BACKEND DEVELOPER TECHNICAL ASSESSMENT

Total Time: 45-60 minutes

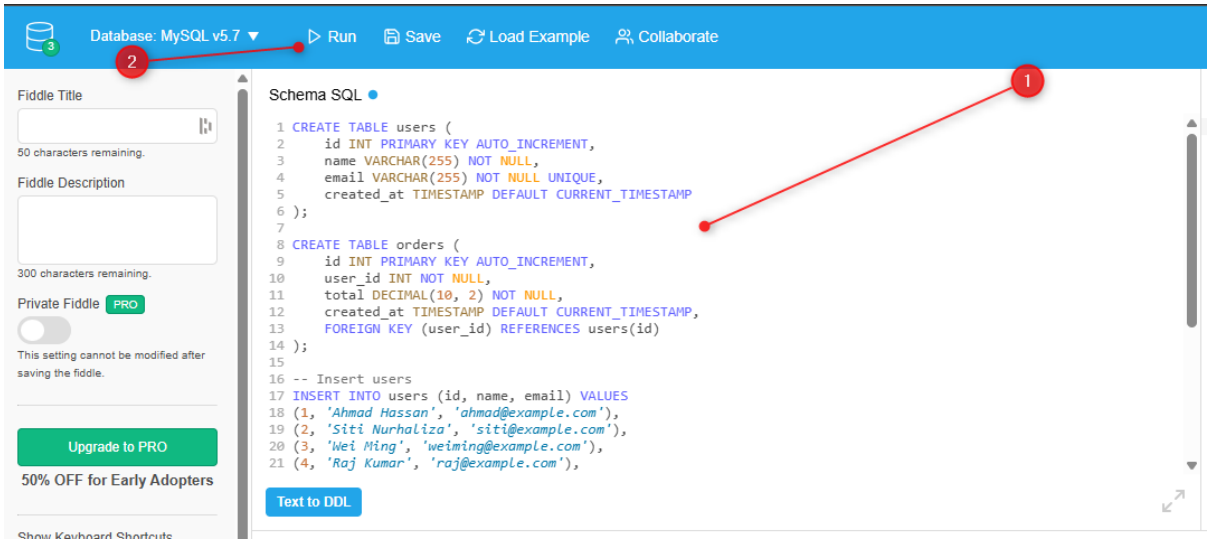
TASK 1: SQL QUERY CHALLENGE (15 minutes)

Background

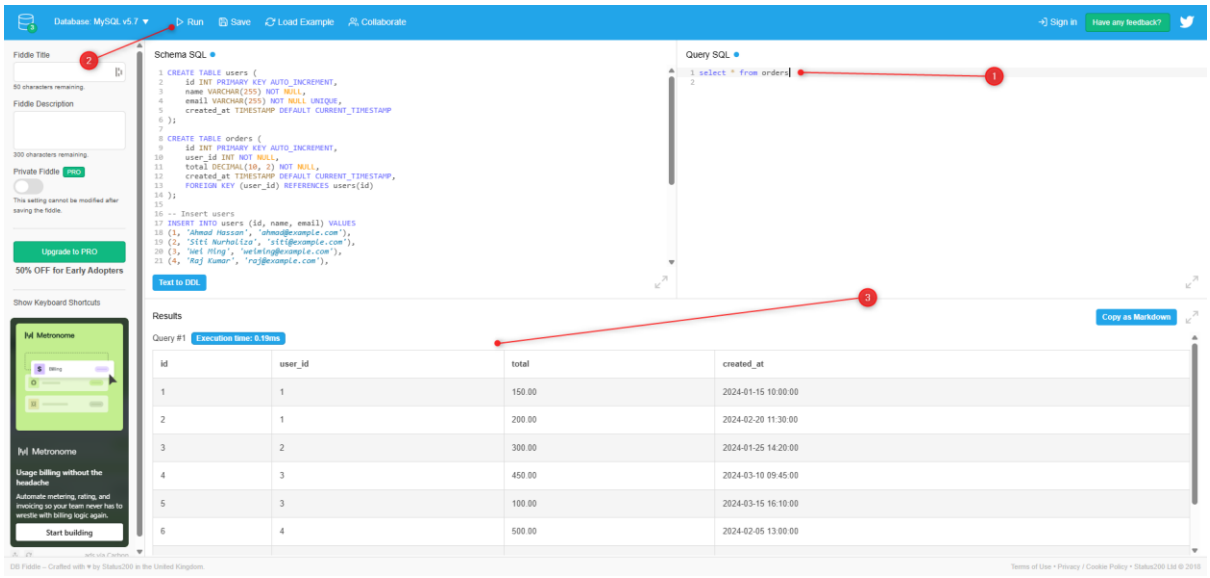
You're working on PetBacker's analytics dashboard and need to write SQL queries to analyze user order data for business reporting. You can test your queries using any of these free online platforms: https://www.db-fiddle.com/

