

## IS 503 – Assignment 2, due: 20 April 2024

**Part A**

1)

$$\text{CapOv100} \leftarrow \pi \text{ session\_id } (\sigma \text{ capacity} > 100 (\text{Session}))$$
$$\text{AllUsersInSessions} \leftarrow \pi \text{ session\_id } (\text{Registration}) \div \text{CapOv100}$$
$$\text{Result} \leftarrow \pi \text{ fname, lname } ( \text{User} \bowtie \text{user\_id}=\text{user\_id} \text{ AllUsersInSessions} )$$

2)

$$\text{SessAttBy3or6} \leftarrow \pi \text{ session\_id } (\sigma \text{ user\_id} = 3 \vee \text{user\_id} = 6 (\text{Registration}))$$
$$\text{SessAttBy3or6andNot4} \leftarrow \text{SessAttBy3or6} - \pi \text{ session\_id } (\sigma \text{ user\_id} = 4 (\text{Registration}))$$
$$\text{Result} \leftarrow \pi \text{ datetime } (\text{Session} \bowtie \text{session\_id} = \text{session\_id} \text{ SessAttBy3or6andNot4})$$

3)

$$\text{AllAppSess} \leftarrow \pi \text{ session\_id, user\_id } (\sigma \text{ approval\_status} = \text{'approved'} (\text{Registration}))$$
$$\text{UsersWithAppSess} \leftarrow \pi \text{ user\_id } (\text{AllAppSess} \bowtie \text{session\_id} = \text{session\_id} (\text{Registration}))$$
$$\text{UsersWithAtLeastOneReview} \leftarrow \pi \text{ user\_id } (\text{UsersWithAppSess} \cap \text{Review})$$
$$\text{Result} \leftarrow \pi \text{ fname, lname } (\text{UsersWithAtLeastOneReview} \bowtie \text{user\_id} = \text{user\_id} \text{ User})$$

4)

PastSessions  $\leftarrow \sigma \text{ datetime } \geq '2024-01-01' \text{ AND } \text{datetime} \leq '2024-05-01' \text{ AND } \text{session\_status} = 'past'$   
(Session)

SessionRegistrationsCount  $\leftarrow \text{session\_id} \Join \text{COUNT}(\ast) \text{ as } \text{registration\_count} (\text{Registration} \Join \text{session\_id}$   
 $= \text{session\_id PastSessions})$

MaxRegCount  $\leftarrow \pi \text{ session\_id, registration\_count} \Join \text{MAX}(\text{registration\_count}) \text{ as } \text{max\_count}$   
(SessionRegistrationsCount)

MostPopularSession  $\leftarrow \pi \text{ session\_id} (\sigma \text{ registration\_count} = \text{max\_count} (\text{MaxRegCount}))$

5)

SessionsWithRating5  $\leftarrow \pi \text{ session\_id} (\sigma \text{ rating} = 5 (\text{Review}))$

SessionsWithPastStatus  $\leftarrow \pi \text{ session\_id} (\sigma \text{ session\_status} = 'past' (\text{Session}))$

CombinedSessionIDs  $\leftarrow \text{SessionsWithRating5} \cup \text{SessionsWithPastStatus}$

CombinedSessionTitles  $\leftarrow \pi \text{ title} (\text{CombinedSessionIDs} \Join \text{session\_id} = \text{session\_id Session})$

## Part B

1)

