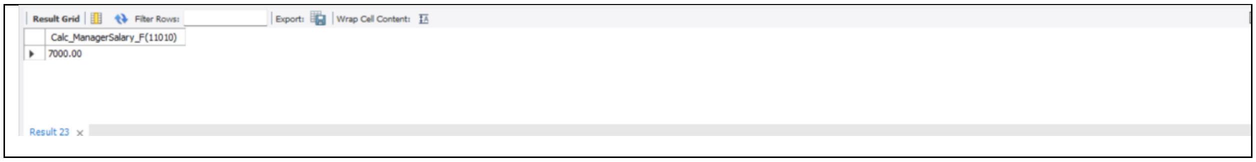
Determine the salary of a manager based on the EmployeeID of the employee who reports to them, which is provided as an input parameter.

Use function:



```
select Calc_ManagerSalary_F(11010)
```

Result after execution:

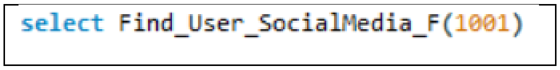


Calc_ManagerSalary_F(11010)
7000.00

5) Function Description:

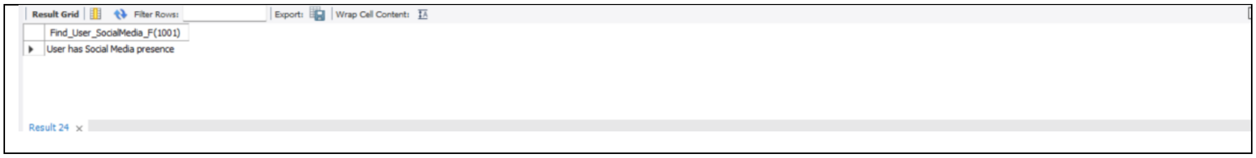
Check whether a user has visited social media pages with the user's ID specified as the input parameter.

Use function:



```sql
select Find_User_SocialMedia_F(1001)
```

Result after execution:



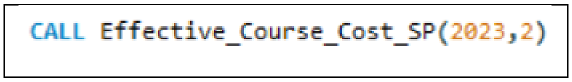
Find_User_SocialMedia_F(1001)
User has Social Media presence

Stored Procedures:

1) Procedure Description:

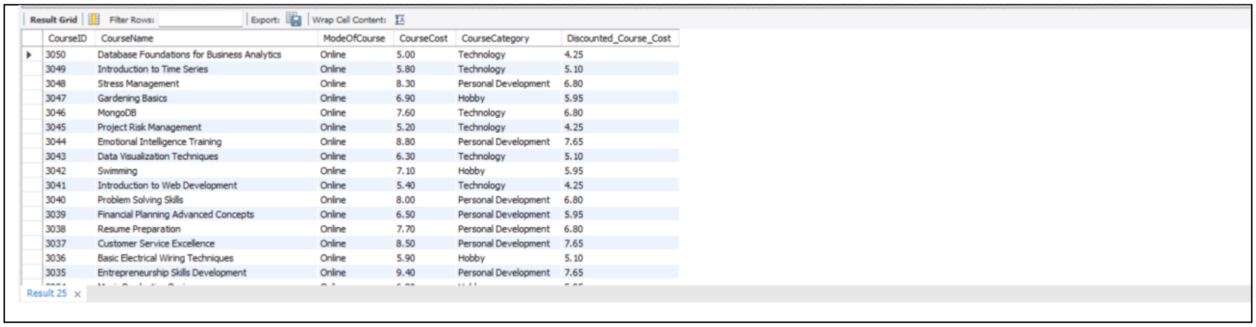
Retrieve the effective course cost after applying any discounts and other relevant course information. The year and month are provided as input parameters to filter the courses based on their availability within that specific timeframe.

