Esra Şekerci

2698215

IS 503 – Assignment 2, due: 20 April 2024

Part A

1)

CapOv100 ← π session_id (σ capacity > 100 (Session))

AllUsersInSessions ← π session_id (Registration) ÷ CapOv100

Result ← π fname, lname (User ⋈ user_id=user_id AllUsersInSessions)

2)

SessAttBy3or6 ← π session_id (σ user_id = 3 ∨ user_id = 6 (Registration))

SessAttBy3or6andNot4 ← SessAttBy3or6 – π session_id (σ user_id = 4 (Registration))

Result ← π datetime (Session ⋈ session_id = session_id SessAttBy3or6andNot4)

3)

AllAppSess ← π session_id, user_id (σ approval_status = 'approved' (Registration))

UsersWithAppSess ← π user_id (AllAppSess ⋈ session_id= session_id (Registration))

UsersWithAtLeastOneReview ← π user_id (UsersWithAppSess ∩ Review)

Result ← π fname, lname (UsersWithAtLeastOneReview ⋈ user_id = user_id User)

4)

PastSessions ← σ datetime >= '2024-01-01' AND datetime <= '2024-05-01' AND session_status = 'past' (Session)

SessionRegistrationsCount ← session_id $\mathscr{F}$ COUNT(*) as registration_count (Registration ⋈ session_id = session_id PastSessions)

MaxRegCount ← π session_id, registration_count $\mathscr{F}$ MAX(registration_count) as max_count (SessionRegistrationsCount)

MostPopularSession ← π session_id (σ registration_count = max_count (MaxRegCount))

5)

SessionsWithRating5 ← π session_id (σ rating = 5 (Review))

SessionsWithPastStatus ← π session_id (σ session_status = 'past' (Session))

CombinedSessionIDs ← SessionsWithRating5 ∪ SessionsWithPastStatus

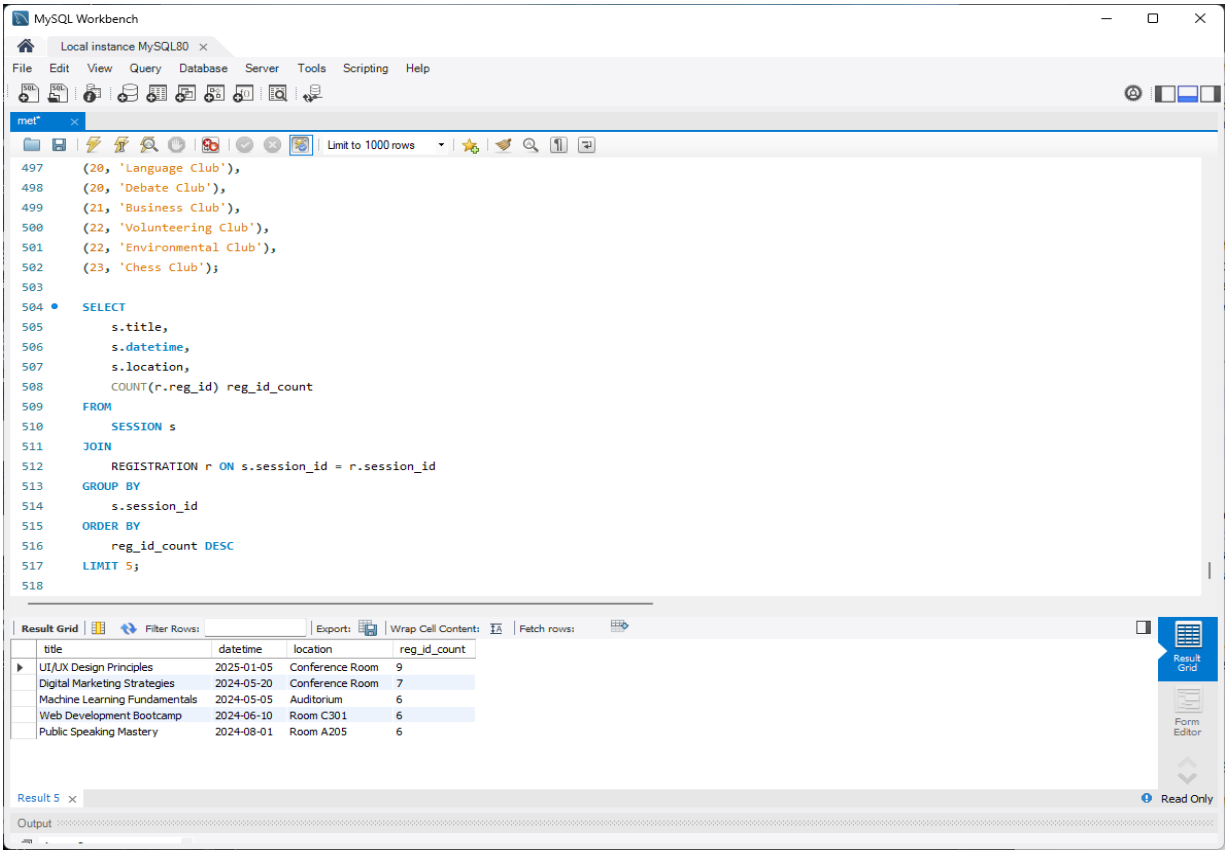
CombinedSessionTitles ← π title (CombinedSessionIDs ⋈ session_id = session_id Session)