SELECT

s.title,

s.datetime,

s.location,

COUNT(r.reg\_id) reg\_id\_count

FROM

SESSION s

JOIN

```

REGISTRATION r ON s.session_id = r.session_id

GROUP BY

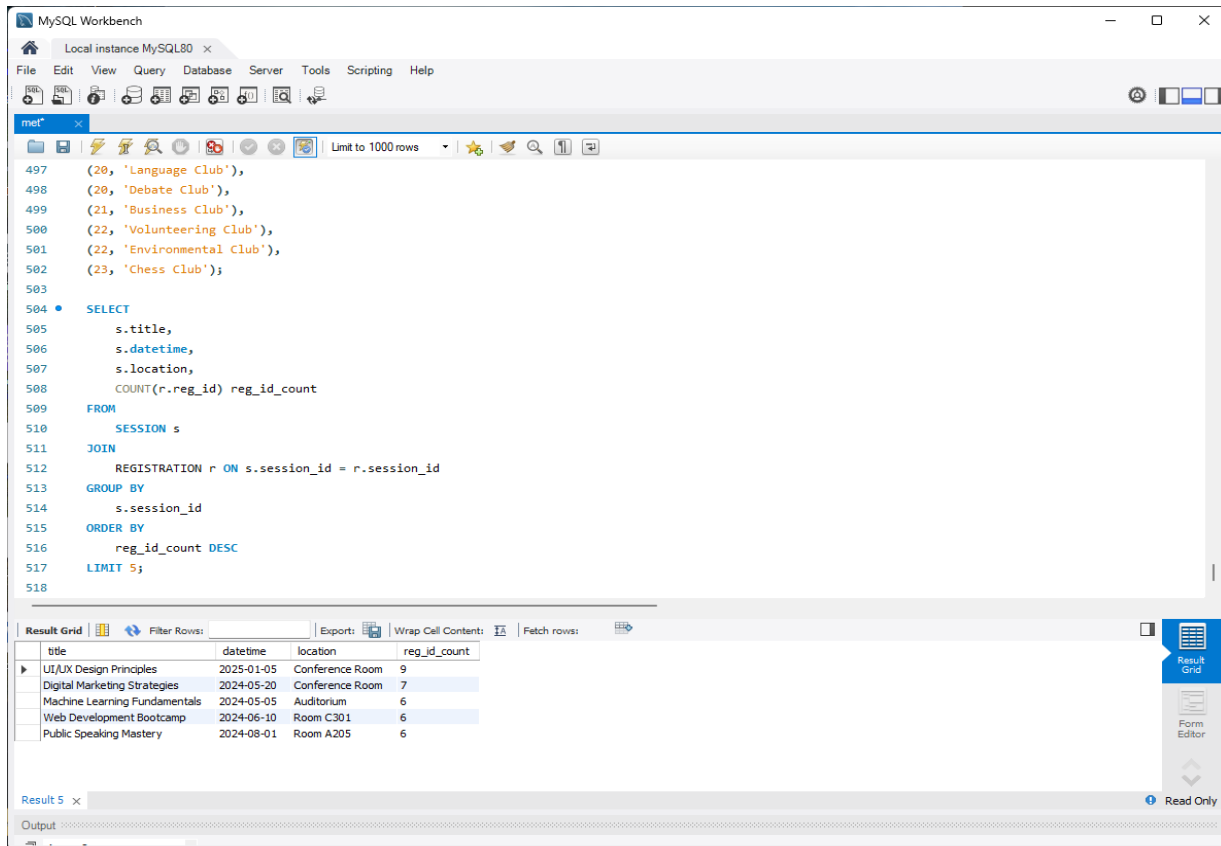
    s.session_id

ORDER BY

    reg_id_count DESC

LIMIT 5;