Call Procedure:



```sql
CALL Effective_Course_Cost_SP(2023,2)
```

Result after execution:



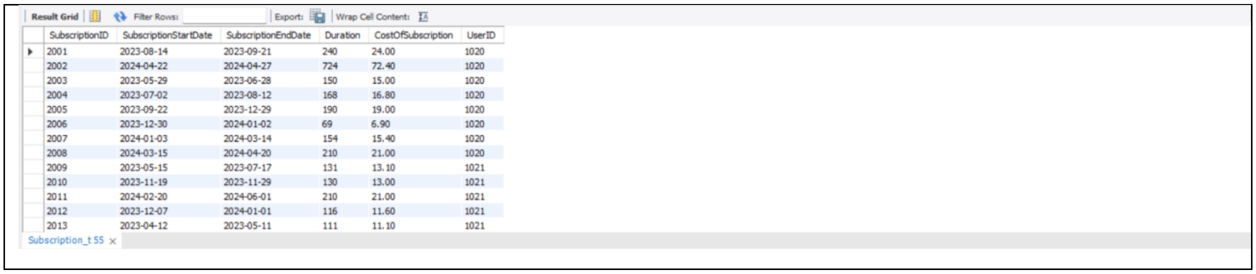
CourseID	CourseName	ModeOfCourse	CourseCost	CourseCategory	Discounted_Course_Cost
3050	Database Foundations for Business Analytics	Online	5.00	Technology	4.25
3049	Introduction to Time Series	Online	5.80	Technology	5.10
3048	Stress Management	Online	8.30	Personal Development	6.80
3047	Gardening Basics	Online	6.90	Hobby	5.95
3046	MongoDB	Online	7.60	Technology	6.80
3045	Project Risk Management	Online	5.20	Technology	4.25
3044	Emotional Intelligence Training	Online	8.80	Personal Development	7.65
3043	Data Visualization Techniques	Online	6.30	Technology	5.10
3042	Swimming	Online	7.10	Hobby	5.95
3041	Introduction to Web Development	Online	5.40	Technology	4.25
3040	Problem Solving Skills	Online	8.00	Personal Development	6.80
3039	Financial Planning Advanced Concepts	Online	6.50	Personal Development	5.95
3038	Resume Preparation	Online	7.70	Personal Development	6.80
3037	Customer Service Excellence	Online	8.50	Personal Development	7.65
3036	Basic Electrical Wiring Techniques	Online	5.90	Hobby	5.10
3035	Entrepreneurship Skills Development	Online	9.40	Personal Development	7.65

2) Procedure Description:

Update the tax information in the Enrolled Table based on the course cost, and it also modifies the duration and cost of subscriptions in the Subscription Table. Unlike a function, this procedure does not require any input parameters.

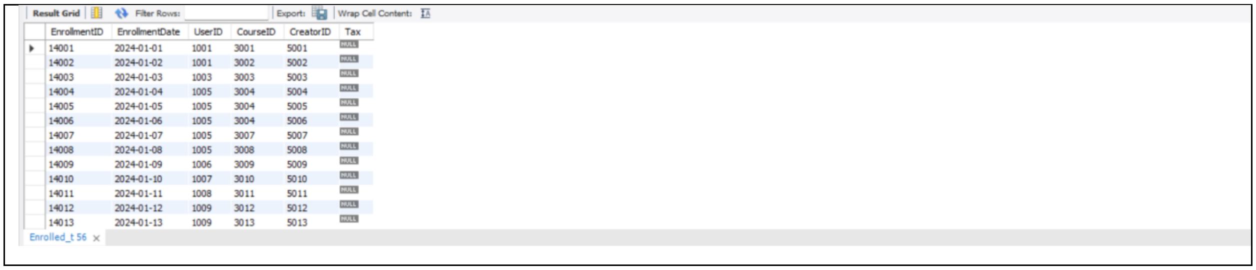
Data before execution:

Subscription_t



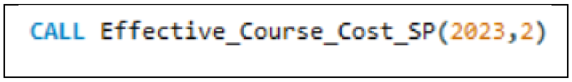
SubscriptionID	SubscriptionStartDate	SubscriptionEndDate	Duration	CostOfSubscription	UserID
2001	2023-08-14	2023-09-21	240	24.00	1020
2002	2024-04-22	2024-04-27	724	72.40	1020
2003	2023-05-29	2023-06-28	150	15.00	1020
2004	2023-07-02	2023-08-12	168	16.80	1020
2005	2023-09-22	2023-12-29	190	19.00	1020
2006	2023-12-30	2024-01-02	69	6.90	1020
2007	2024-01-03	2024-03-14	154	15.40	1020
2008	2024-03-15	2024-04-20	210	21.00	1020
2009	2023-05-15	2023-07-17	131	13.10	1021
2010	2023-11-19	2023-11-29	130	13.00	1021
2011	2024-02-20	2024-06-01	210	21.00	1021
2012	2023-12-07	2024-01-01	116	11.60	1021
2013	2023-04-12	2023-05-11	111	11.10	1021

Enrolled_t



EnrollmentID	EnrollmentDate	UserID	CourseID	CreatorID	Tax
14001	2024-01-01	1001	3001	5001	NULL
14002	2024-01-02	1001	3002	5002	NULL
14003	2024-01-03	1003	3003	5003	NULL
14004	2024-01-04	1005	3004	5004	NULL
14005	2024-01-05	1005	3004	5005	NULL
14006	2024-01-06	1005	3004	5006	NULL
14007	2024-01-07	1005	3007	5007	NULL
14008	2024-01-08	1005	3008	5008	NULL
14009	2024-01-09	1006	3009	5009	NULL
14010	2024-01-10	1007	3010	5010	NULL
14011	2024-01-11	1008	3011	5011	NULL
14012	2024-01-12	1009	3012	5012	NULL
14013	2024-01-13	1009	3013	5013	NULL

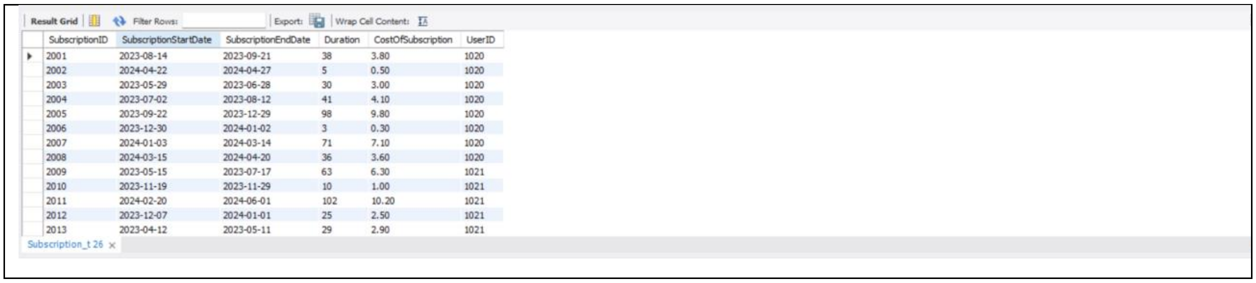
Call Procedure:



```
CALL Effective_Course_Cost_SP(2023,2)
```

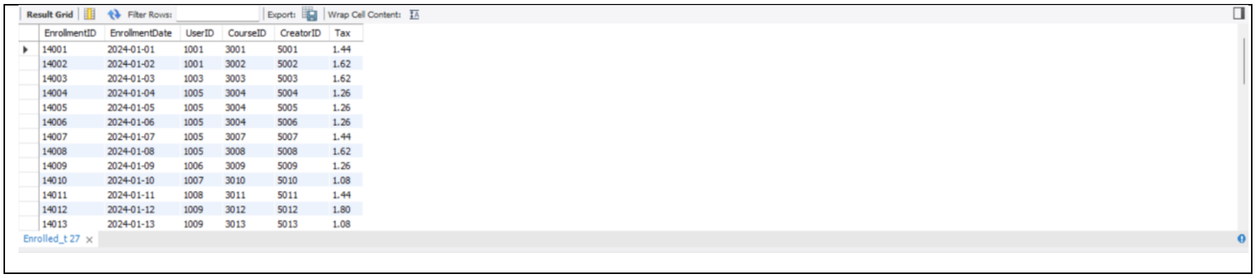
Result after execution:

Subscription_t



SubscriptionID	SubscriptionStartDate	SubscriptionEndDate	Duration	CostOfSubscription	UserID
2001	2023-08-14	2023-09-21	38	3.80	1020
2002	2024-04-22	2024-04-27	5	0.50	1020
2003	2023-05-29	2023-06-28	30	3.00	1020
2004	2023-07-02	2023-08-12	41	4.10	1020
2005	2023-09-22	2023-12-29	98	9.80	1020
2006	2023-12-30	2024-01-02	3	0.30	1020
2007	2024-01-03	2024-03-14	71	7.10	1020
2008	2024-03-15	2024-04-20	36	3.60	1020
2009	2023-05-15	2023-07-17	63	6.30	1021
2010	2023-11-19	2023-11-29	10	1.00	1021
2011	2024-02-20	2024-06-01	102	10.20	1021
2012	2023-12-07	2024-01-01	25	2.50	1021
2013	2023-04-12	2023-05-11	29	2.90	1021

Enrolled_t

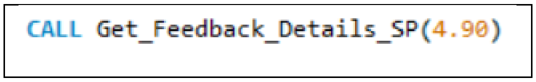


EnrollmentID	EnrollmentDate	UserID	CourseID	CreatorID	Tax
14001	2024-01-01	1001	3001	5001	1.44
14002	2024-01-02	1001	3002	5002	1.62
14003	2024-01-03	1003	3003	5003	1.62
14004	2024-01-04	1005	3004	5004	1.26
14005	2024-01-05	1005	3004	5005	1.26
14006	2024-01-06	1005	3004	5006	1.26
14007	2024-01-07	1005	3007	5007	1.44
14008	2024-01-08	1005	3008	5008	1.62
14009	2024-01-09	1006	3009	5009	1.26
14010	2024-01-10	1007	3010	5010	1.08
14011	2024-01-11	1008	3011	5011	1.44
14012	2024-01-12	1009	3012	5012	1.80
14013	2024-01-13	1009	3013	5013	1.08

3) Procedure Description:

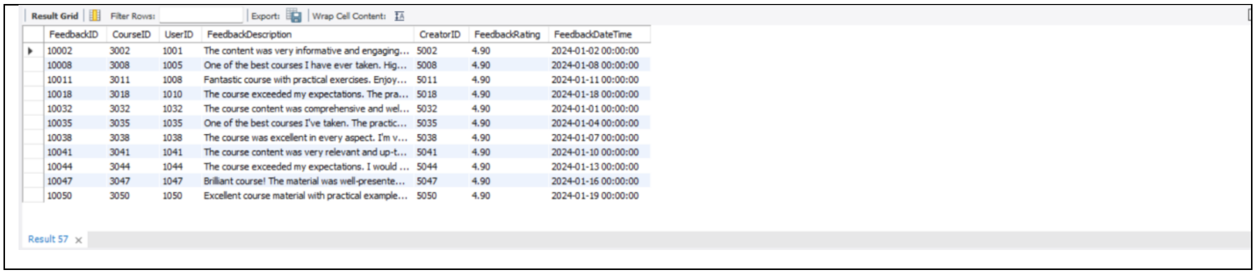
Retrieve pertinent feedback details from the feedback_t table, filtered based on the feedback rating provided as an input parameter.

Call Procedure:



```
CALL Get_Feedback_Details_SP(4.90)
```

Result after execution:

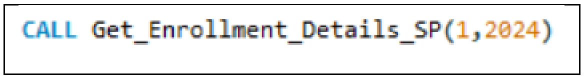


FeedbackID	CourseID	UserID	FeedbackDescription	CreatorID	FeedbackRating	FeedbackDateTime
10002	3002	1001	The content was very informative and engaging...	5002	4.90	2024-01-02 00:00:00
10008	3008	1005	One of the best courses I have ever taken. Hig...	5008	4.90	2024-01-08 00:00:00
10011	3011	1008	Fantastic course with practical exercises. Enjoy...	5011	4.90	2024-01-11 00:00:00
10018	3018	1010	The course exceeded my expectations. The pra...	5018	4.90	2024-01-18 00:00:00
10032	3032	1032	The course content was comprehensive and wel...	5032	4.90	2024-01-01 00:00:00
10035	3035	1035	One of the best courses I've taken. The practic...	5035	4.90	2024-01-04 00:00:00
10038	3038	1038	The course was excellent in every aspect. I'm v...	5038	4.90	2024-01-07 00:00:00
10041	3041	1041	The course content was very relevant and up-t...	5041	4.90	2024-01-10 00:00:00
10044	3044	1044	The course exceeded my expectations. I would ...	5044	4.90	2024-01-13 00:00:00
10047	3047	1047	Brilliant course! The material was well-presente...	5047	4.90	2024-01-16 00:00:00
10050	3050	1050	Excellent course material with practical example...	5050	4.90	2024-01-19 00:00:00

4) Procedure Description:

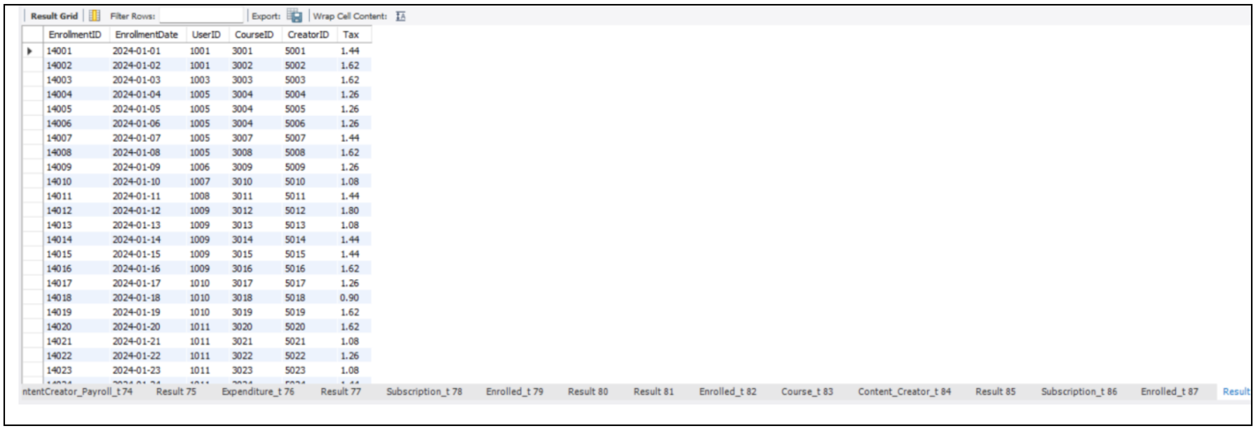
Retrieve relevant enrollment details from the enrolled_t table, filtering them based on the input parameters for month and year.

Call Procedure:



```
CALL Get_Enrollment_Details_SP(1,2024)
```

Result after execution:

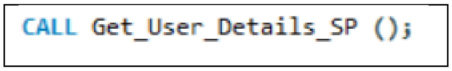


EnrollmentID	EnrollmentDate	UserID	CourseID	CreatorID	Tax
14001	2024-01-01	1001	3001	5001	1.44
14002	2024-01-02	1001	3002	5002	1.62
14003	2024-01-03	1003	3003	5003	1.62
14004	2024-01-04	1005	3004	5004	1.26
14005	2024-01-05	1005	3004	5005	1.26
14006	2024-01-06	1005	3004	5006	1.26
14007	2024-01-07	1005	3007	5007	1.44
14008	2024-01-08	1005	3008	5008	1.62
14009	2024-01-09	1006	3009	5009	1.26
14010	2024-01-10	1007	3010	5010	1.08
14011	2024-01-11	1008	3011	5011	1.44
14012	2024-01-12	1009	3012	5012	1.80
14013	2024-01-13	1009	3013	5013	1.08
14014	2024-01-14	1009	3014	5014	1.44
14015	2024-01-15	1009	3015	5015	1.44
14016	2024-01-16	1009	3016	5016	1.62
14017	2024-01-17	1010	3017	5017	1.26
14018	2024-01-18	1010	3018	5018	0.90
14019	2024-01-19	1010	3019	5019	1.62
14020	2024-01-20	1011	3020	5020	1.62
14021	2024-01-21	1011	3021	5021	1.08
14022	2024-01-22	1011	3022	5022	1.26
14023	2024-01-23	1011	3023	5023	1.08

