

OLIN Assessment POSTGRESQL

SOAL 1

Schema Table:

1. CREATE TABLE users (id INT PRIMARY KEY, name VARCHAR(255), email VARCHAR(255));
2. CREATE TABLE orders (id INT PRIMARY KEY, user_id INT, amount DECIMAL(10,2), created_at TIMESTAMP);
3. CREATE TABLE order_items (id INT PRIMARY KEY, order_id INT, product_name VARCHAR(255), price DECIMAL(10,2), quantity INT);
4. INSERT INTO users (id, name, email) VALUES (1, 'John Doe', 'johndoe@example.com'), (2, 'Jane Smith', 'janesmith@example.com'), (3, 'Bob Johnson', 'bobjohnson@example.com');
5. INSERT INTO orders (id, user_id, amount, created_at) VALUES (1, 1, 100.00, '2022-01-02 10:30:00'), (2, 2, 50.00, '2022-01-03 09:00:00'), (3, 1, 150.00, '2022-01-04 14:15:00'), (4, 3, 200.00, '2022-01-05 17:45:00'), (5, 2, 75.00, '2022-01-06 11:20:00');
6. INSERT INTO order_items (id, order_id, product_name, price, quantity) VALUES (1, 1, 'T-Shirt', 25.00, 2), (2, 1, 'Jeans', 50.00, 1), (3, 2, 'Socks', 10.00, 5), (4, 3, 'Shoes', 75.00, 2), (5, 4, 'Jacket', 100.00, 1), (6, 5, 'Sweater', 25.00, 3);

Dengan Schema Table diatas, buatlah 1 query yang paling optimal untuk menampilkan pengguna yang menghabiskan uang minimal 1000 pada transaksi yang dilakukan pada atau setelah 1 Januari 2022 dan kelompokkan hasilnya berdasarkan nama pengguna.

SOAL 2

Schema Table:

1. CREATE DATABASE first;
2. CREATE TABLE users (id SERIAL PRIMARY KEY, name VARCHAR(255), email VARCHAR(255));
3. INSERT INTO users (name, email) VALUES ('John Doe', 'john.doe@example.com'), ('Jane Smith', 'jane.smith@example.com');
4. CREATE DATABASE second;
5. CREATE TABLE orders (id SERIAL PRIMARY KEY, user_id INT, amount NUMERIC(10,2), created_at TIMESTAMP);
6. INSERT INTO orders (user_id, amount, created_at) VALUES (1, 100.00, now()), (2, 200.00, now()), (1, 50.00, now());

Dengan Schema Table diatas, buatlah 1 query yang paling optimal untuk menampilkan nama, amount dan created_at per user.

OLIN Assessment Golang

SOAL 1

Diberikan sebuah array bilangan bulat `nums` dan sebuah bilangan bulat `target`, kembalikan indeks dari dua bilangan tersebut sedemikian rupa sehingga jumlahnya sama dengan `target`.

Anda dapat berasumsi bahwa setiap input akan memiliki tepat satu solusi, dan Anda tidak boleh menggunakan elemen yang sama dua kali. Anda dapat mengembalikan jawabannya dalam urutan apa pun. Buatlah function dalam Go untuk menunjukkan hasil seperti dibawah.

Contoh 1:

Input: `nums = [2,7,11,15]`, `target = 9` Output: `[0, 1]`

SOAL 2

Diberikan sebuah array bilangan bulat `nums`, kembalikan semua triplet `[nums[i], nums[j], nums[k]]` sedemikian rupa sehingga $i \neq j$, $i \neq k$, dan $j \neq k$, dan $nums[i] + nums[j] + nums[k] == 0$.

Perhatikan bahwa himpunan solusi tidak boleh mengandung triplet yang duplikat.

Buatlah function dalam Go untuk menunjukkan hasil seperti dibawah dan sebutkan time complexity nya.

Contoh 1:

Input: `nums = [-1,0,1,2,-1,-4]` Output: `[[-1,-1,2],[-1,0,1]]`

SOAL 3

Diberikan string `s = "wordgoodgoodgoodbestword"` dan array kata-kata `words = ["word","good","best","word"]`.

Kembalikan semua indeks awal dari substring(s) dalam `s` yang merupakan penggabungan dari setiap kata dalam `words` tepat satu kali, dalam urutan apa pun, dan tanpa karakter yang di antaranya.

Output yang diharapkan: `[8]`

Catatan:

1. Silakan unggah jawaban Anda ke GitHub, pastikan repositori tersebut tidak diatur sebagai repositori pribadi.
2. Format untuk menjawab soal Query dapat berupa file dengan ekstensi .md, .txt, .go, atau .sql, sesuai dengan preferensi Anda (dalam repo github yang sama).
3. Untuk menjawab soal Go, jika Anda membangun solusi dalam bentuk REST API dan menerapkan framework serta arsitektur bersih (clean architecture), itu akan menjadi nilai tambah.
4. Waktu maksimal untuk mengerjakan tugas adalah 2 hari sejak Anda menerima soal.
5. Jika Anda memiliki pertanyaan lebih lanjut, jangan ragu untuk menghubungi HR atau mengirim email ke devananda@olin.co.id.