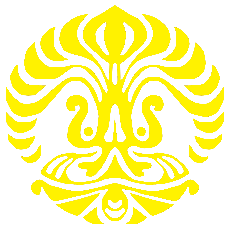
**LAPORAN TUTORIAL LAB 2**

**BASIS DATA**



**EDUARDUS TJTIRAHARDJA**

**2106653602**

**KELAS A**

**FAKULTAS ILMU KOMPUTER**

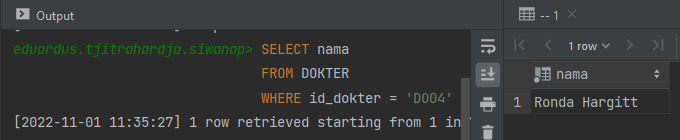
**UNIVERSITAS INDONESIA**

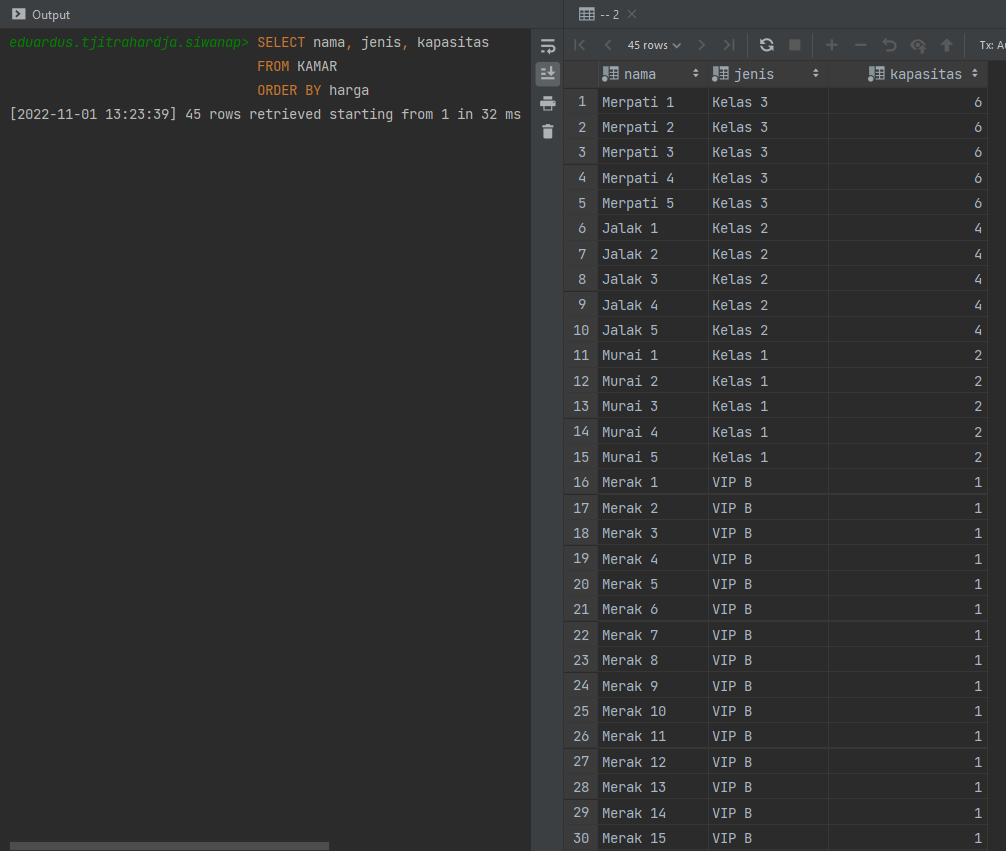
**DEPOK**

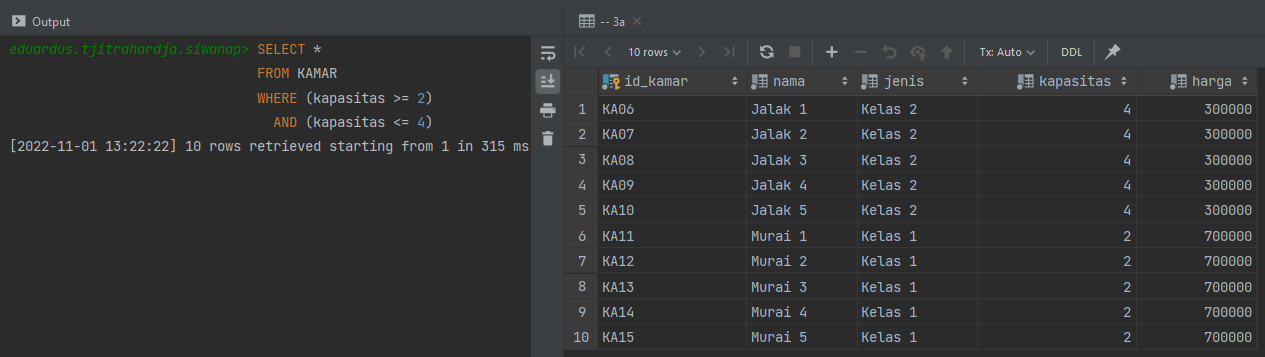
**2022/2023**

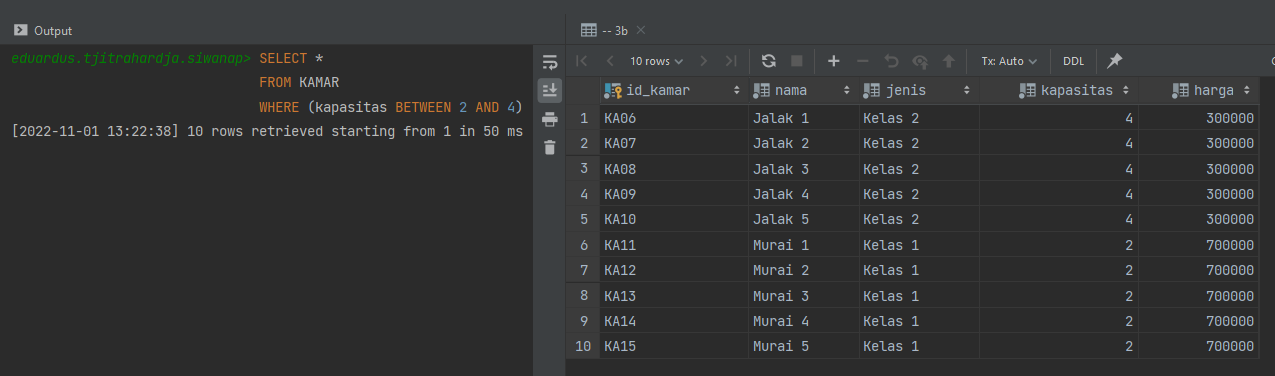
**Latihan 1**

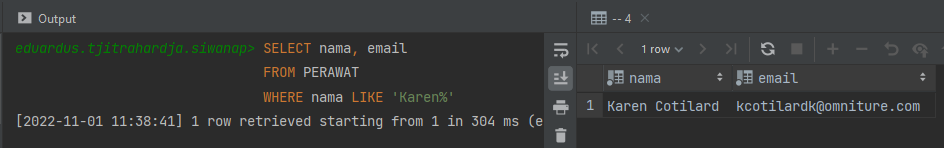
1. **[SQL]** Jalankan SQL Query pada Contoh 1 hingga Contoh 26 di atas dan cantumkan hasilnya pada laporan.

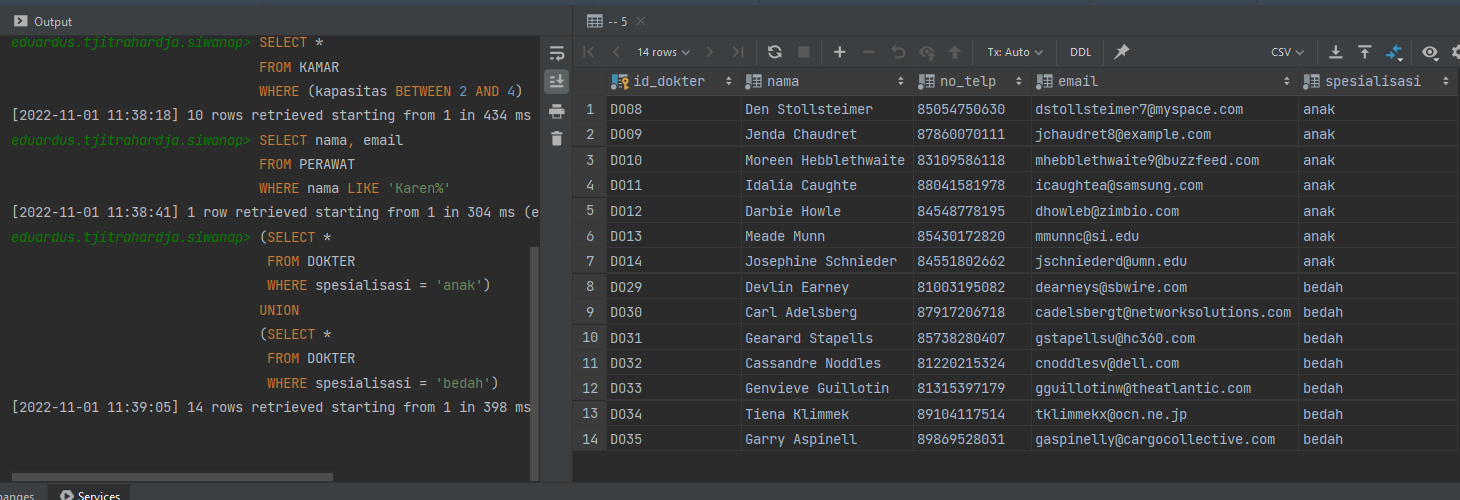
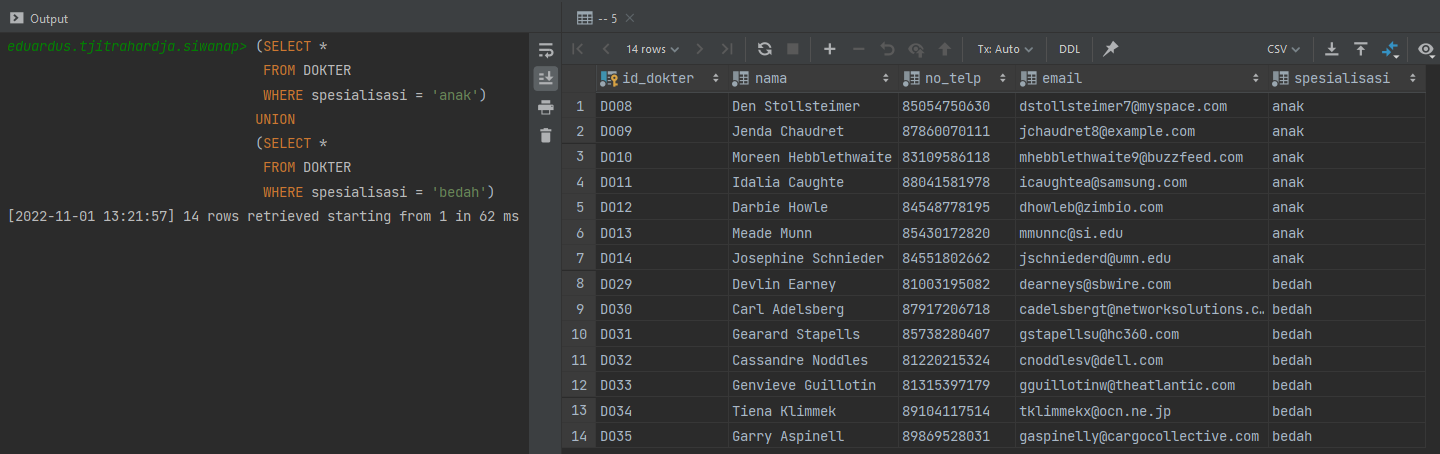
****