Seed Data (Copy and run this first):

```
-- Create tables

CREATE TABLE users ( id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL UNIQUE, created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP );

CREATE TABLE orders ( id INT PRIMARY KEY AUTO_INCREMENT, user_id INT NOT NULL, total DECIMAL(10, 2) NOT NULL, created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY (user_id) REFERENCES users(id) );

-- Insert users

INSERT INTO users (id, name, email) VALUES (1, 'Ahmad Hassan', 'ahmad@example.com'), (2, 'Siti Nurhaliza', 'siti@example.com'), (3, 'Wei Ming', 'weiming@example.com'), (4, 'Raj Kumar', 'raj@example.com'), (5, 'Lisa Tan', 'lisa@example.com'), (6, 'Farah Aini', 'farah@example.com'), (7, 'David Lim', 'david@example.com');

-- Insert orders

INSERT INTO orders (id, user_id, total, created_at) VALUES (1, 1, 150.00, '2024-01-15 10:00:00'), (2, 1, 200.00, '2024-02-20 11:30:00'), (3, 2, 300.00, '2024-01-25 14:20:00'), (4, 3, 450.00, '2024-03-10 09:45:00'), (5, 3, 100.00, '2024-03-15 16:10:00'), (6, 4, 500.00, '2024-02-05 13:00:00'), (7, 5, 250.00, '2024-01-30 12:15:00'), (8, 1, 175.00, '2024-03-20 15:30:00');
```



Verify the inserted data:



Your Tasks

QUERY 1: Top 5 Users by Total Order Amount

Find the top 5 users who have spent the most money (by total order amount).

Expected Output Columns:

- user_id
- name
- email
- total_spent (sum of all their orders)

Sort by: Total spent in descending order (highest spender first)

QUERY 2: Users Who Haven't Placed Any Orders

Find all users who exist in the database but have never placed an order.

Expected Output Columns:

- user_id
- name
- email

Sort by: User ID in ascending order

TASK 2: PHP MATCHING ALGORITHM (30-45 minutes)

Background

You're building a feature for PetBacker's platform that matches pet owners' service requests with available service providers. Your task is to create a matching algorithm that finds suitable providers for each request.

Your Task

Write a PHP function that takes two arrays (user requests and service listings) and returns the best matches for each request.

Function Signature:

```
<?php
function matchRequestsToListings($userRequests, $serviceListings) {
    // Your code here
    // Return format:
    // ["REQ001" => [matched listing_ids], "REQ002" => [matched
listing_ids]]
```

```
}
?>
```

Input Data

User Requests Array:

```
<?php
$userRequests = [
    [
        "request_id" => "REQ001",
        "user_id" => "USER_101",
        "service_name" => "dog_walking",
        "coordinate" => ["lat" => 3.1390, "lng" => 101.6869],
        "pet_type" => "dog",
        "pet_weight" => "8kg",
        "pet_details" => [
            "breed" => "Golden Retriever",
            "is_aggressive" => false,
            "special_notes" => "Friendly, loves treats",
            "booking_type" => "scheduled"
        ]
    ],
    [
        "request_id" => "REQ002",
        "user_id" => "USER_102",
        "service_name" => "pet_sitting",
        "coordinate" => ["lat" => 3.1520, "lng" => 101.7123],
        "pet_type" => "dog",
        "pet_weight" => "22kg",
        "pet_details" => [
            "breed" => "German Shepherd",
            "is_aggressive" => true,
            "special_notes" => "Needs experienced handler",
            "booking_type" => "last_minute"
        ]
    ]
];
?>
```