```



2-a)

```

CREATE OR REPLACE VIEW ActiveUsersWithMultipleRegistrations AS

SELECT

    u.user_id,

    u.phone_number,

    u.email_address,

    u.account_type

FROM

    USER u

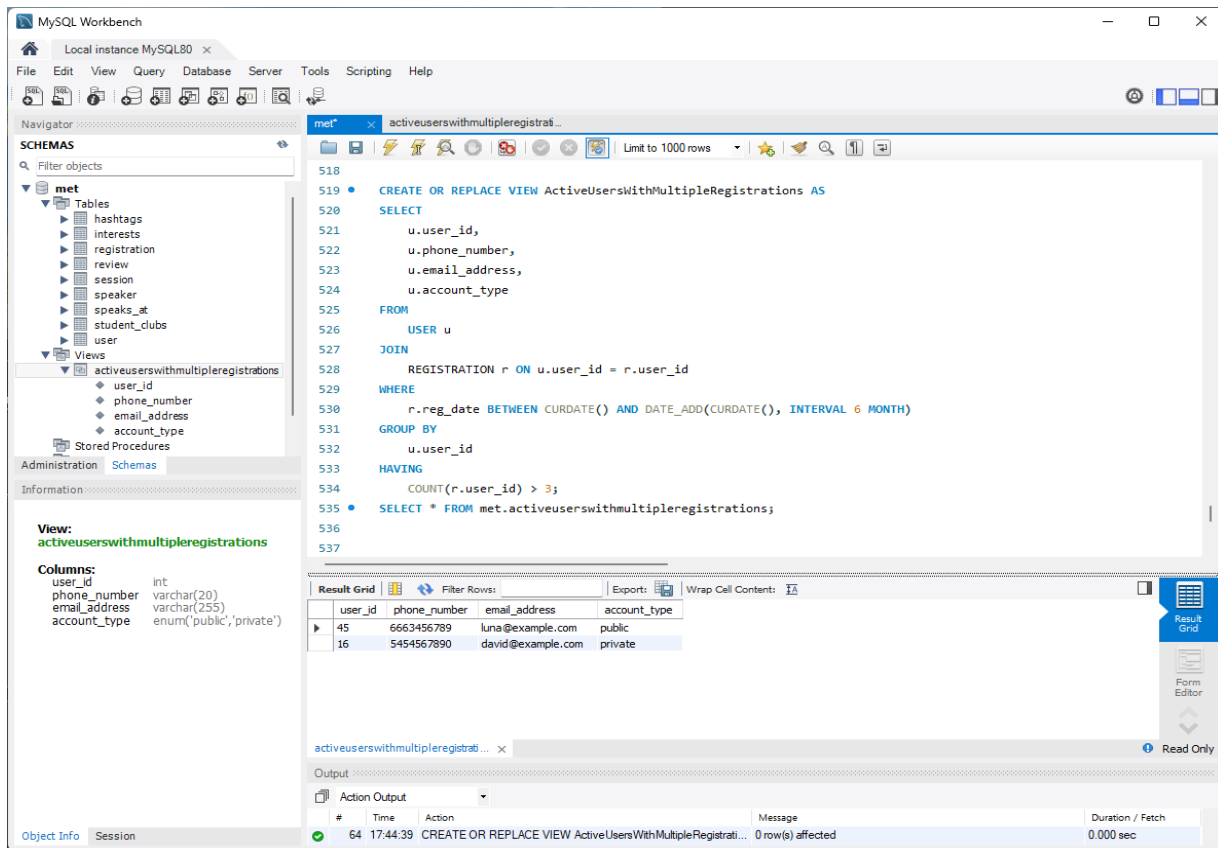
JOIN

```

```

REGISTRATION r ON u.user_id = r.user_id
WHERE
    r.reg_date BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 6 MONTH)
GROUP BY
    u.user_id
HAVING
    COUNT(r.user_id) > 3;
SELECT * FROM met.activeuserswithmultipleregistrations;

```



2-b)

-- Set the delimiter to handle stored procedure creation

DELIMITER //

CREATE TRIGGER PreventSessionOverCapacity

BEFORE INSERT ON REGISTRATION

FOR EACH ROW

BEGIN

```

DECLARE session_capacity INT;
DECLARE registered_users_count INT;

-- Get the capacity of the session
SELECT capacity INTO session_capacity
FROM SESSION
WHERE session_id = NEW.session_id;

-- Get the count of registered users for the session
SELECT COUNT(*) INTO registered_users_count
FROM REGISTRATION
WHERE session_id = NEW.session_id;

-- Check if the session has reached its maximum capacity
IF registered_users_count >= session_capacity THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Session has reached maximum capacity. Registration is not allowed.';
END IF;
END //

-- Reset the delimiter to semicolon
DELIMITER ;

```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

met

Tables

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

No object selected

```

536
537
538 DROP TRIGGER IF EXISTS PreventSessionOverCapacity;
539
540 -- Set the delimiter to handle stored procedure creation
541 DELIMITER //
542
543 CREATE TRIGGER PreventSessionOverCapacity
544 BEFORE INSERT ON REGISTRATION
545 FOR EACH ROW
546 BEGIN
547     DECLARE session_capacity INT;
548     DECLARE registered_users_count INT;
549
550     -- Get the capacity of the session
551     SELECT capacity INTO session_capacity
552     FROM SESSION
553     WHERE session_id = NEW.session_id;
554
555     -- Get the count of registered users for the session
556     SELECT COUNT(*) INTO registered_users_count
557     FROM REGISTRATION
558     WHERE session_id = NEW.session_id;
559
560     -- Check if the session has reached its maximum capacity
561     IF registered_users_count >= session_capacity THEN
562         SIGNAL SQLSTATE '45000'
563         SET MESSAGE_TEXT = 'Session has reached maximum capacity. Registration is not allowed.';
564     END IF;
565 END //
566
567 -- Reset the delimiter to semicolon
568 DELIMITER ;
569
570 -- Attempt to register a user for a session

```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
95	18:40:00	DROP TRIGGER IF EXISTS PreventSessionOverCapacity	0 rows affected	0.015 sec
96	18:40:12	CREATE TRIGGER PreventSessionOverCapacity BEFORE INSERT ON REGISTRATION FOR EACH ROW BEGIN DECLARE session_capacity...	0 rows affected	0.015 sec