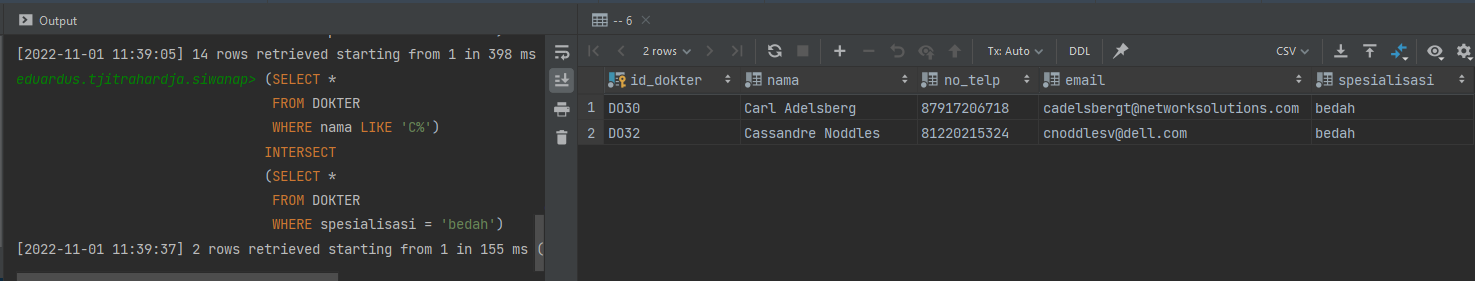
****

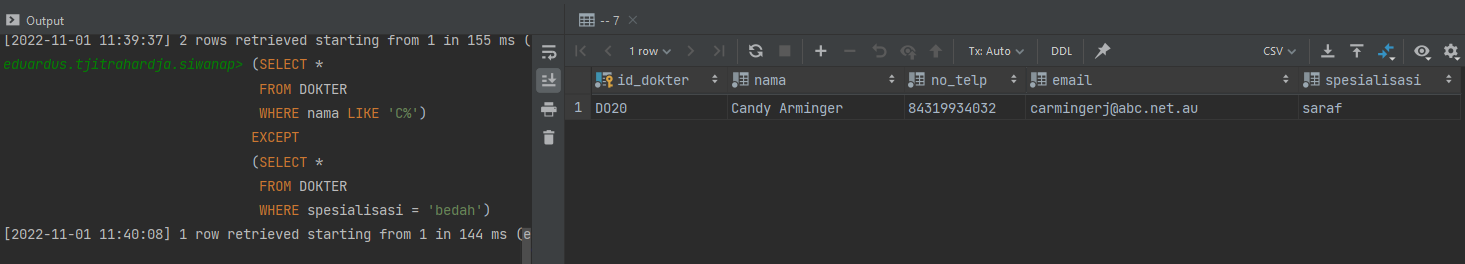
****

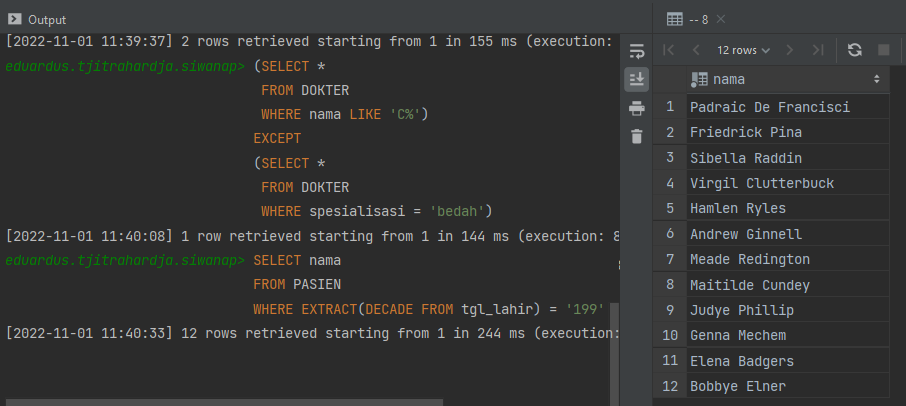
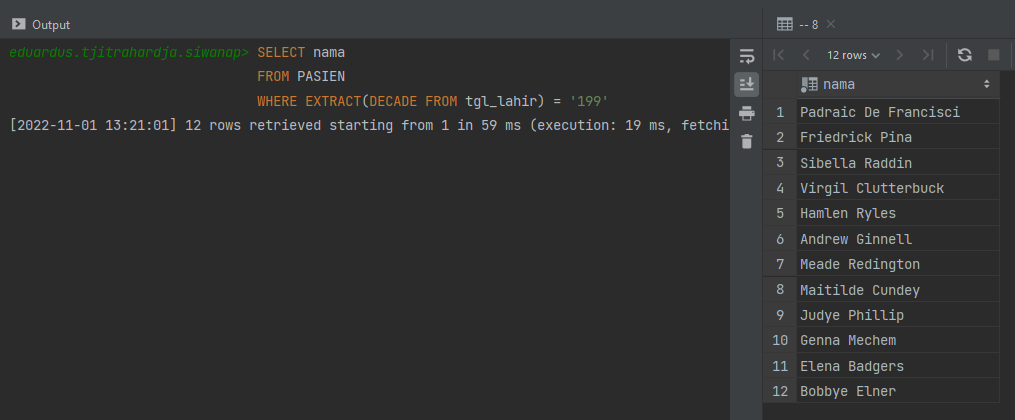
****

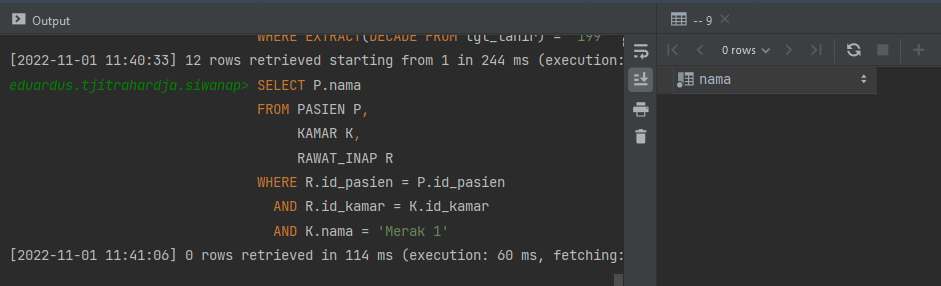
****

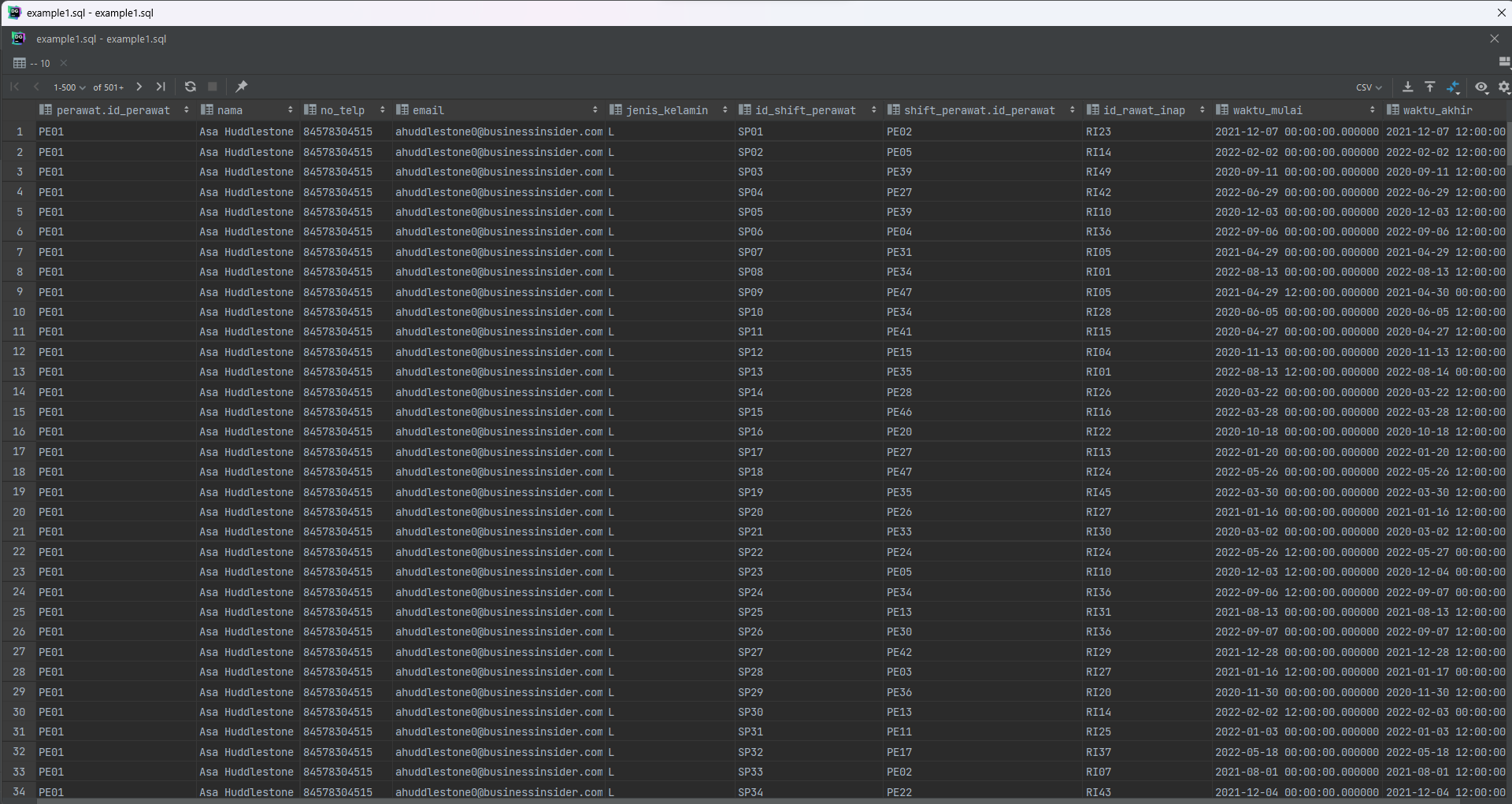
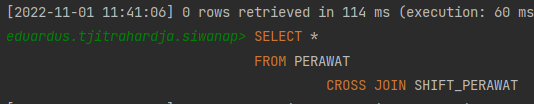
****

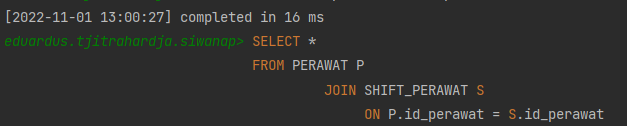
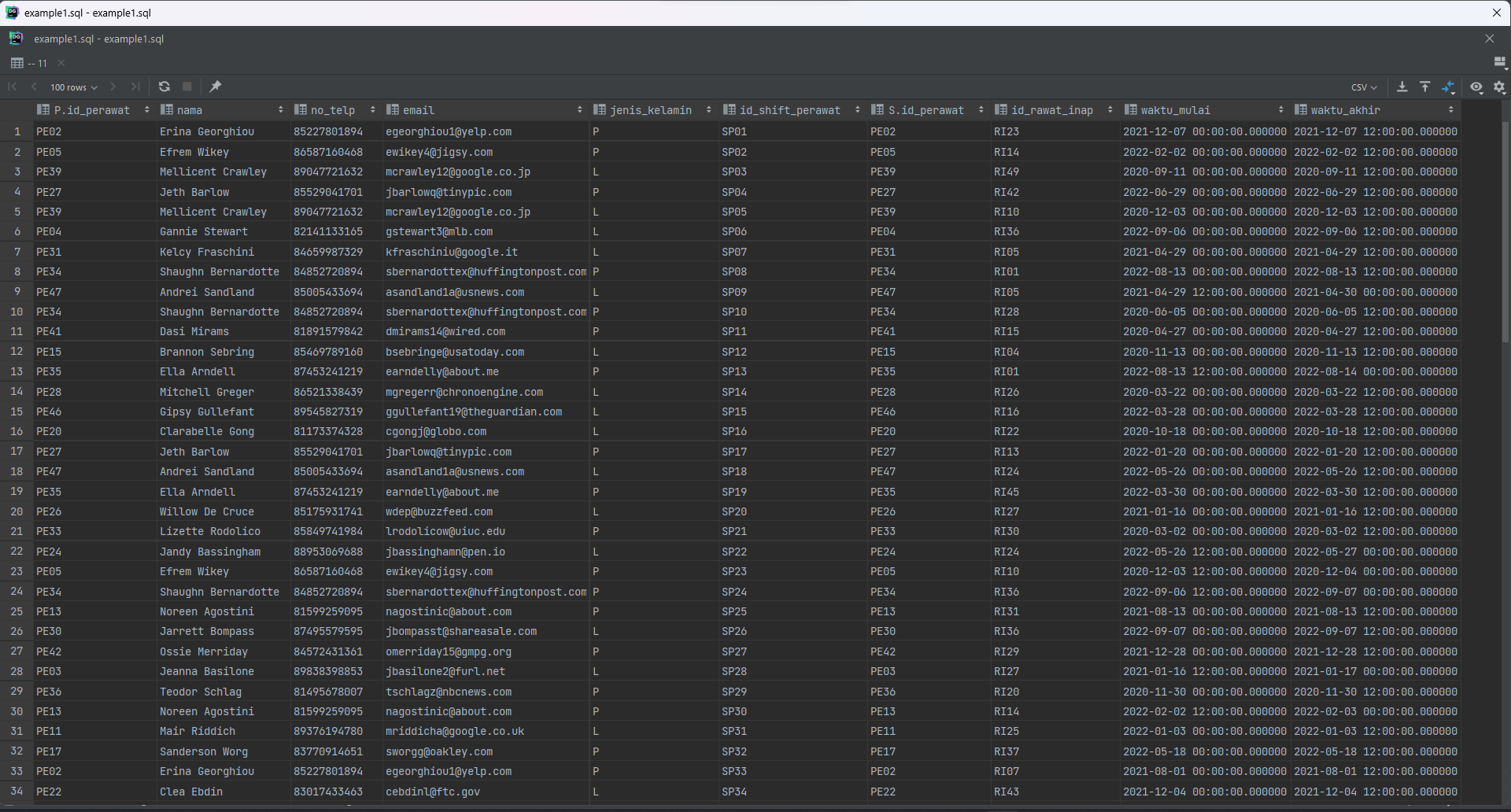
****