Service Listings Array:

```
```

```
<?php
$serviceListings = [
    [
        "listing_id" => "LIST001",
        "user_id" => "PROVIDER_201",
        "service_name" => "dog_walking",
        "accepted_pet" => ["dog", "cat"],
        "coordinate" => ["lat" => 3.1405, "lng" => 101.6895],
        "distance_willing_to_travel" => 3,
        "pet_weight" => ["1-5kg", "5-10kg"],
        "profiling" => [
            "can_handle_aggressive" => false,
            "accept_last_minute_booking" => true,
            "experience_years" => 2
        ]
    ],
    [
        "listing_id" => "LIST002",
        "user_id" => "PROVIDER_202",
        "service_name" => "pet_sitting",
        "accepted_pet" => ["dog", "cat", "rabbit"],
        "coordinate" => ["lat" => 3.1580, "lng" => 101.7150],
        "distance_willing_to_travel" => 5,
        "pet_weight" => ["10-15kg", "15-25kg", "25kg+"],
        "profiling" => [
            "can_handle_aggressive" => true,
            "accept_last_minute_booking" => false,
            "experience_years" => 5
        ]
    ],
    [
        "listing_id" => "LIST003",
        "user_id" => "PROVIDER_203",
        "service_name" => "dog_walking",
        "accepted_pet" => ["dog"],
        "coordinate" => ["lat" => 3.1500, "lng" => 101.7000],
        "distance_willing_to_travel" => 2,
        "pet_weight" => ["1-5kg", "5-10kg", "10-15kg"],
```

```
    "profiling" => [
        "can_handle_aggressive" => false,
        "accept_last_minute_booking" => true,
        "experience_years" => 1
    ]
],
[
    "listing_id" => "LIST004",
    "user_id" => "PROVIDER_201",
    "service_name" => "pet_sitting",
    "accepted_pet" => ["dog", "cat"],
    "coordinate" => ["lat" => 3.1405, "lng" => 101.6895],
    "distance_willing_to_travel" => 3,
    "pet_weight" => ["1-5kg", "5-10kg"],
    "profiling" => [
        "can_handle_aggressive" => false,
        "accept_last_minute_booking" => false,
        "experience_years" => 2
    ]
],
[
    "listing_id" => "LIST005",
    "user_id" => "PROVIDER_204",
    "service_name" => "pet_sitting",
    "accepted_pet" => ["dog", "cat"],
    "coordinate" => ["lat" => 3.1510, "lng" => 101.7100],
    "distance_willing_to_travel" => 4,
    "pet_weight" => ["15-25kg", "25kg+"],
    "profiling" => [
        "can_handle_aggressive" => true,
        "accept_last_minute_booking" => true,
        "experience_years" => 7
    ]
],
[
    "listing_id" => "LIST006",
    "user_id" => "PROVIDER_205",
    "service_name" => "dog_walking",
    "accepted_pet" => ["dog"],
    "coordinate" => ["lat" => 3.1420, "lng" => 101.6900],
    "distance_willing_to_travel" => 5,
    "pet_weight" => ["5-10kg", "10-15kg"],
    "profiling" => [
        "can_handle_aggressive" => false,
        "accept_last_minute_booking" => true,
        "experience_years" => 3
    ]
],
[
```

```
    "listing_id" => "LIST007",
    "user_id" => "PROVIDER_206",
    "service_name" => "grooming",
    "accepted_pet" => ["dog", "cat"],
    "coordinate" => ["lat" => 3.1400, "lng" => 101.6880],
    "distance_willing_to_travel" => 10,
    "pet_weight" => ["1-5kg", "5-10kg", "10-15kg"],
    "profiling" => [
        "can_handle_aggressive" => true,
        "accept_last_minute_booking" => true,
        "experience_years" => 4
    ]
],
[
    "listing_id" => "LIST008",
    "user_id" => "PROVIDER_207",
    "service_name" => "pet_sitting",
    "accepted_pet" => ["cat", "rabbit"],
    "coordinate" => ["lat" => 3.1550, "lng" => 101.7130],
    "distance_willing_to_travel" => 5,
    "pet_weight" => ["1-5kg", "5-10kg"],
    "profiling" => [
        "can_handle_aggressive" => false,
        "accept_last_minute_booking" => true,
        "experience_years" => 3
    ]
],
[
    "listing_id" => "LIST009",
    "user_id" => "PROVIDER_201",
    "service_name" => "dog_walking",
    "accepted_pet" => ["dog"],
    "coordinate" => ["lat" => 3.1405, "lng" => 101.6895],
    "distance_willing_to_travel" => 3,
    "pet_weight" => ["10-15kg", "15-25kg"],
    "profiling" => [
        "can_handle_aggressive" => false,
        "accept_last_minute_booking" => true,
        "experience_years" => 2
    ]
],
[
    "listing_id" => "LIST010",
    "user_id" => "PROVIDER_208",
    "service_name" => "pet_sitting",
    "accepted_pet" => ["dog"],
    "coordinate" => ["lat" => 3.1530, "lng" => 101.7110],
    "distance_willing_to_travel" => 6,
    "pet_weight" => ["15-25kg"],
```

```
        "profiling" => [
            "can_handle_aggressive" => false,
            "accept_last_minute_booking" => true,
            "experience_years" => 2
        ]
    ]
];
?>
```