Object Info Session

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

met

Tables

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

No object selected

```

570 -- Attempt to register a user for a session
571 INSERT INTO registration (reg_id, requirements, reg_date, approval_status, anonymity, survey_status, user_id, session_id)
572 VALUES
573 (77, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 1, 12),
574 (78, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 2, 12),
575 (79, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 3, 12),
576 (80, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 4, 12),
577 (81, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 5, 12),
578 (82, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 6, 12),
579 (83, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 7, 12),
580 (84, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 8, 12),
581 (85, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 9, 12),
582 (86, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 10, 12),
583 (87, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 11, 12),
584 (88, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 12, 12),
585 (89, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 13, 12),
586 (90, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 14, 12),
587 (91, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 15, 12),
588 (92, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 16, 12),
589 (93, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 17, 12),
590 (94, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 18, 12),
591 (95, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 19, 12),
592 (96, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 20, 12);
593
594 SELECT * FROM registration;
595 DELETE FROM registration
596 WHERE reg_id BETWEEN 77 AND 96;

```

Result Grid

req_id	requirements	req_date	approval_status	anonymity	survey_status	user_id	session_id
72	for students keen on history	2024-05-26	approved	anonymous	done	19	11
73	technical departments only	2024-05-27	rejected	anonymous	in progress	3	18
74	for bookworms only	2024-05-28	approved	not anonymous	done	4	19
75	students with programming skills	2024-05-29	rejected	anonymous	in progress	12	18

registration22 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
97	18:40:48	INSERT INTO registration (reg_id, requirements, reg_date, approval_status, anonymity, survey_status, user_id, session_id) VALUES (77, technical de...	Error Code: 1644. Session has reached maximum capacity. Registration is not allowed.	0.000 sec
98	18:40:56	SP FCT * FROM registration LIMIT 0: 1000	75 rows returned	0.000 sec / 0.000 sec

Object Info Session

3)

SELECT

s.title AS session\_title,

COUNT(r.reg\_id) AS registrations\_count

FROM

SESSION s

JOIN

REGISTRATION r ON s.session\_id = r.session\_id

WHERE

r.approval\_status = 'approved'

GROUP BY

s.session\_id

HAVING

registrations\_count < 3;

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'met' database selected. Below it, the 'Table: session' is highlighted, showing its columns: session\_id (int PK), capacity (int), title (varchar(255)), description (varchar(255)), visual (data), location (varchar(255)), session\_status (enum('past', 'pr')), and session\_type (enum('workshc')). The main editor shows a SQL query with line numbers 594 to 613. The query is a SELECT statement with aliases, a JOIN, a WHERE clause, a GROUP BY clause, and a HAVING clause. The 'Result Grid' at the bottom shows the results of the query, with columns 'session\_title' and 'registrations\_count'. The results are as follows:

session_title	registrations_count
Marketing Analytics Seminar	1
Leadership Workshop	2
Artificial Intelligence Ethics	1
Frontend Development Seminar	1
Financial Planning Seminar	2
Machine Learning Fundamentals	2
Data Visualization Techniques	1
Introduction to Data Science	2
E-commerce Strategies	1

4)