5) Procedure Description:

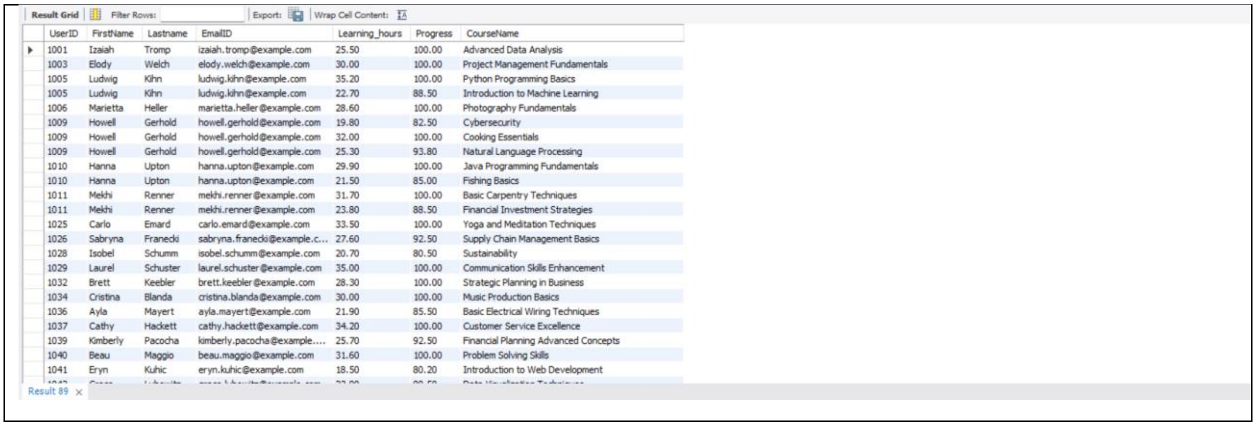
Retrieve user and course details for individuals who have spent over 20 learning hours or have a course progression greater than 80%. This procedure does not require any input parameters.

Call Procedure:



```
CALL Get_User_Details_SP ();
```

Result after execution:



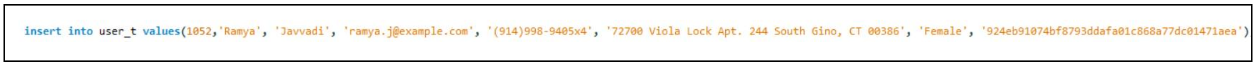
UserID	FirstName	Lastname	EmailID	Learning_hours	Progress	CourseName
1001	Izaiah	Tromp	izaiah.tromp@example.com	25.50	100.00	Advanced Data Analysis
1003	Elody	Welch	elody.welch@example.com	30.00	100.00	Project Management Fundamentals
1005	Ludwig	Kihn	ludwig.kihn@example.com	35.20	100.00	Python Programming Basics
1005	Ludwig	Kihn	ludwig.kihn@example.com	22.70	88.50	Introduction to Machine Learning
1006	Marietta	Heller	marietta.heller@example.com	28.60	100.00	Photography Fundamentals
1009	Howell	Gerhold	howell.gerhold@example.com	19.80	82.50	Cybersecurity
1009	Howell	Gerhold	howell.gerhold@example.com	32.00	100.00	Cooking Essentials
1009	Howell	Gerhold	howell.gerhold@example.com	25.30	93.80	Natural Language Processing
1010	Hanna	Upton	hanna.upton@example.com	29.90	100.00	Java Programming Fundamentals
1010	Hanna	Upton	hanna.upton@example.com	21.50	85.00	Fishing Basics
1011	Mekhi	Renner	mekhi.renner@example.com	31.70	100.00	Basic Carpentry Techniques
1011	Mekhi	Renner	mekhi.renner@example.com	23.80	88.50	Financial Investment Strategies
1025	Carlo	Emard	carlo.emard@example.com	33.50	100.00	Yoga and Meditation Techniques
1026	Sabryna	Franecki	sabryna.franecki@example.c...	27.60	92.50	Supply Chain Management Basics
1028	Isobel	Schumm	isobel.schumm@example.com	20.70	80.50	Sustainability
1029	Laurel	Schuster	laurel.schuster@example.com	35.00	100.00	Communication Skills Enhancement
1032	Brett	Keebler	brett.keebler@example.com	28.30	100.00	Strategic Planning in Business
1034	Cristina	Blanda	cristina.blanda@example.com	30.00	100.00	Music Production Basics
1036	Ayla	Mayert	ayla.mayert@example.com	21.90	85.50	Basic Electrical Wiring Techniques
1037	Cathy	Hackett	cathy.hackett@example.com	34.20	100.00	Customer Service Excellence
1039	Kimberly	Pacocha	kimberly.pacocha@example....	25.70	92.50	Financial Planning Advanced Concepts
1040	Beau	Maggio	beau.maggio@example.com	31.60	100.00	Problem Solving Skills
1041	Eryn	Kuhic	eryn.kuhic@example.com	18.50	80.20	Introduction to Web Development