Requirements

Your matching algorithm should consider:

1. Service Type The listing must offer the same service requested
2. Pet Type The provider must accept the pet type in their accepted_pet array
3. Pet Weight The pet's weight must fall within one of the provider's weight ranges
4. Distance Calculate the distance between request and listing coordinates, ensure its within the provider's distance_willing_to_travel (in kilometers)
5. Aggressive Handling If the pet is aggressive, the provider must have can_handle_aggressive set to true
6. Booking Type If it's a last_minute booking, the provider must have accept_last_minute_booking set to true

Distance Calculation Helper

You can use the Haversine formula or this simplified approximation:

```
<?php
function calculateDistance($lat1, $lng1, $lat2, $lng2) {
    $earthRadius = 6371; // km
    $dLat = deg2rad($lat2 - $lat1);
    $dLng = deg2rad($lng2 - $lng1);
    $a = sin($dLat/2) * sin($dLat/2) +
         cos(deg2rad($lat1)) * cos(deg2rad($lat2)) *
         sin($dLng/2) * sin($dLng/2);
    $c = 2 * atan2(sqrt($a), sqrt(1-$a));
    return $earthRadius * $c;
}
?>
```

Notes

- You can use any built-in PHP functions
- Consider edge cases and data validation
- Write clean, readable code
- You may create helper functions if needed

Deliverables for Task 2

1. The complete PHP function with your matching logic
2. Brief explanation of your approach (2-3 sentences)
3. Any assumptions you made

SUBMISSION INSTRUCTIONS

Please create separate files for each task and submit as a ZIP file:

File Structure

Create the following files:

1. query1.sql
   - Contains your SQL query for Task 1 - Query 1 (Top 5 users)
   - Include a comment with your explanation
2. query2.sql
   - Contains your SQL query for Task 1 - Query 2 (Users with no orders) -
   - Include a comment with your explanation
3. matching_algorithm.php
   - Contains your complete PHP matching function - Include comments explaining your approach
   - Include any assumptions as comments at the top

File Content Format

query1.sql

example:

```
/* TASK 1 - QUERY 1: Top 5 Users by Total Order Amount

Explanation: [Your explanation here - 1-2 sentences]

*/ [Your SQL query here]
```

query2.sql

example:

```
/* TASK 1 - QUERY 2: Users Who Haven't Placed Any Orders

Explanation: [Your explanation here - 1-2 sentences]

*/[Your SQL query here]
```

matching_algorithm.php

example:

```
<?php

/* TASK 2: Pet Service Matching Algorithm

Approach: [Your approach explanation - 2-3 sentences]

Assumptions: [List any assumptions you made] */

function matchRequestsToListings($userRequests, $serviceListings) {

// Your code here

}

// Helper functions (if any)

?>
```

Final Submission

1. Create a folder named: [YourName]_BackendAssessment
    a. Example: AhmadHassan_BackendAssessment
2. Place all 3 files in the folder: - query1.sql - query2.sql - matching_algorithm.php
3. Compress the folder as ZIP
4. Submit the ZIP file via [specify your submission method: email/form/etc]

File Naming Convention

Final ZIP file name: [YourName]_BackendAssessment.zip Example: AhmadHassan_BackendAssessment.zip

---

Good luck! Feel free to ask clarifying questions if anything is unclear.