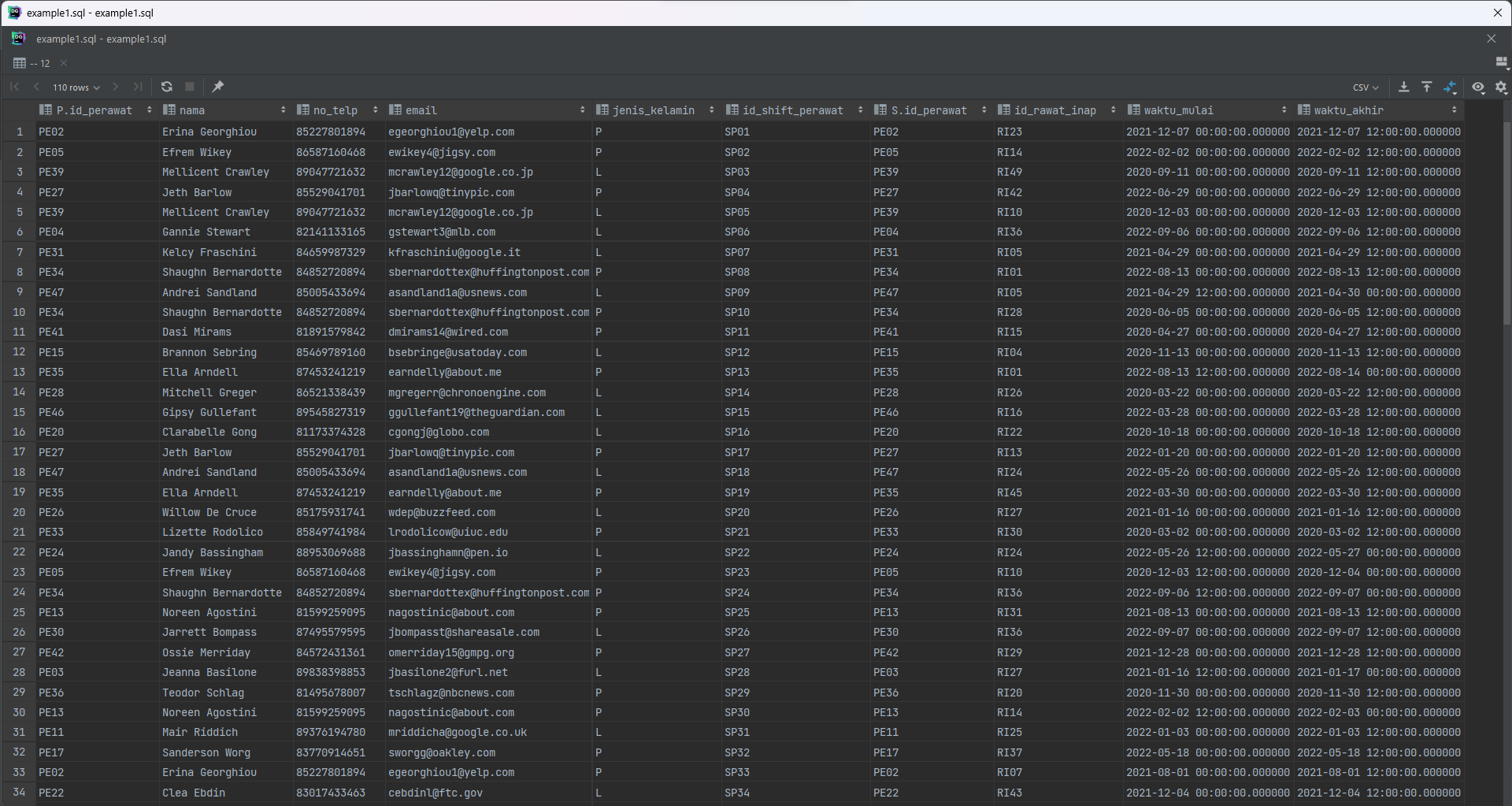
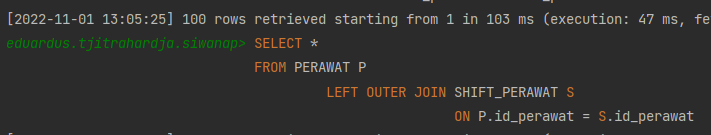
****

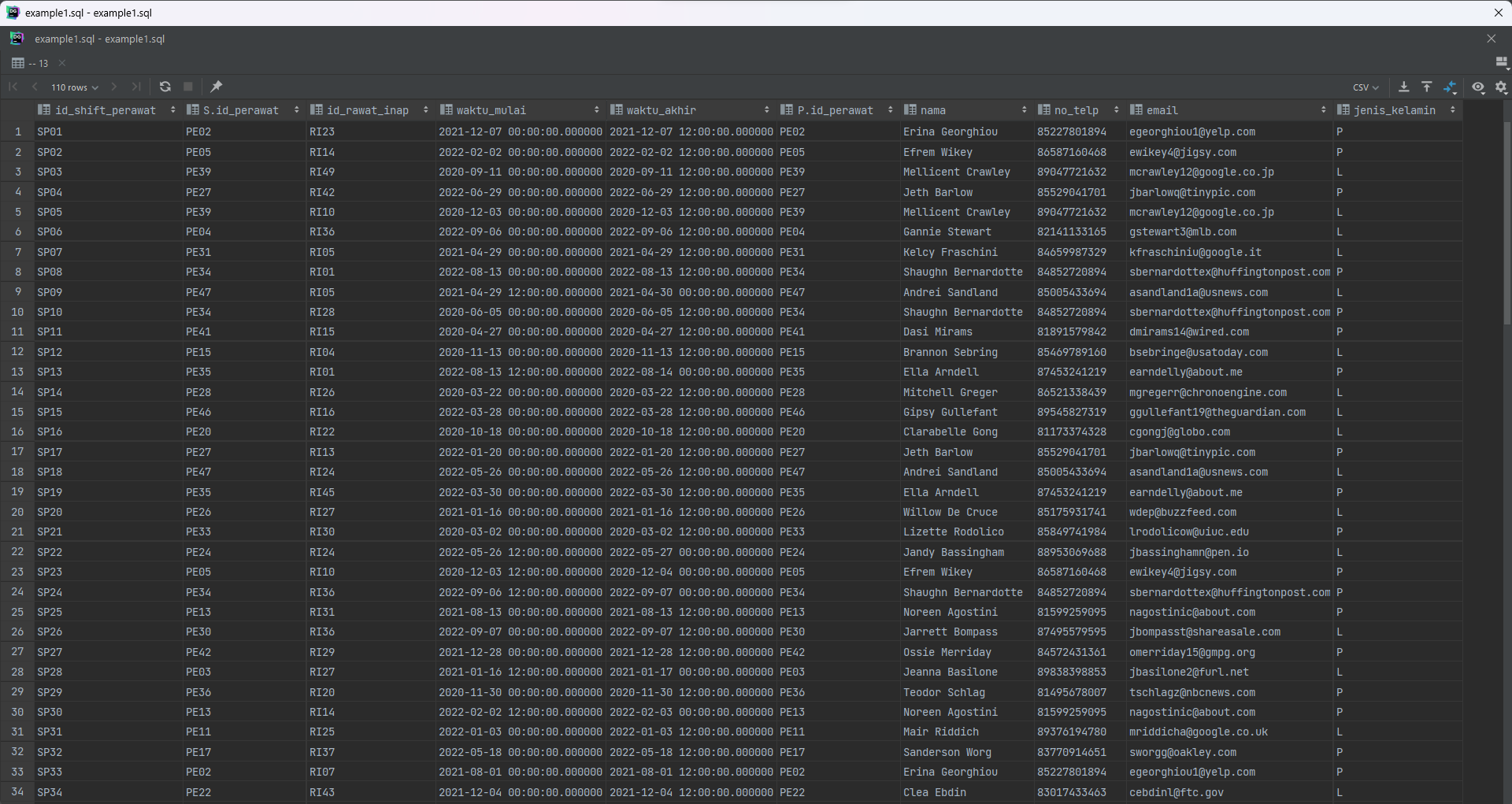
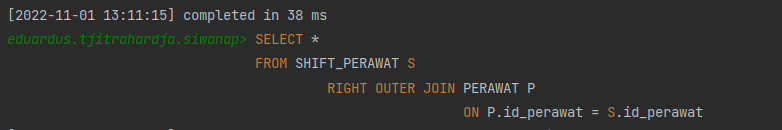


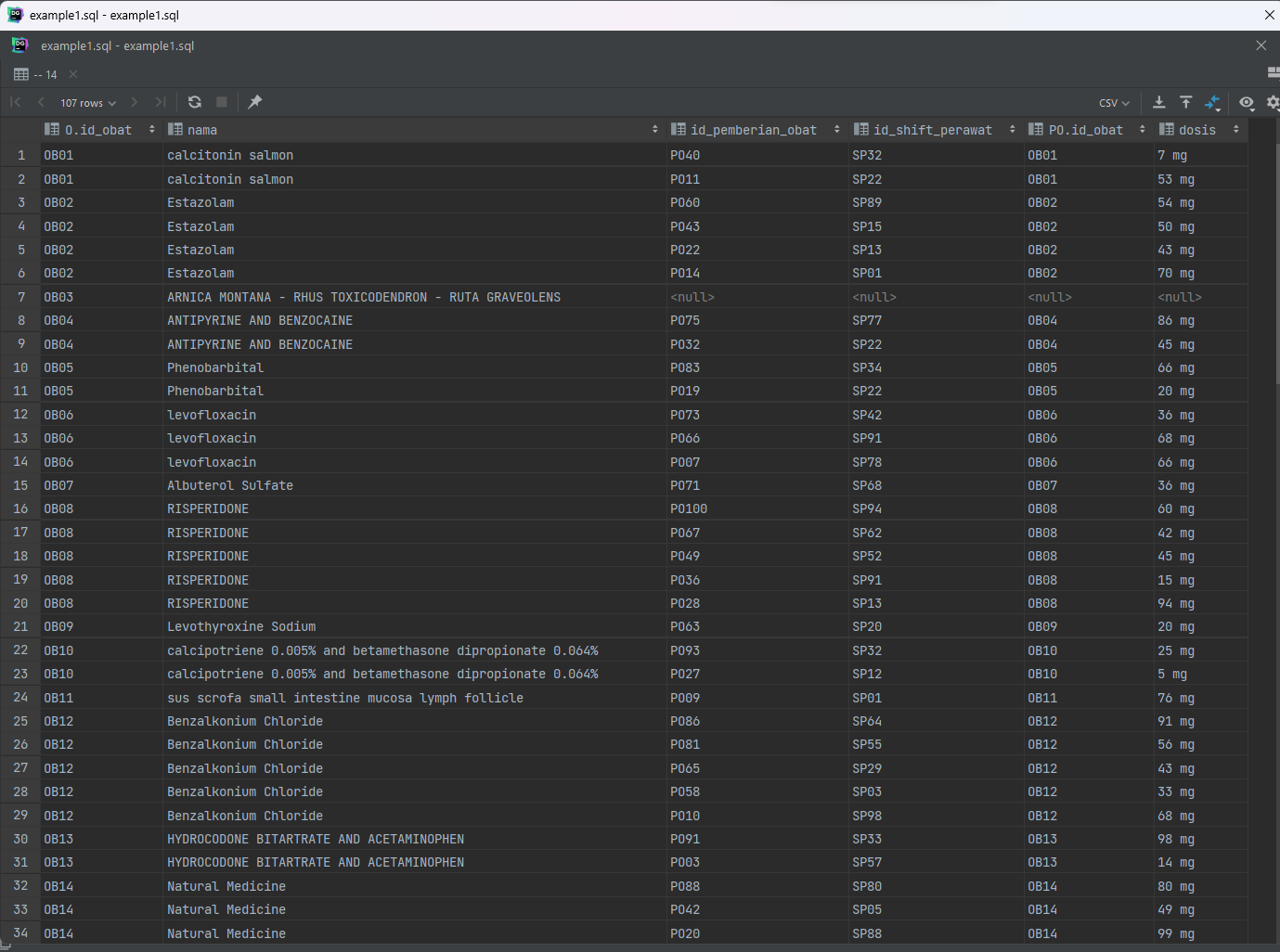
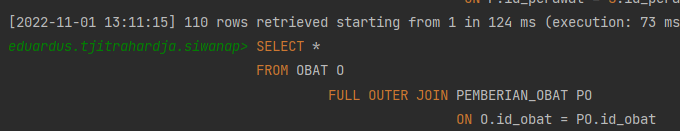
****