Part B

1)

```
SELECT
   s.title,
   s.datetime,
   s.location,
   COUNT(r.reg_id) reg_id_count
FROM
   SESSION s
JOIN
```

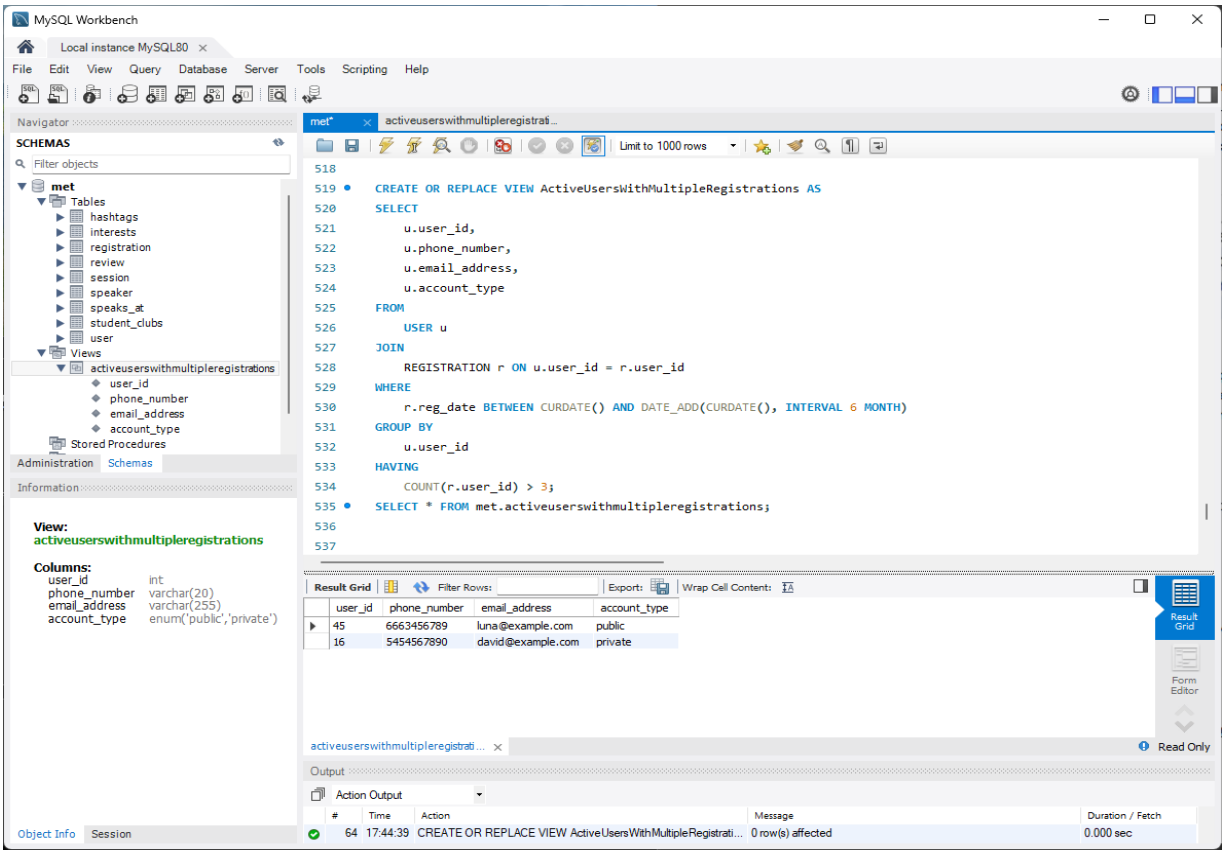
```
    REGISTRATION r ON s.session_id = r.session_id
GROUP BY
    s.session_id
ORDER BY
    reg_id_count DESC
LIMIT 5;
```