Triggers:

1) Trigger Description:

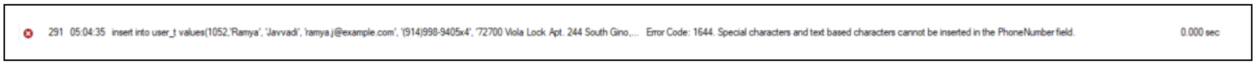
Inform users that inserting special characters or text in the PhoneNumber field of the user_t table is prohibited.

Invoke trigger:



```
insert into user_t values(1052,'Ramya', 'Javvadi', 'ramya.j@example.com', '(914)998-9405x4', '72700 Viola Lock Apt. 244 South Gino, CT 00386', 'Female', '924eb91074bf8793ddafa01c868a77dc01471aea')
```

Result after execution:

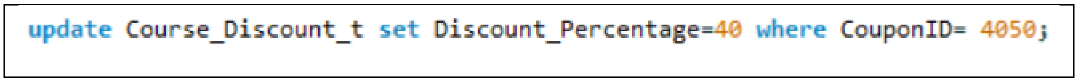


```
291 05:04:35 insert into user_t values(1052,'Ramya', 'Javvadi', 'ramya.j@example.com', '(914)998-9405x4', '72700 Viola Lock Apt. 244 South Gino,...  Error Code: 1644. Special characters and text based characters cannot be inserted in the PhoneNumber field.  0.000 sec
```

2) Trigger Description:

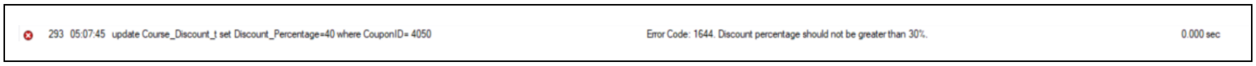
Restrict the discount percentage from exceeding 30% in the course_discount_t table.

Invoke trigger:



```
update Course_Discount_t set Discount_Percentage=40 where CouponID= 4050;
```

Result after execution:

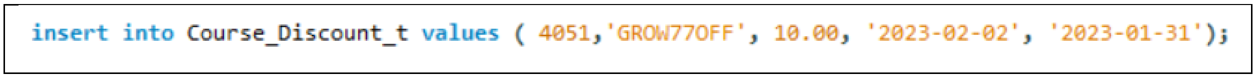


```
293 05:07:45 update Course_Discount_t set Discount_Percentage=40 where CouponID= 4050      Error Code: 1644. Discount percentage should not be greater than 30%.      0.000 sec
```

3) Trigger Description:

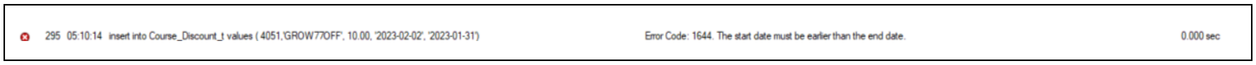
Enforce that the start date is consistently earlier than the end date in the course_discount_t table.