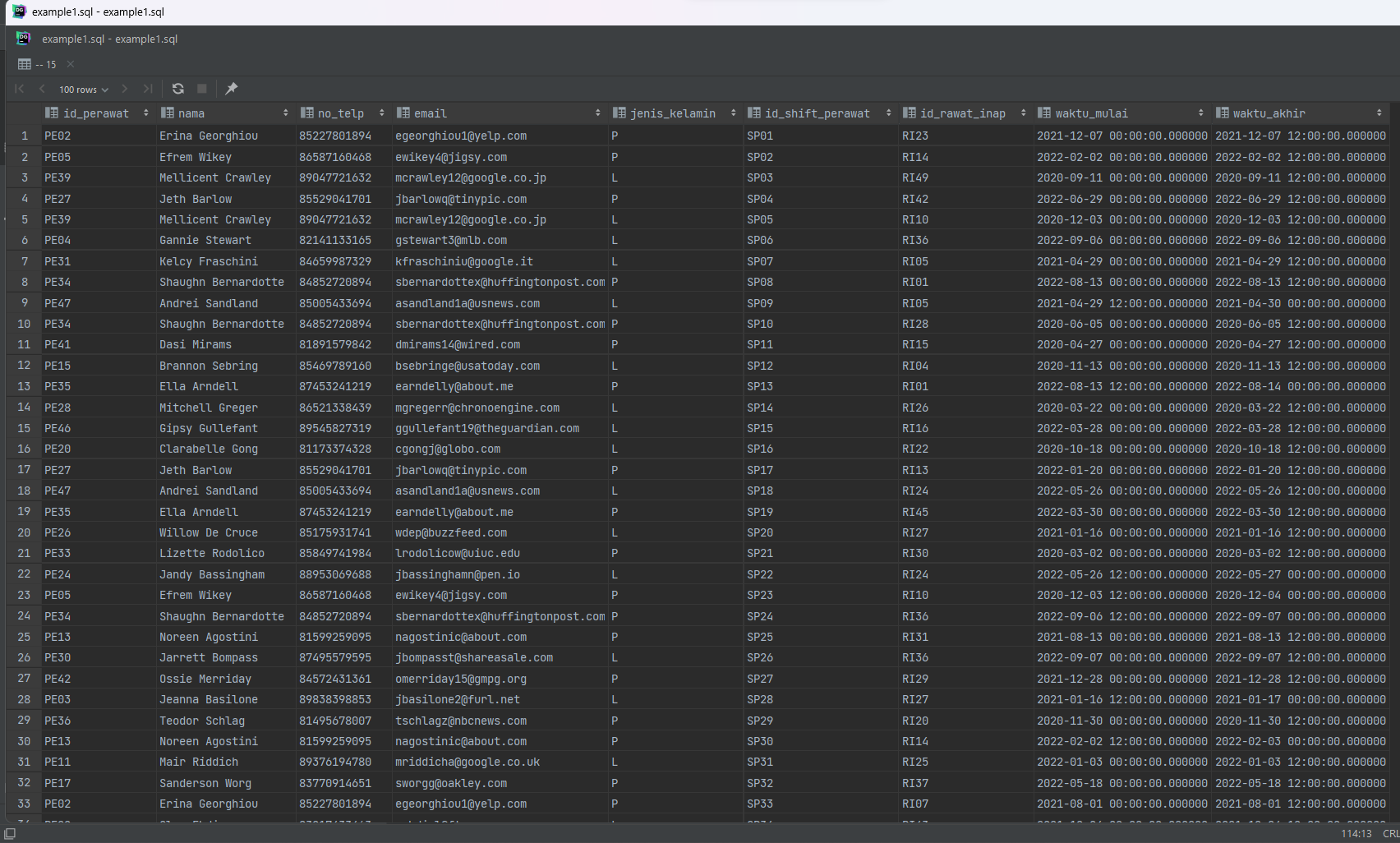
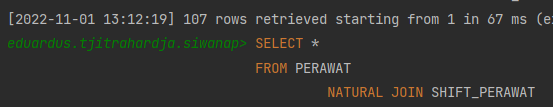
****