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
```

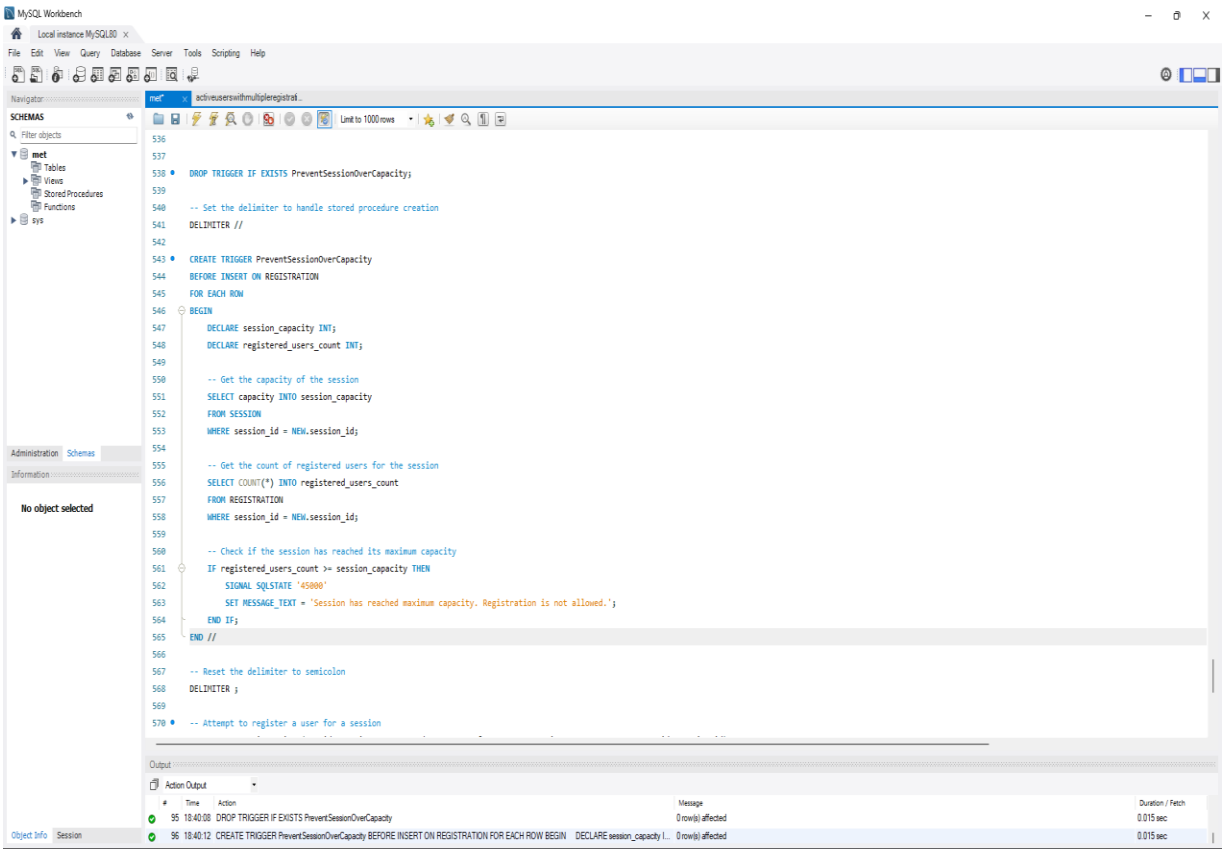
```
    REGISTRATION r ON u.user_id = r.user_id
WHERE
    r.reg_date BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 