Invoke trigger:



```
insert into Course_Discount_t values ( 4051,'GROW77OFF', 10.00, '2023-02-02', '2023-01-31');
```

Result after execution:

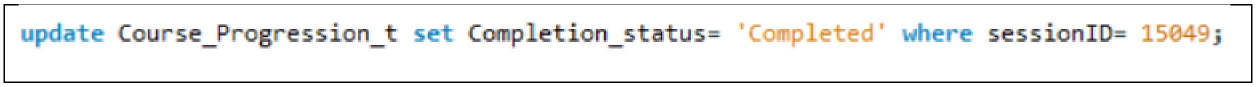


```
295 05:10:14 insert into Course_Discount_t values ( 4051,'GROW77OFF', 10.00, '2023-02-02', '2023-01-31')      Error Code: 1644. The start date must be earlier than the end date.      0.000 sec
```

4) Trigger Description:

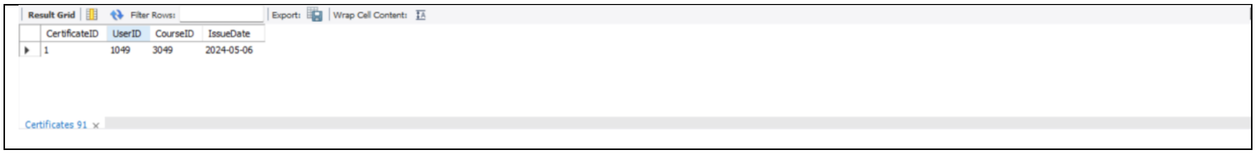
Automatically track course completion status and generate certificates accordingly. Upon completion, a new entry is added to the Certificate_t table to signify the achievement.

Invoke trigger:



```
update Course_Progression_t set Completion_status= 'Completed' where sessionID= 15049;
```

Result after execution (Certificate_t):

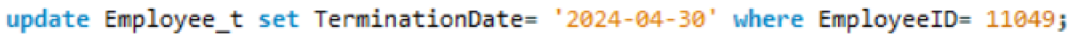


CertificateID	UserID	CourseID	IssueDate
1	1049	3049	2024-05-06

5) Trigger Description:

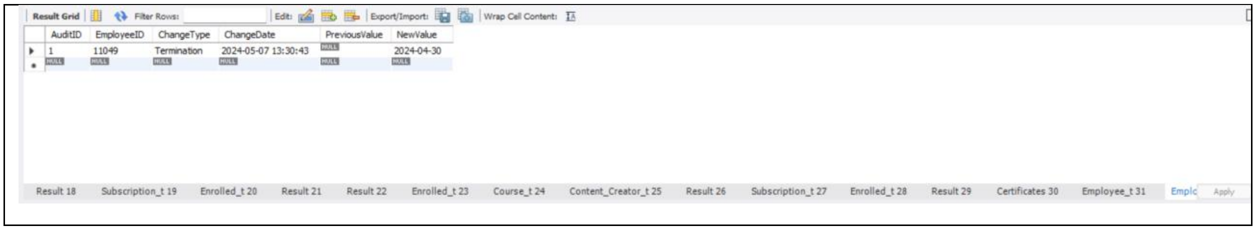
Implement an audit mechanism to record any alterations made in the Employee_t table. All modifications are meticulously logged in the EmployeeAudit_t table for comprehensive tracking.

Invoke trigger:



```
update Employee_t set TerminationDate= '2024-04-30' where EmployeeID= 11049;
```

Result after execution (EmployeeAudit_t):



AuditID	EmployeeID	ChangeType	ChangeDate	PreviousValue	NewValue
1	11049	Termination	2024-05-07 13:30:43	NULL	2024-04-30
NULL	NULL	NULL	NULL	NULL	NULL