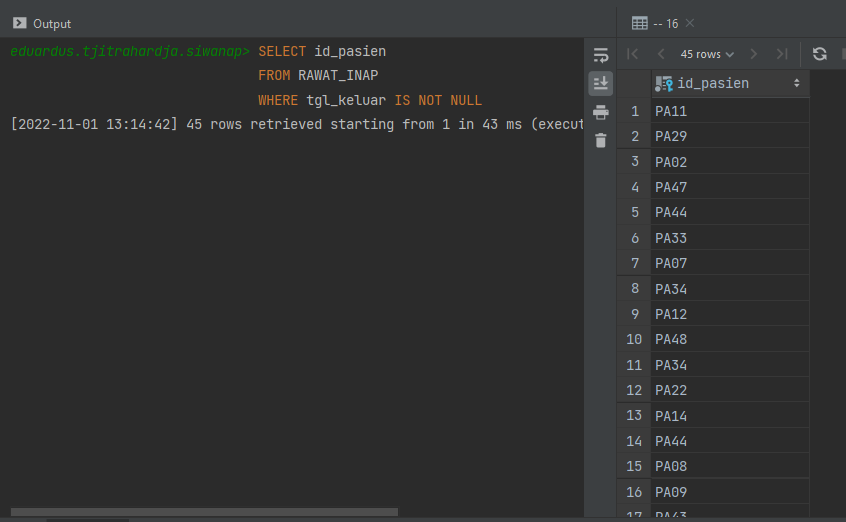
****

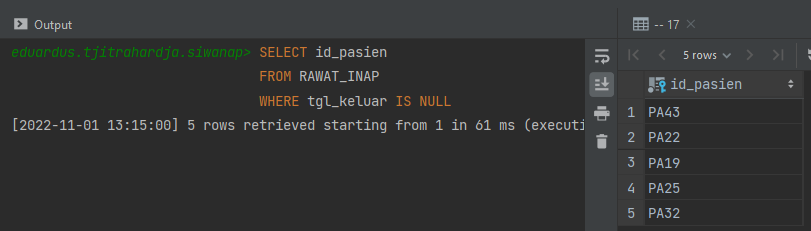
****

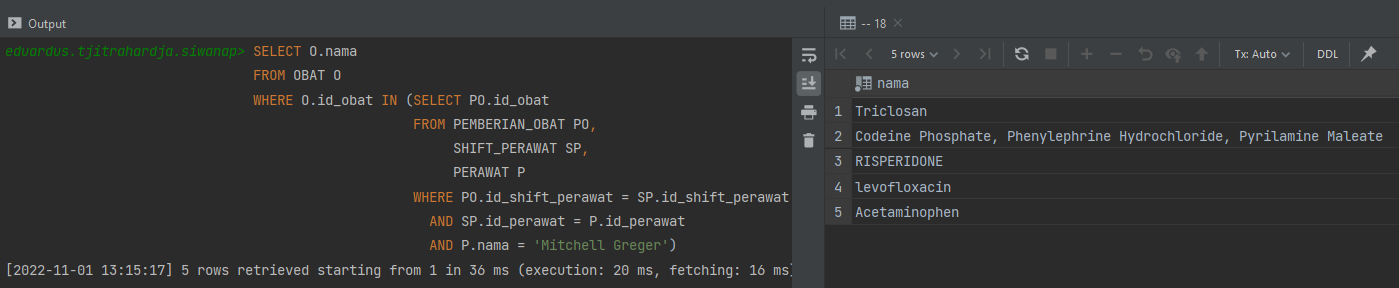
****

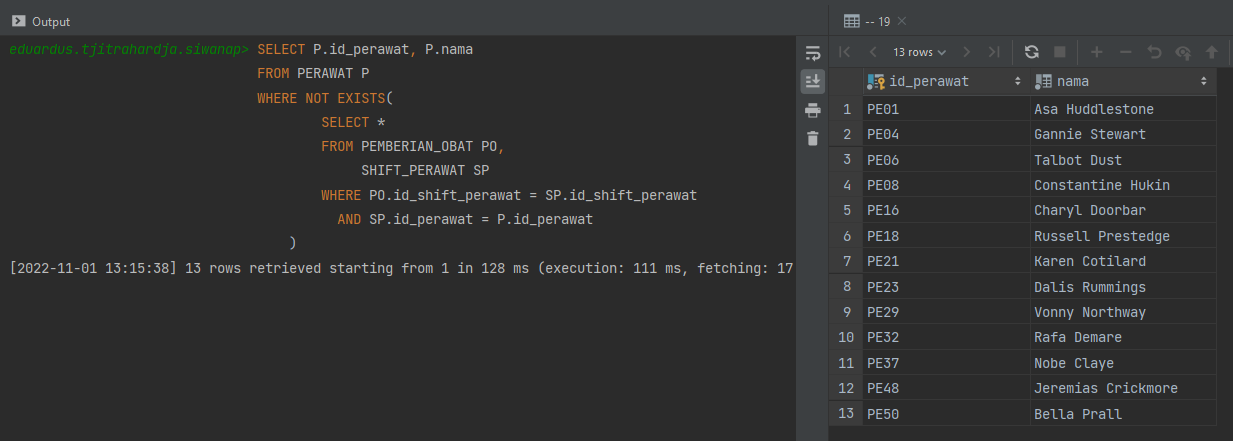
****

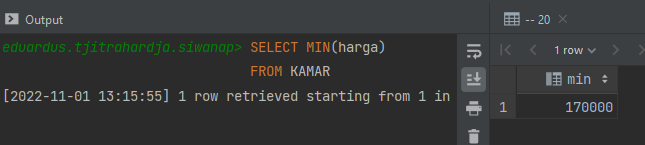
****

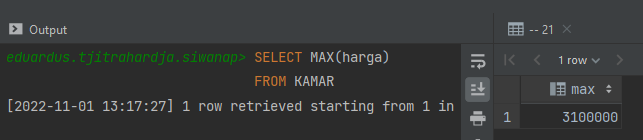
****

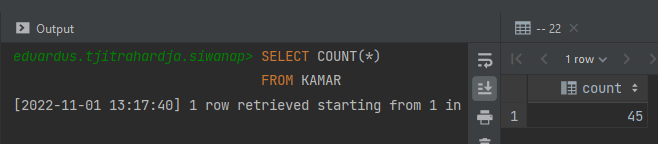
****

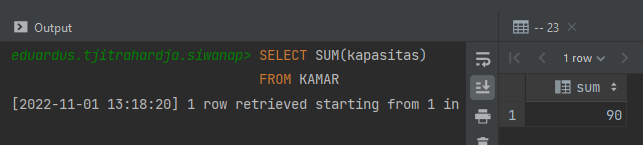
****

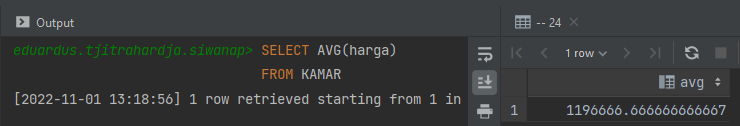
****

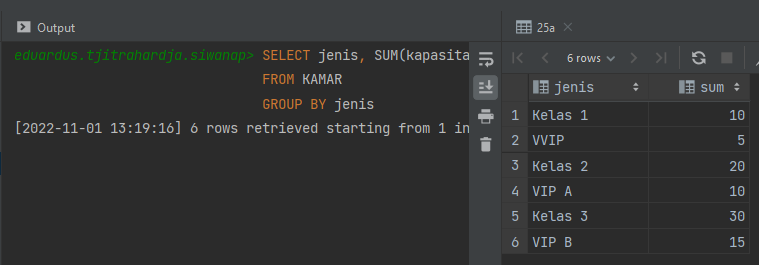
****

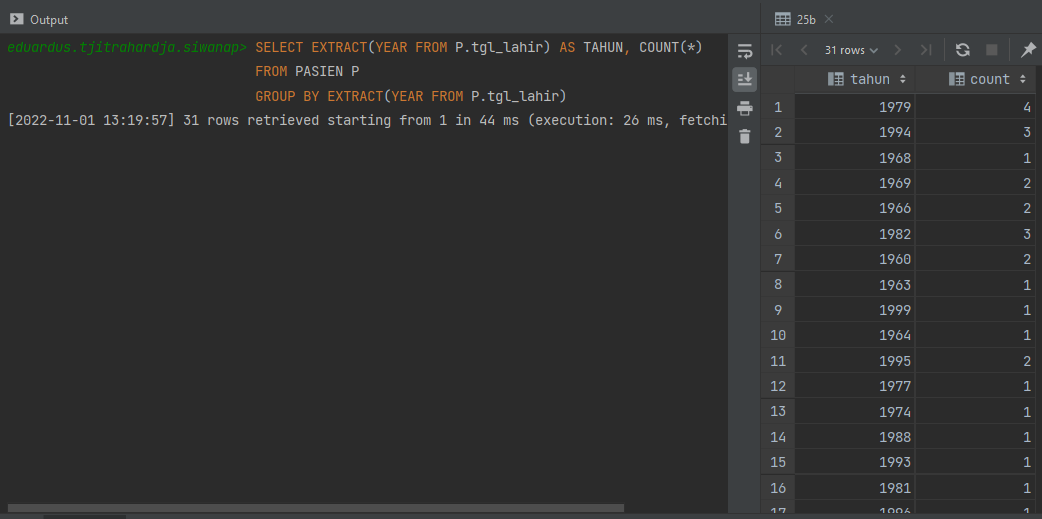
****

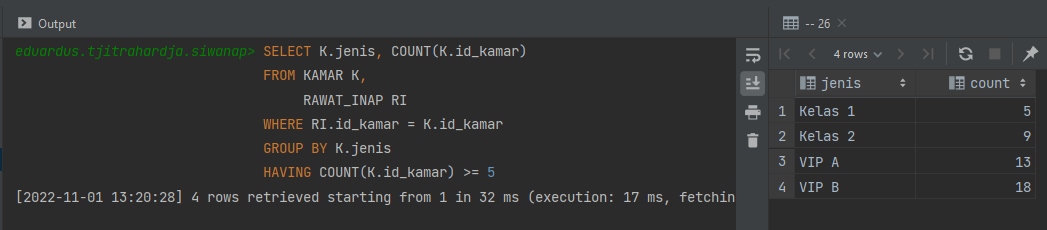
****

****

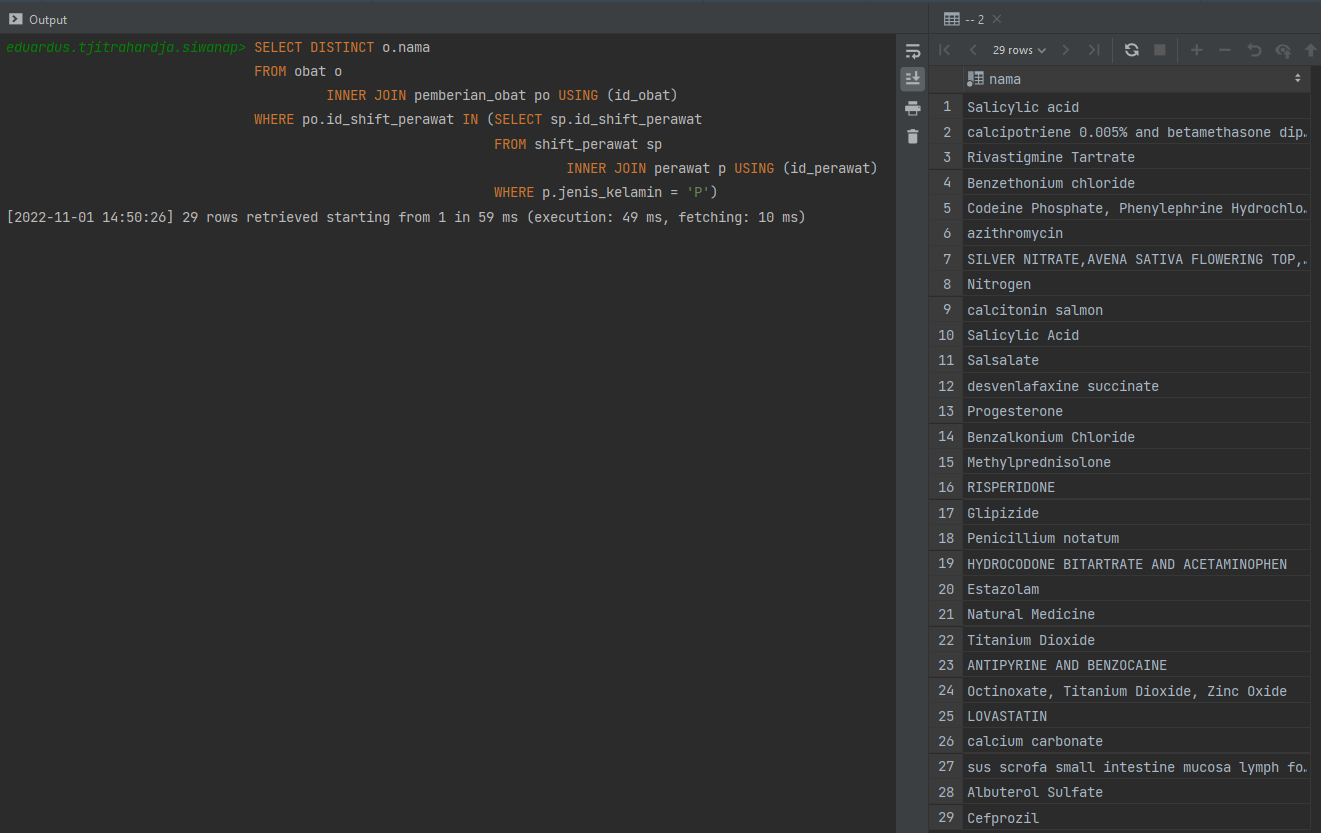
****

****

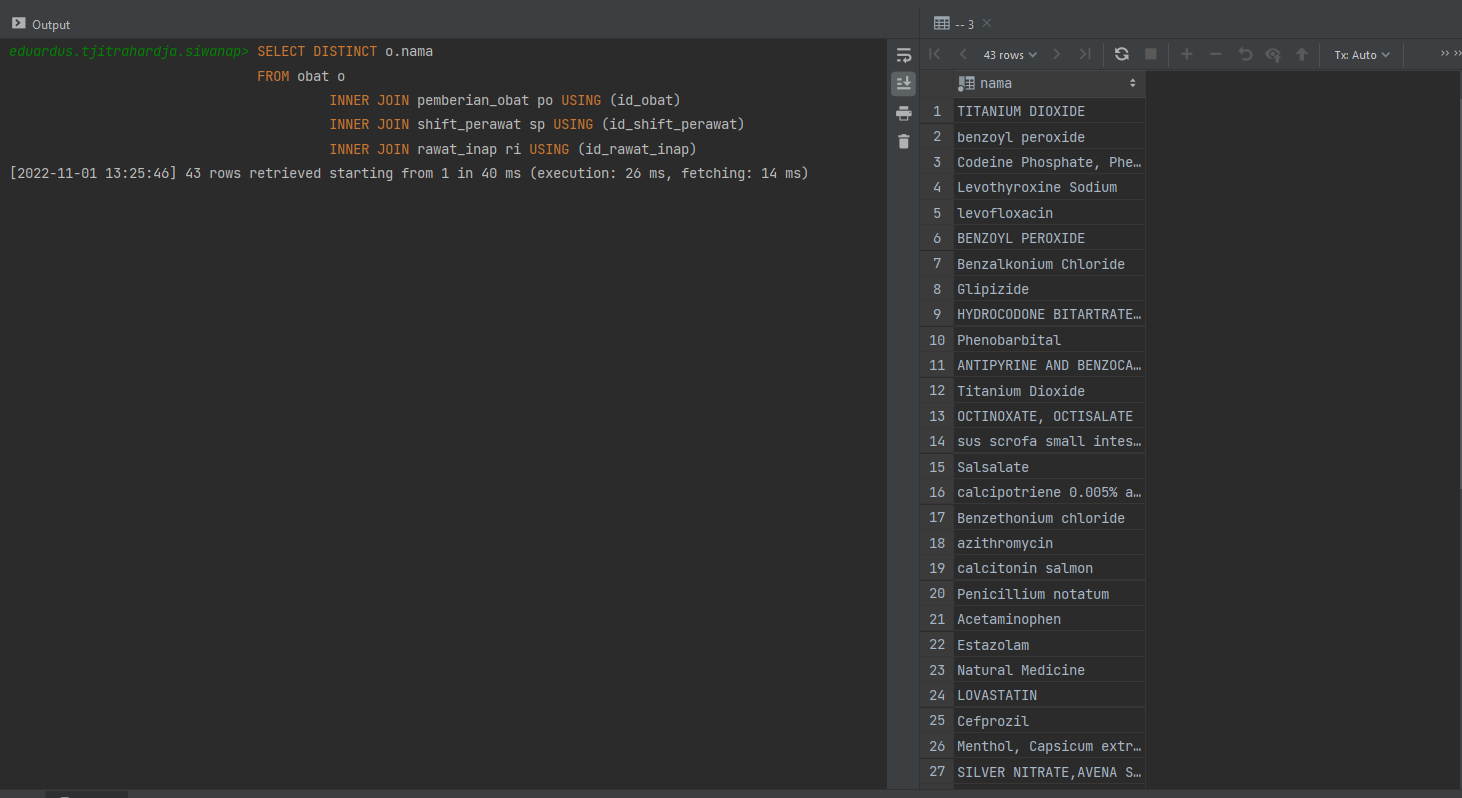
****

****

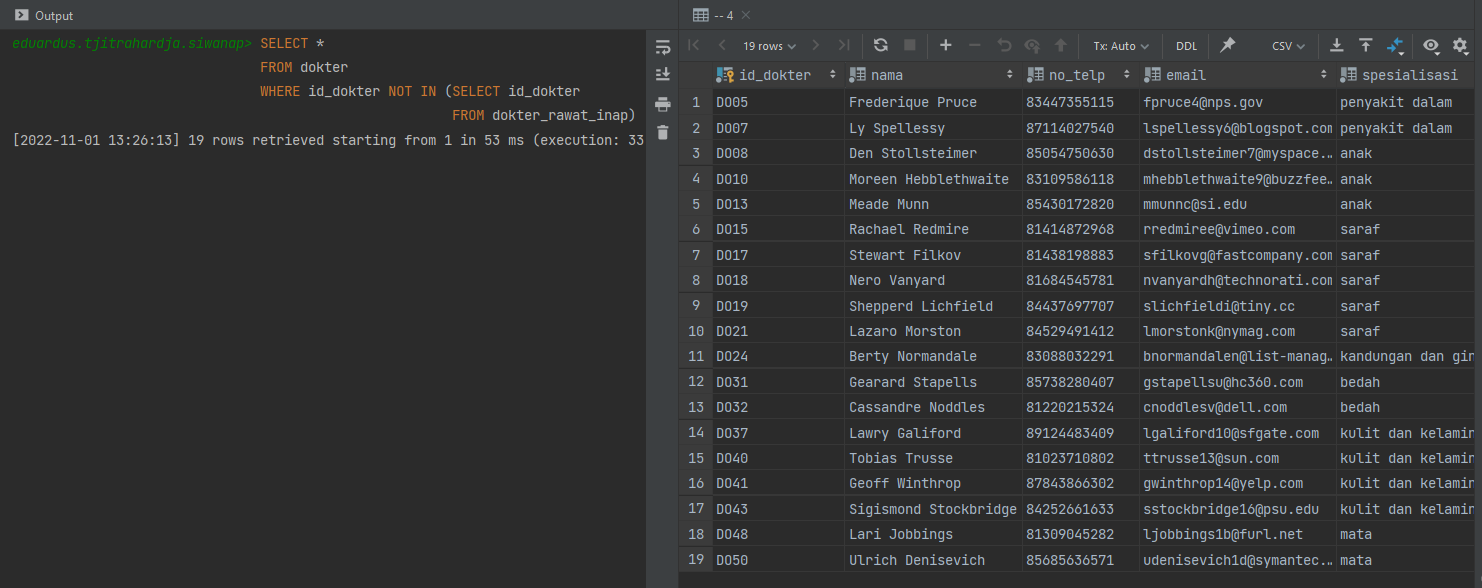
1. **[SQL]** Tampilkan nama obat yang telah diresepkan oleh perawat wanita dengan menggunakan keyword **IN**.