6 MONTH)
GROUP BY
    u.user_id
HAVING
    COUNT(r.user_id) > 3;
SELECT * FROM met.activeuserswithmultipleregistrations;
```



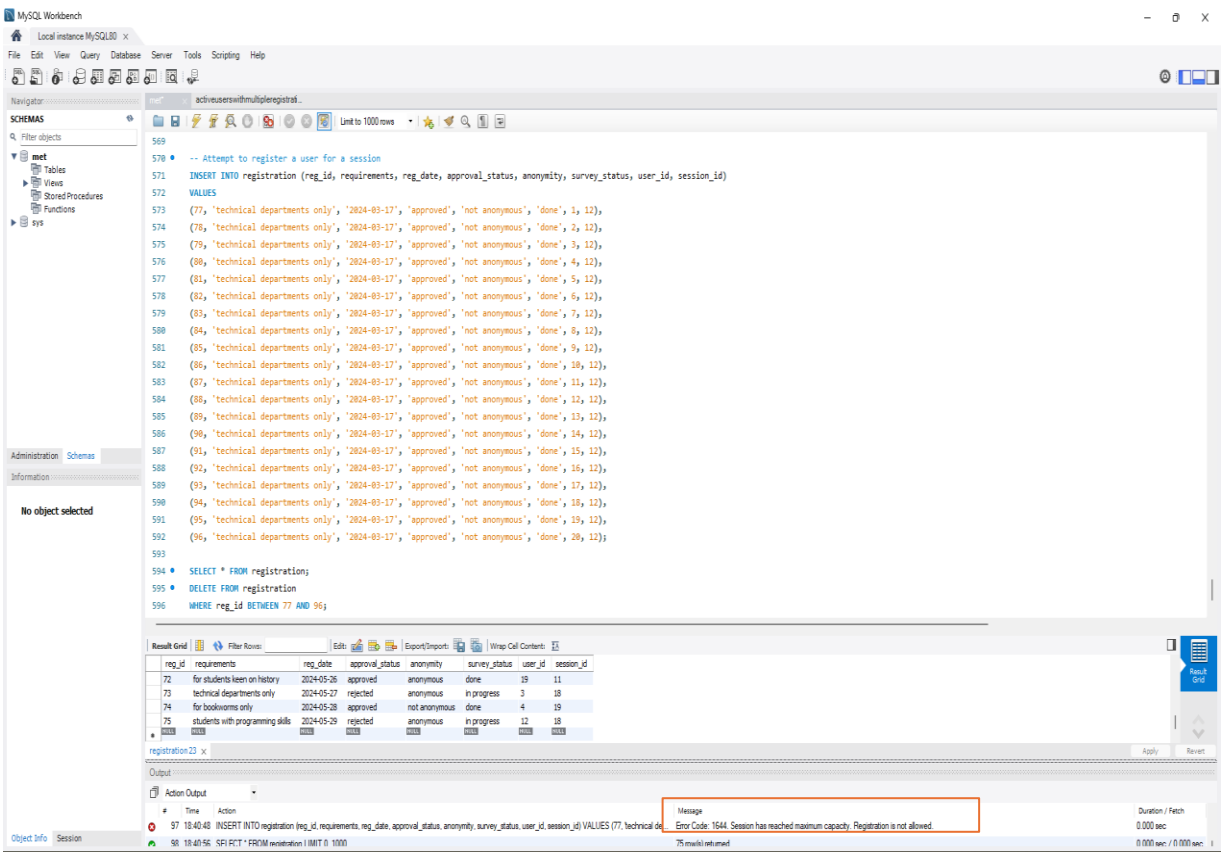
2-b)