SELECT

s.s\_id,

sp.first\_name,

sp.last\_name,

AVG(r.rating) AS average\_rating

FROM

speaker sp

JOIN

speaks\_at s ON sp.speaker\_id = s.s\_id

JOIN

Session se ON s.session\_id = se.session\_id

LEFT JOIN

Review r ON se.session\_id = r.session\_id

GROUP BY

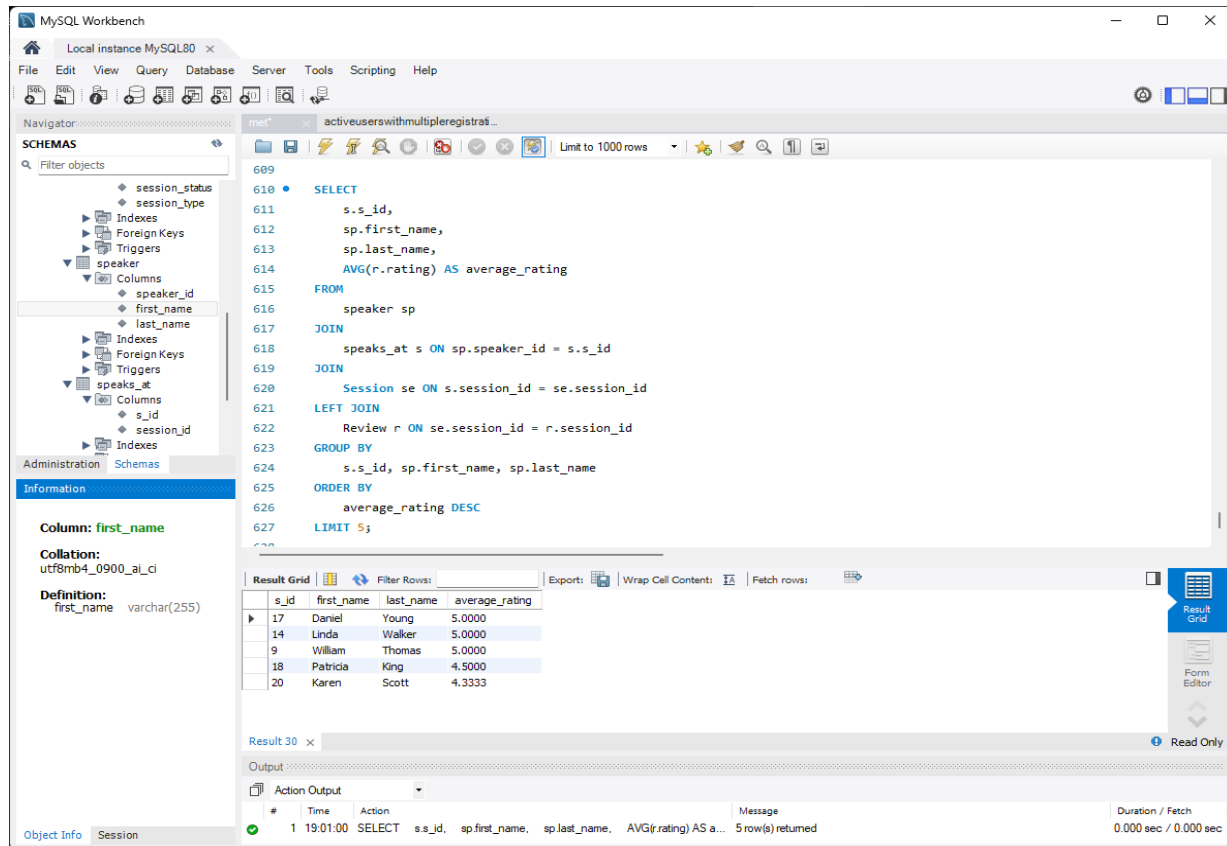
s.s\_id, sp.first\_name, sp.last\_name

ORDER BY

average\_rating DESC

LIMIT 5;





5)

SELECT DISTINCT

u.user\_id,  
u.first\_name,  
u.last\_name

FROM

User u

JOIN

Interests i ON u.user\_id = i.user\_id

JOIN

Registration r ON u.user\_id = r.user\_id

WHERE

(i.interest = 'programming' OR i.interest = 'computer')

AND (r.Requirements LIKE '%programming%' OR r.Requirements LIKE '%computer%' OR  
r.Requirements LIKE '%technic%');

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Limit to 1000 rows

634  
635  
636  
637 • **SELECT DISTINCT**  
638     u.user\_id,  
639     u.first\_name,  
640     u.last\_name  
641 **FROM**  
642     User u  
643 **JOIN**  
644     Interests i ON u.user\_id = i.user\_id  
645 **JOIN**  
646     Registration r ON u.user\_id = r.user\_id  
647 **WHERE**  
648     (i.interest = 'programming' OR i.interest = 'computer')  
649     AND (r.Requirements LIKE '%programming%' OR r.Requirements LIKE '%computer%' OR r.Requirements LIKE '%technic%');

**Table: registration**

**Columns:**

Column	Details
reg_id	int PK
requirements	varchar(255)
reg_date	date
approval_status	enum('approved', 'pending', 'rejected')
anonymity	enum('anonymous', 'known')
survey_status	enum('done', 'progress')
user_id	int
session_id	int

**Result Grid**

	user_id	first_name	last_name
1	John	Smith	
4	Jennifer	Davis	
15	Charlotte	Lewis	

**Result 39 x** Read Only

**Action Output**

#	Time	Action	Message	Duration / Fetch
9	19:07:37	SELECT u.user_id, u.first_name, u.last_name FROM User u J...	4 row(s) returned	0.000 sec / 0.000 sec