****

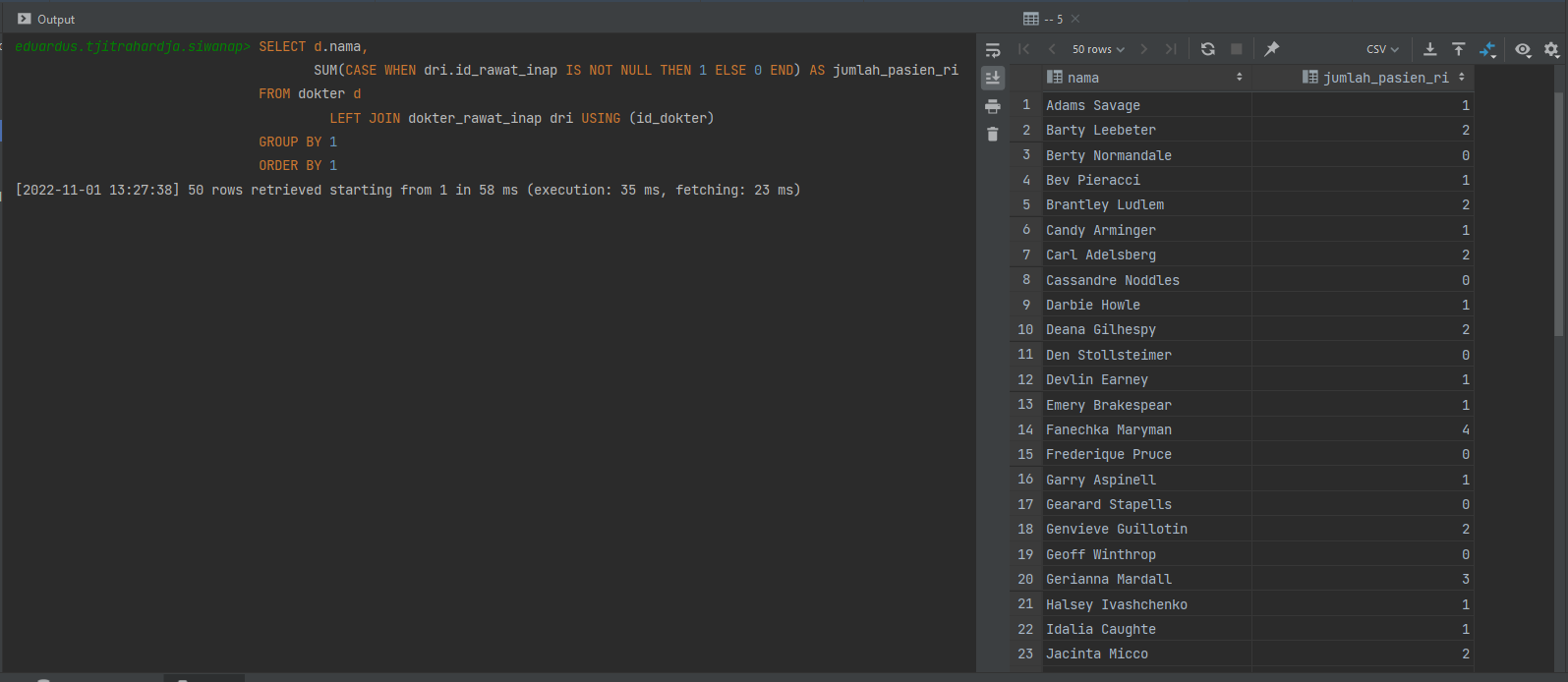
1. **[SQL]** Tampilkan nama UNIK obat yang **setidaknya** telah diberikan pada pasien.

****

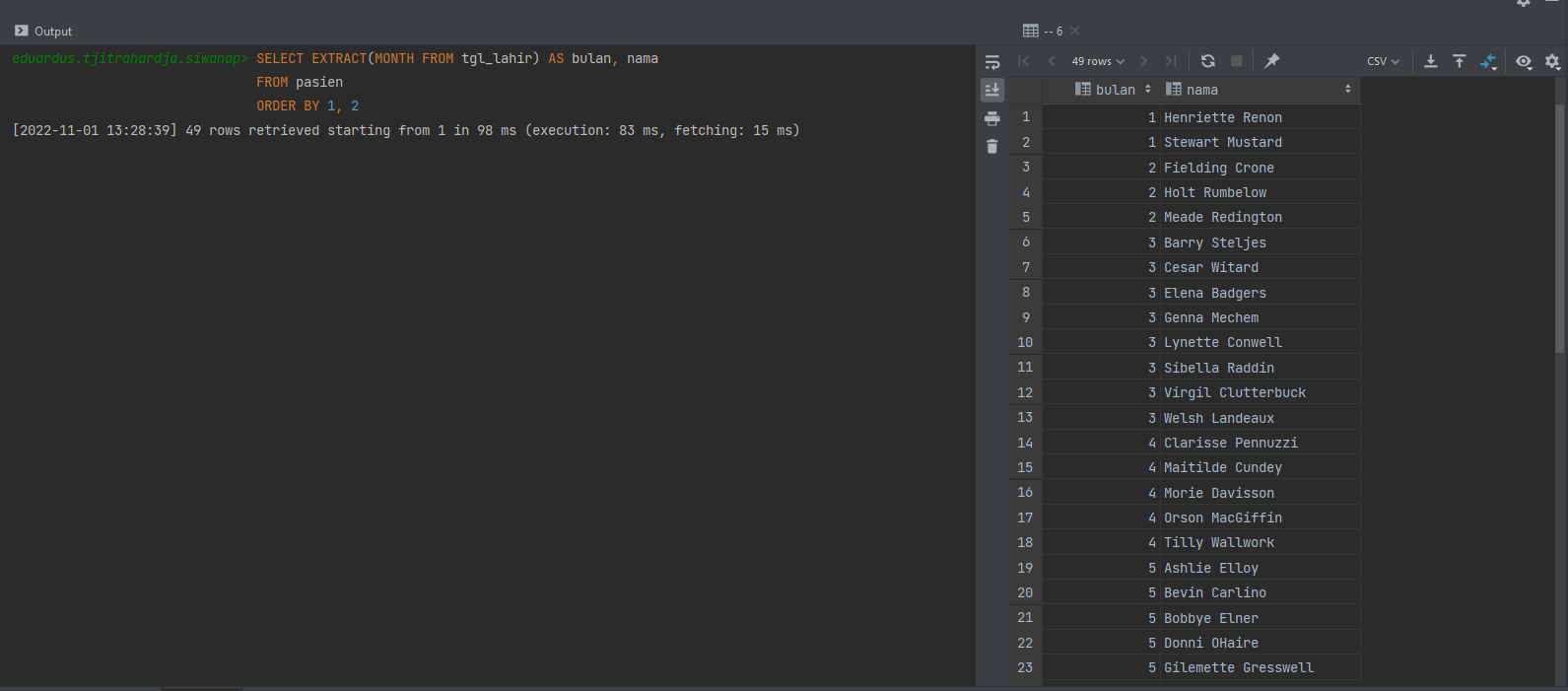
1. **[SQL]** Tampilkan daftar dokter yang tidak pernah merawat pasien rawat inap.

****

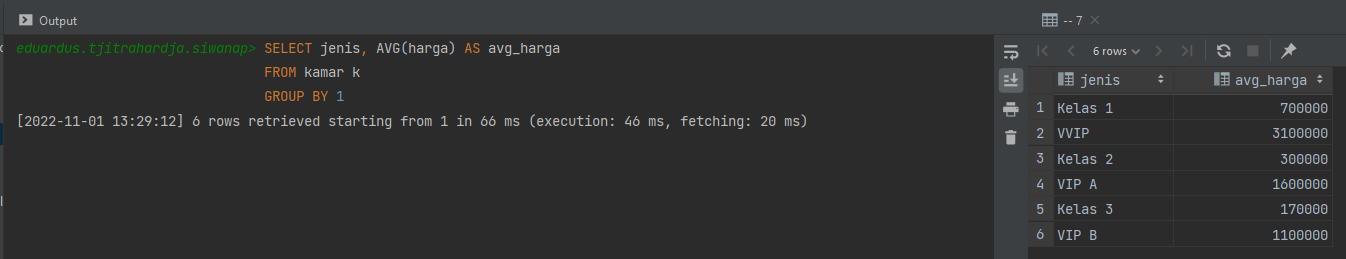
1. **[SQL]** Tampilkan nama dokter dengan total jumlah pasien rawat inap yang telah ditugaskan kepada dokter tersebut diurutkan berdasarkan ascending alphabetical order (A-Z) dari namanya, tanpa peduli jika dokter tersebut memiliki pasien rawat inap atau tidak.

****

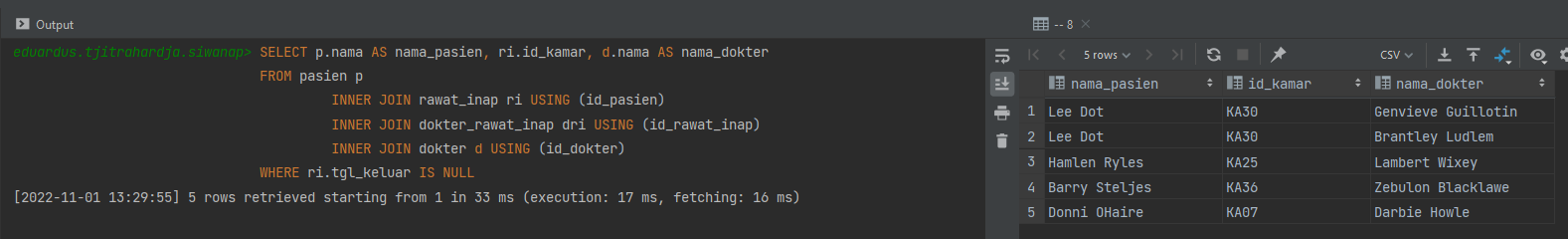
1. **[SQL]** Tampilkan jumlah pasien yang lahir pada setiap bulan. Anda disarankan menggunakan EXTRACT

****

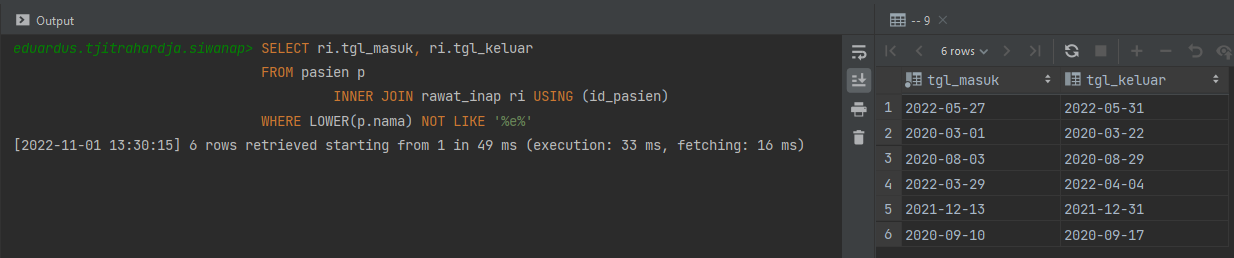
1. **[SQL]** Tampilkan **jenis kamar** dan **harga rata-rata** kamar untuk setiap jenis kamar.

****

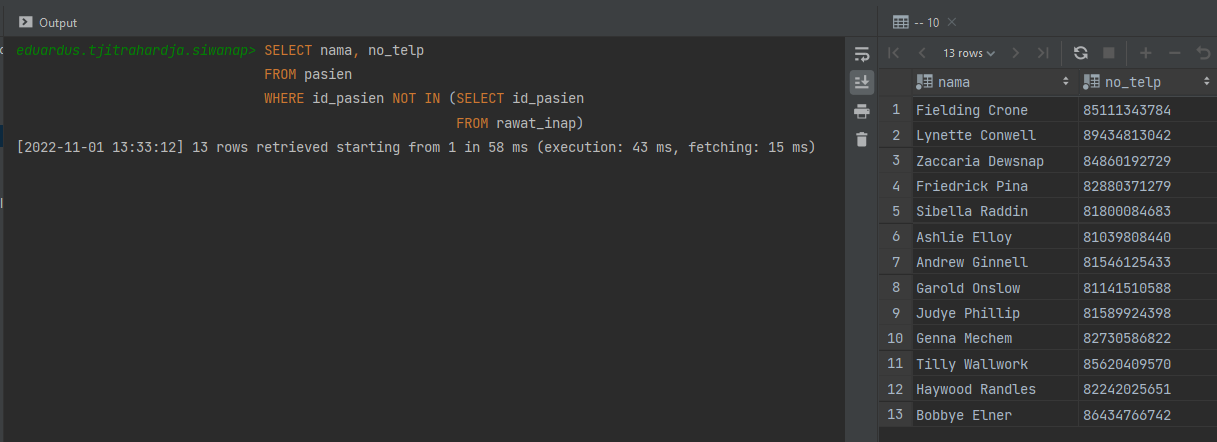
1. **[SQL]** Tampilkan **nama pasien, id** kamar, dan **nama dokter yang bertugas** untuk setiap pasien yang masih dirawat di rumah sakit.

****

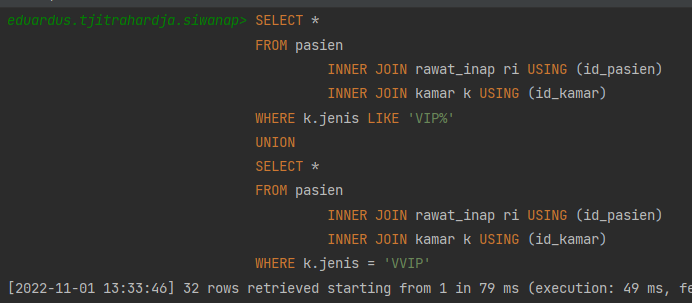
1. **[SQL]** Tampilkan **tanggal masuk** dan **tanggal keluar** setiap pasien yang namanya tidak mengandung huruf E (tidak case sensitive).