```
-- 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 ;
```



```
DROP TRIGGER IF EXISTS PreventSessionOverCapacity;

-- 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 ;

-- Attempt to register a user for a session
```



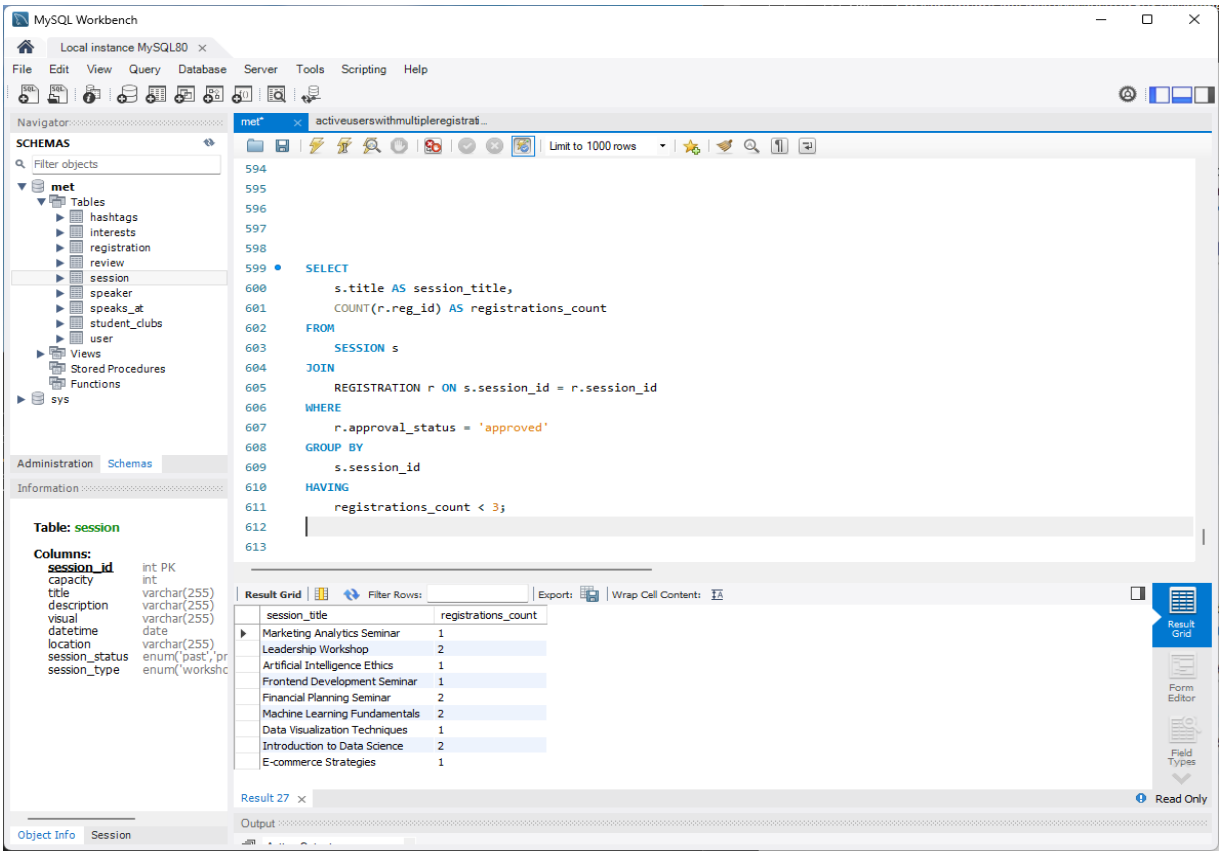
```
-- Attempt to register a user for a session
INSERT INTO registration (reg_id, requirements, reg_date, approval_status, anonymity, survey_status, user_id, session_id)
VALUES
(77, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 1, 12),
(78, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 2, 12),
(79, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 3, 12),
(80, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 4, 12),
(81, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 5, 12),
(82, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 6, 12),
(83, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 7, 12),
(84, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 8, 12),
(85, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 9, 12),
(86, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 10, 12),
(87, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 11, 12),
(88, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 12, 12),
(89, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 13, 12),
(90, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 14, 12),
(91, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 15, 12),
(92, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 16, 12),
(93, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 17, 12),
(94, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 18, 12),
(95, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 19, 12),
(96, 'technical departments only', '2024-03-17', 'approved', 'not anonymous', 'done', 20, 12);