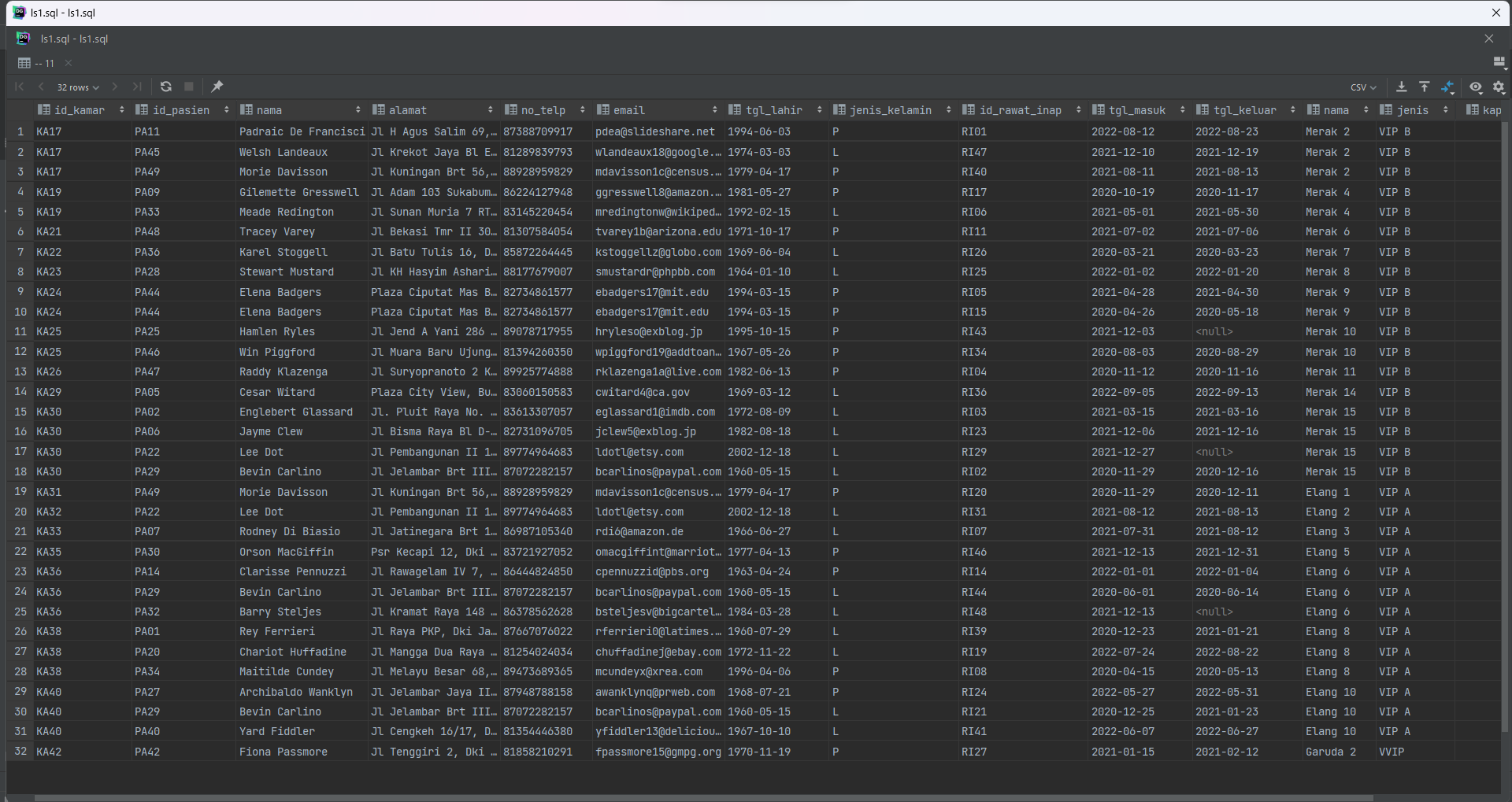
****

1. **[SQL]** Tampilkan **nama** dan **nomor telepon** pasien yang tidak pernah menjadi pasien rawat inap.

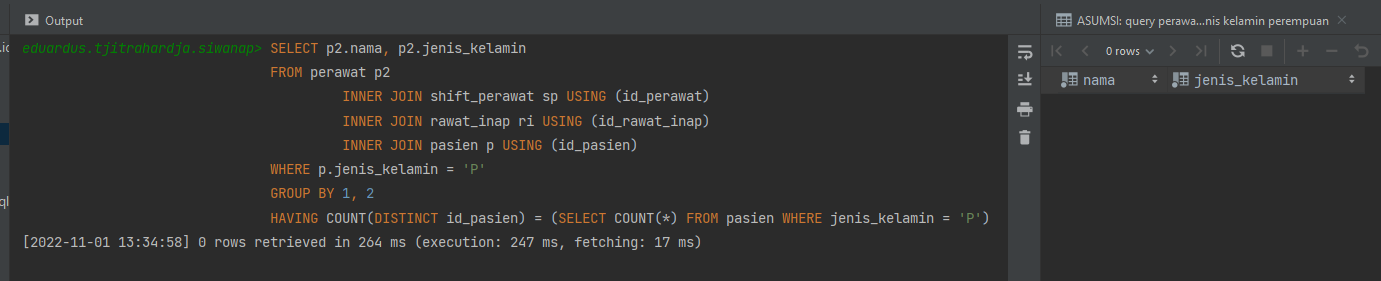
****

1. **[SQL]** Tampilkan daftar pasien yang pernah menjadi pasien rawat inap di kamar jenis VIP atau VVIP. Anda harus menggunakan keyword **UNION.**

****

****

1. **[SQL]** Tampilkan **nama** dan **jenis** **kelamin** perawat yang merawat semua pasien wanita.

****

1. **[Trivia]** Apakah kita mungkin mendapatkan data tertentu dari operasi inner join menggunakan keyword **IN**?

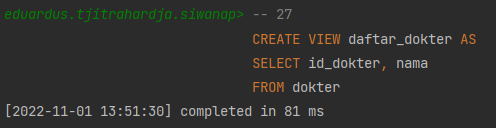
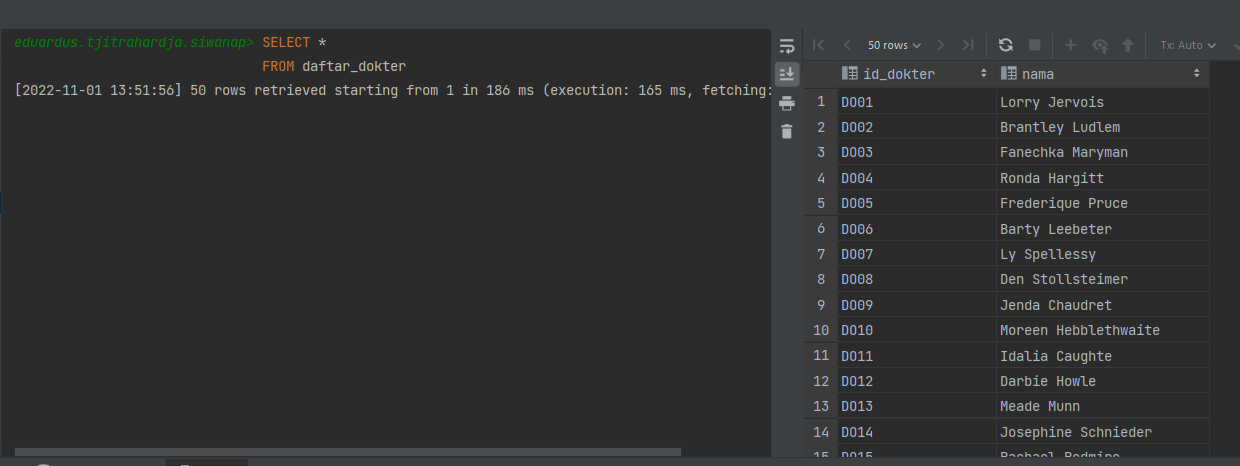
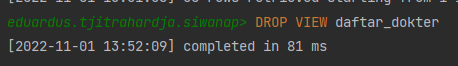
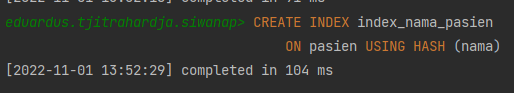
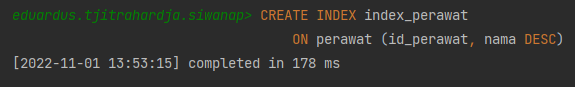
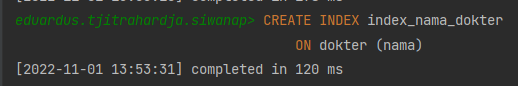
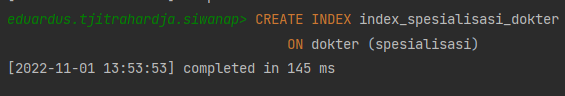
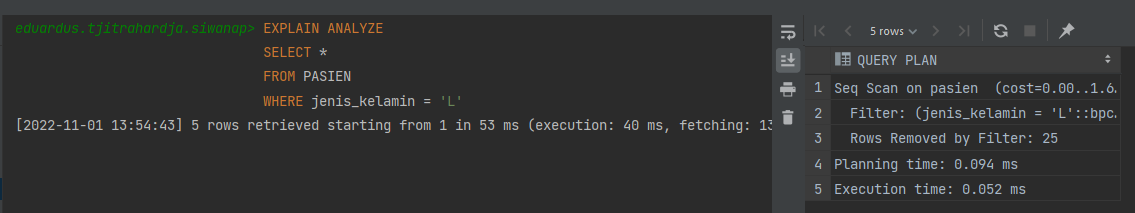
**Mungkin, kita tinggal query suatu table yang ada WHERE id IN (SELECT id FROM [table] WHERE [condition]) dimana idnya itu adalah PK-nya menghubungi kedua table tersebut. Contohnya seperti di nomor 2 bisa ditulis dengan INNER JOIN semua, seperti:**

SELECT o.nama  
FROM obat o  
 INNER JOIN pemberian\_obat po USING (id\_obat)  
 INNER JOIN shift\_perawat sp USING (id\_shift\_perawat)  
 INNER JOIN perawat p USING (id\_perawat)  
WHERE p.jenis\_kelamin = 'P';

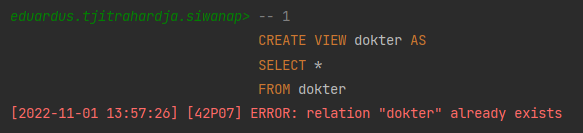
**Bisa juga ditulis dengan salah satu dari INNER JOINnya diganti dengan IN, seperti:**

SELECT o.nama  
FROM obat o  
 INNER JOIN pemberian\_obat po USING (id\_obat)  
WHERE po.id\_shift\_perawat IN (SELECT sp.id\_shift\_perawat  
 FROM shift\_perawat sp  
 INNER JOIN perawat p USING (id\_perawat)  
 WHERE p.jenis\_kelamin = 'P');

**Latihan 2**

1. **[SQL]** Jalankan SQL Query pada Contoh 27 hingga Contoh 35 di atas dan cantumkan hasilnya pada laporan.
   * 1. ****
     2. ****
     3. ****
     4. ****
     5. ****
     6. ****
        1. 
        2. ****
     7. ****
     8. ****
2. **View**
   1. **[Trivia]** Apa yang akan terjadi jika kita membuat Viewmenggunakan nama yang sama dengan nama tabel yang ada pada database? Jelaskan!

**Tidak bisa, console akan mengeluarkan error “ERROR: relation "dokter" already exists” seperti ini:**

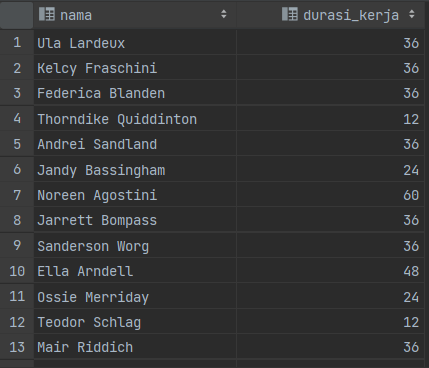


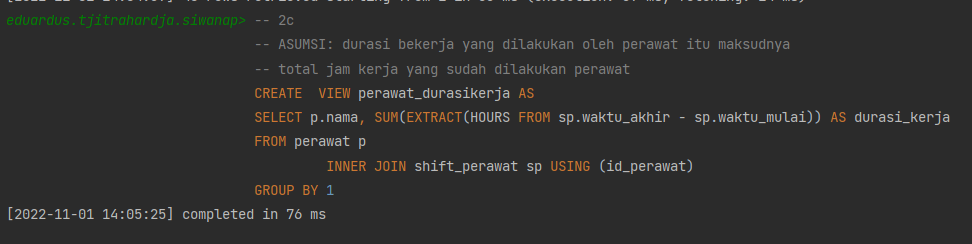
**karena VIEW berguna sebagai table sementara, sehingga jika kita menamakannya sama dengan salah satu table yang sudah ada akan bertabrakan namanya.**

* 1. **[Trivia]** Apa fungsi TEMP atau TEMPORARY di View?

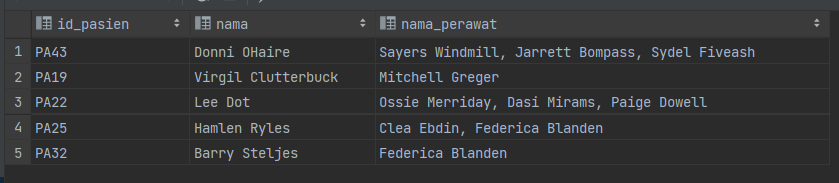
**TEMP atau TEMPORARY di View berguna untuk memberitahu SQL server bahwa View ini adalah *temporary*/sementara. Dimana setelah sesi sekarang berakhir, view *temporary* tersebut akan didrop.**

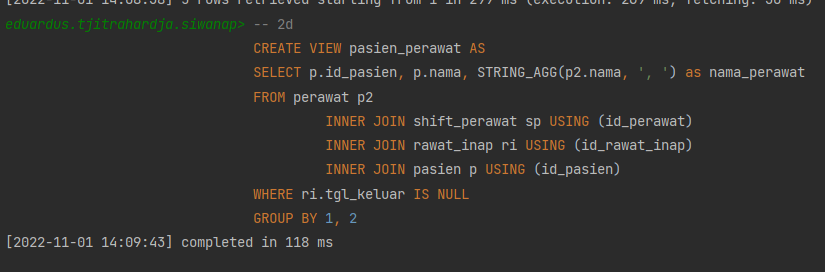
* 1. **[SQL]** Buatlah View yang menyimpan **nama** beserta **durasi bekerja** yang dilakukan oleh perawat.





* 1. **[SQL]** Buatlah View yang menyimpan **nama-nama perawat** yang merawat pasien dikelompokkan berdasarkan id pasien dan **nama pasien** yang belum keluar dari rumah sakit (HINT: string\_agg)





1. **Indexing and Analyze**

Diberikan query berikut.

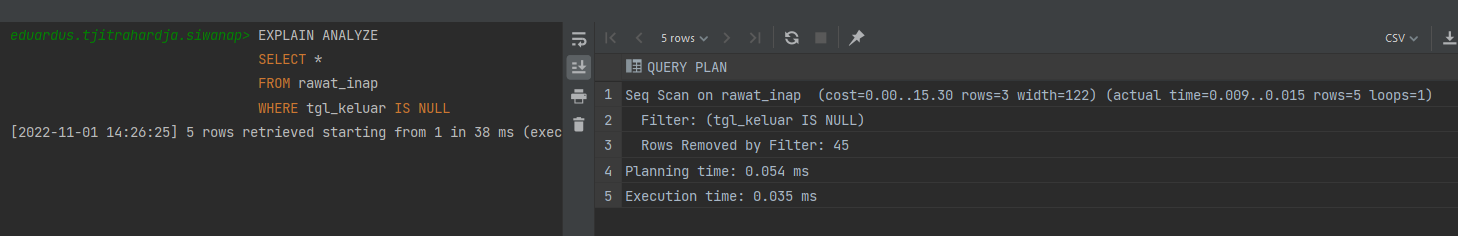
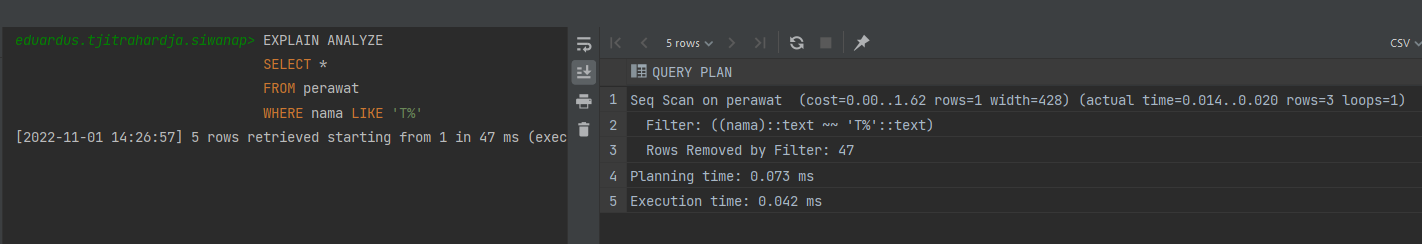
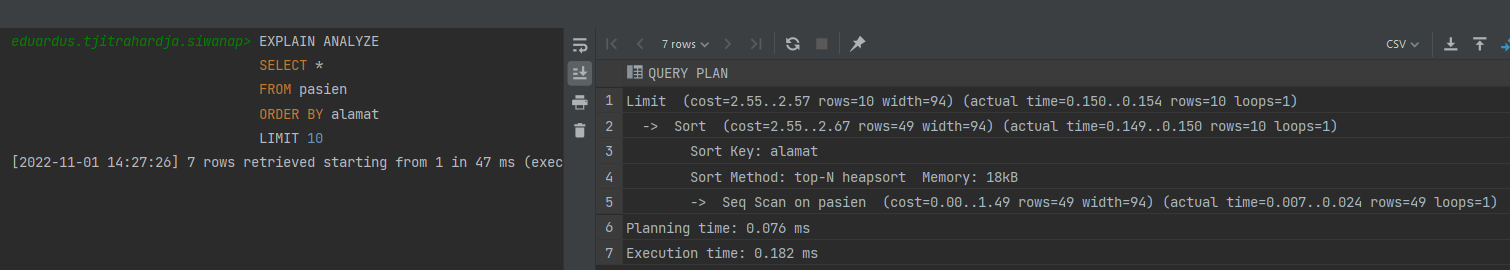
SELECT \* FROM kamar ORDER BY harga DESC;

SELECT \* FROM rawat\_inap WHERE tgl\_keluar IS NULL;

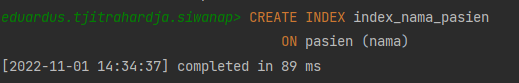
SELECT \* FROM perawat WHERE nama LIKE 'T%';

SELECT \* FROM pasien ORDER BY alamat LIMIT 10;

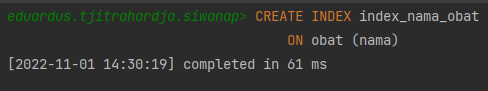
1. **[SQL]** Jalankan perintah EXPLAIN ANALYZE untuk setiap *query* di atas. *Screenshot* eksekusinya dan tulis hasilnya pada tabel di bawah, sertakan dalam laporan submisi Anda.

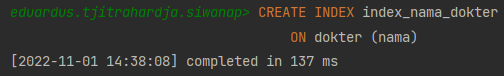
1. **[SQL]** Buat *index* berikut (*method* nya terserah Anda):
2. index\_nama\_pasien pada tabel **PASIEN** kolom **nama**.



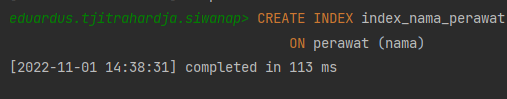
1. index\_nama\_obat pada tabel **OBAT** kolom **nama**.



1. index\_nama\_dokter pada tabel **DOKTER** kolom **nama**.

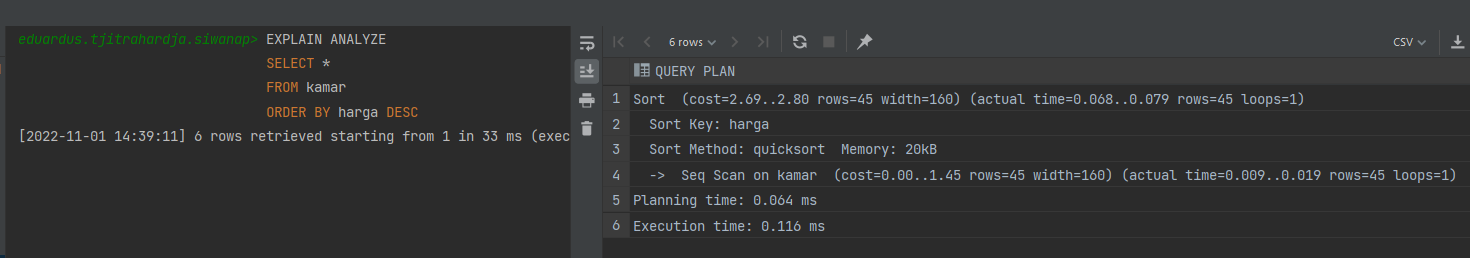
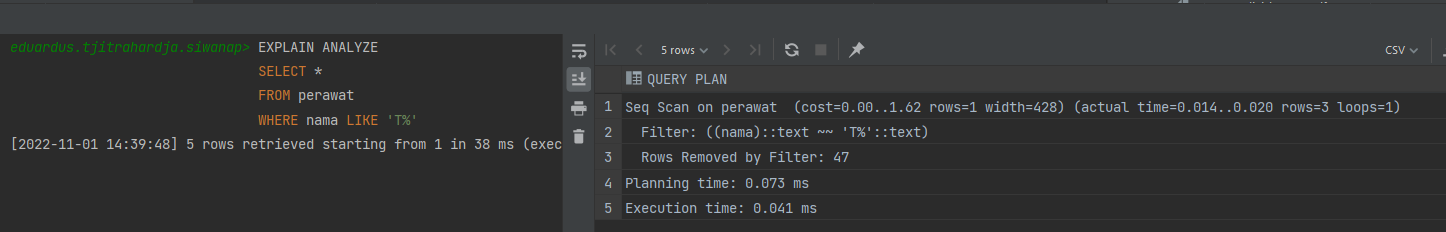
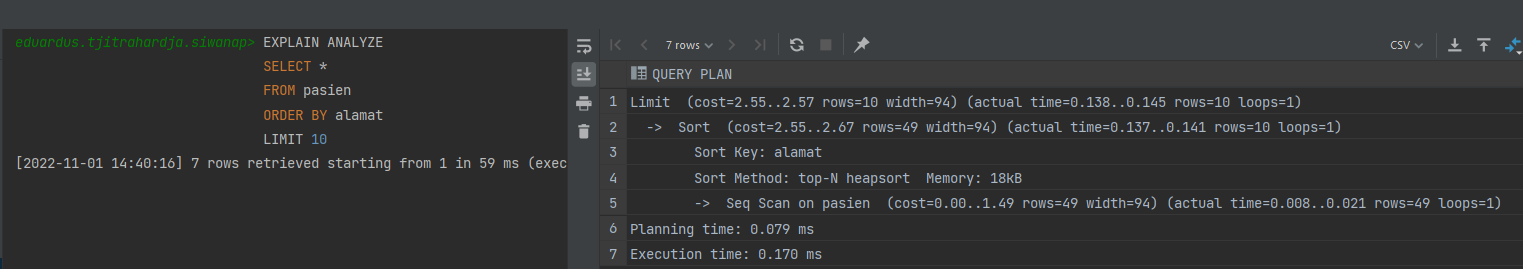


1. index\_nama\_perawat pada tabel **PERAWAT** kolom **nama**.



Tampilkan *query* dan *index* untuk setiap tabel di dalam laporan submisi Anda.

1. **[SQL]** Jalankan kembali **setiap *query*** **SELECT** di atas dari pertanyaan nomor 3 menggunakan perintah EXPLAIN ANALYZE. *Screenshot* eksekusinya dan tulis hasilnya pada tabel di bawah, sertakan dalam laporan submisi Anda.

1. **[Trivia]** Bandingkan *planning time* dan *execution time* (menggunakan tabel di bawah) dari *query* saat tanpa index dan setelah menggunakan *index*. Mana yang lebih baik? Berikan penjelasan!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Query*** | ***Planning Time*** | | ***Execution Time*** | |
| **TANPA INDEX** | **DENGAN INDEX** | **TANPA INDEX** | **DENGAN INDEX** |
| **1** | 0.064ms | 0.064ms | 0.086ms | 0.116ms |
| **2** | 0.054ms | 0.054ms | 0.035ms | 0.035ms |
| **3** | 0.073ms | 0.073ms | 0.042ms | 0.041ms |
| **4** | 0.076ms | 0.079ms | 0.182ms | 0.170ms |

**Seharusnya, dengan index akan lebih cepat execution timenya. Namun, di saya relatif sama untuk planning dan execution timenya.**