SELECT * FROM registration;
DELETE FROM registration
WHERE reg_id BETWEEN 77 AND 96;
```

reg_id	requirements	reg_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

```
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  SELECT * FROM registration LIMIT 0, 1000  75 row(s) returned  0.000 sec / 0.000 sec
```

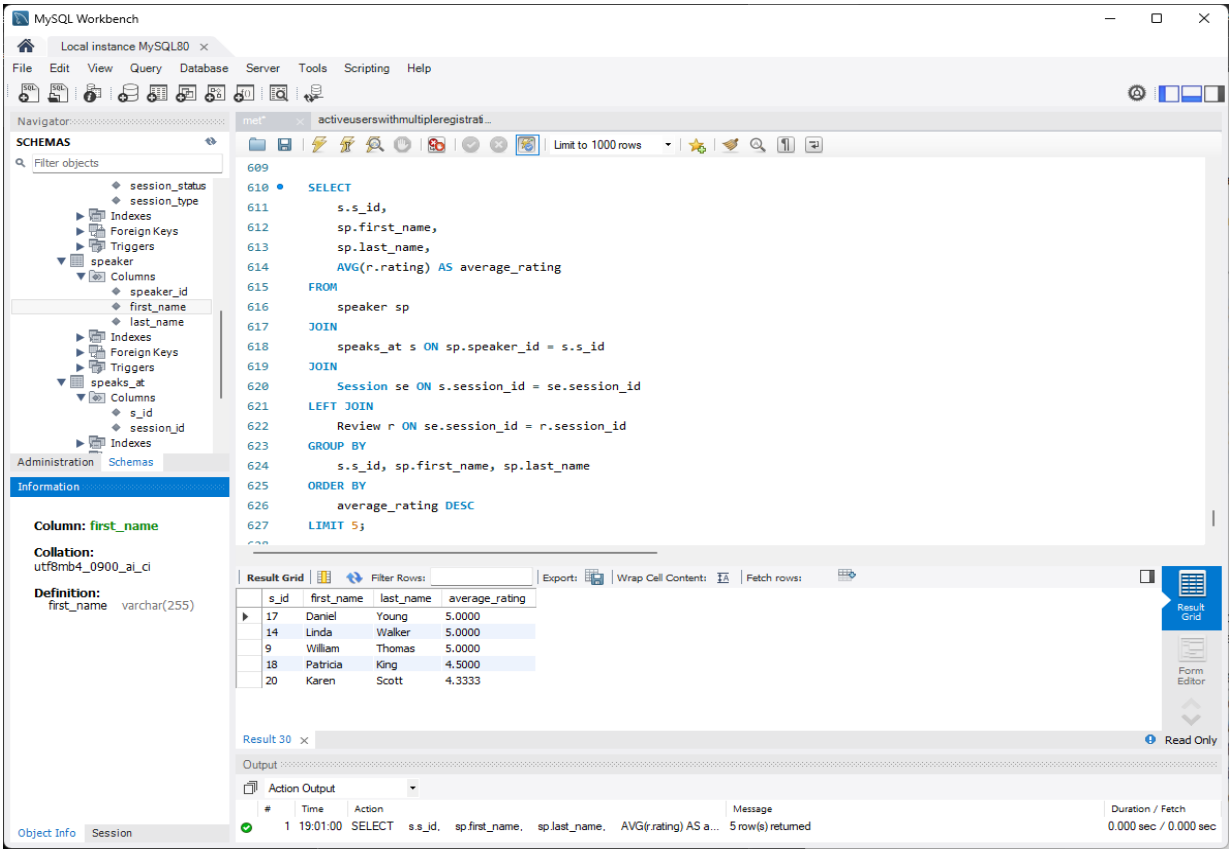
3)

```
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;
```



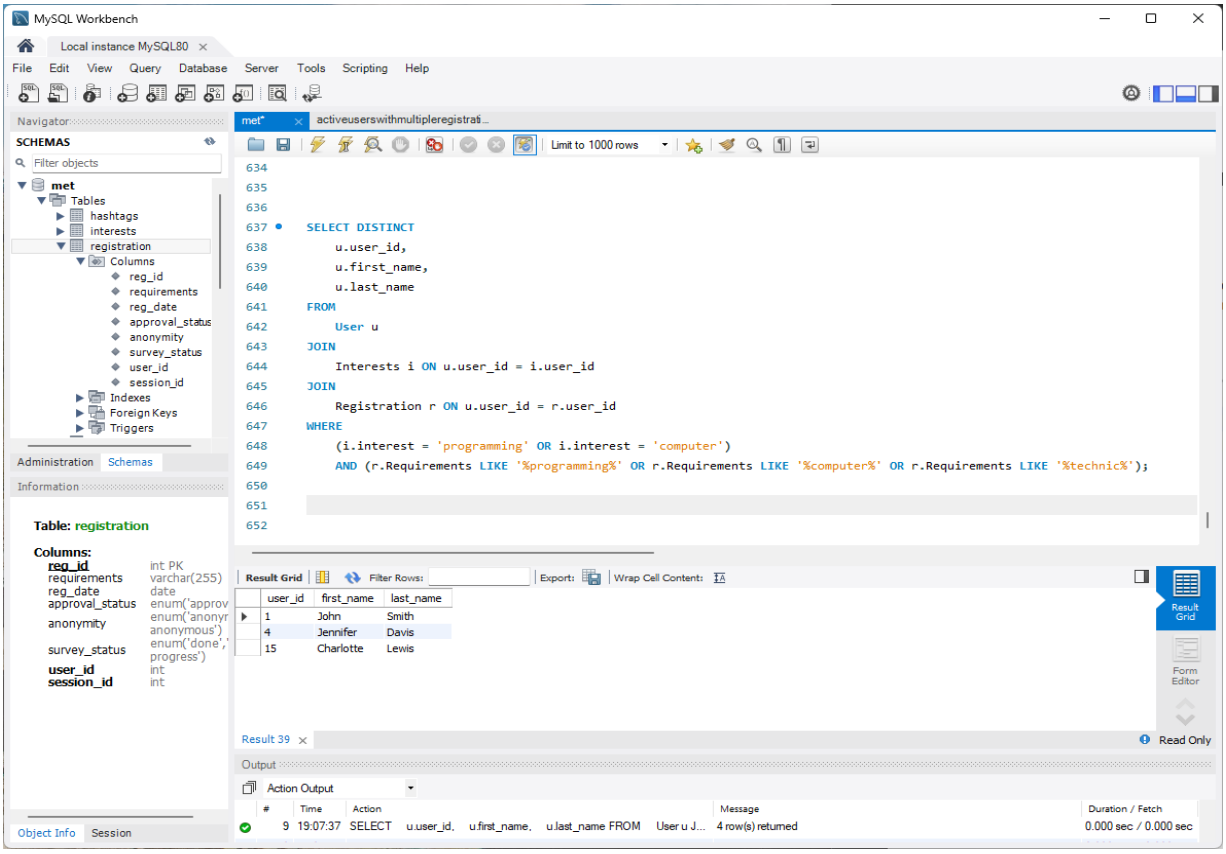
4)

```
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
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
met
Tables
hashtags
interests
registration
Columns
reg_id
requirements
reg_date
approval_status
anonymity
survey_status
user_id
session_id
Indexes
Foreign Keys
Triggers
Administration Schemas
Information
Table: registration
Columns:
reg_id int PK
requirements varchar(255)
reg_date date
approval_status enum('approv
anonymity enum('anonyr anonymous')
survey_status enum('done', progress')
user_id int
session_id int
Object Info Session
met*
activeuserswithmultipleregistrati...
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%');
650
651
652
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
user_id first_name last_name
1 John Smith
4 Jennifer Davis
15 Charlotte Lewis
Result Grid
Form Editor
Result 39
Read Only
Output
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