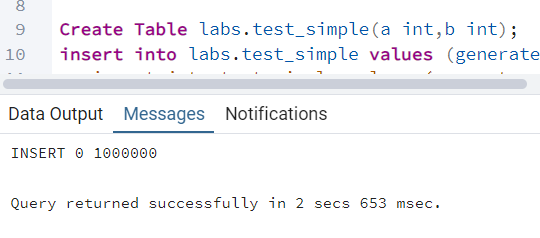
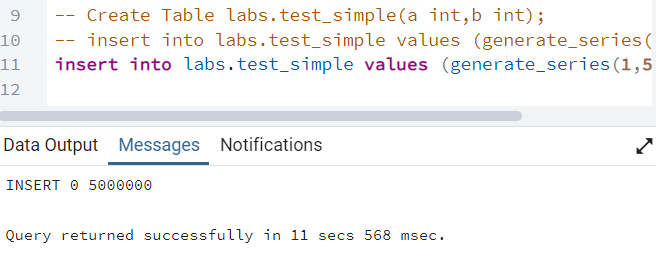
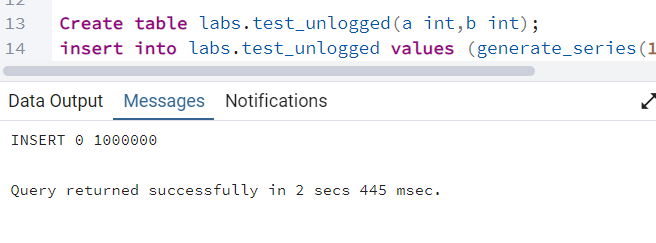
**Task 1.**

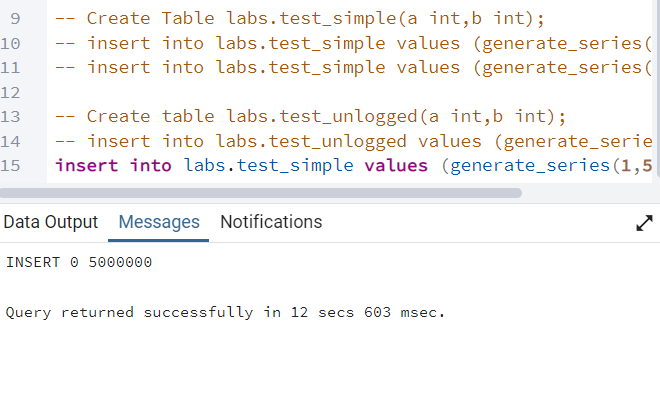
1-2.





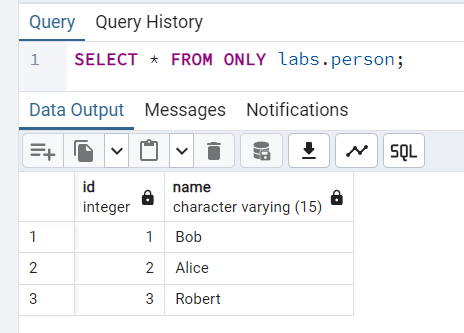
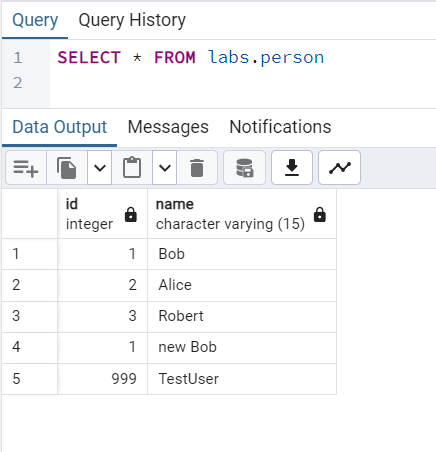
3-4.

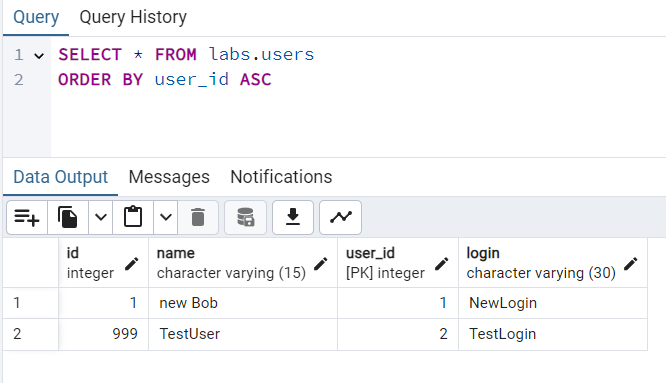




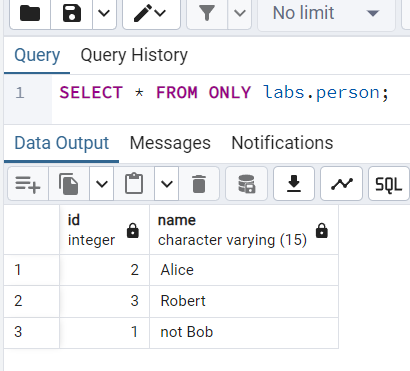
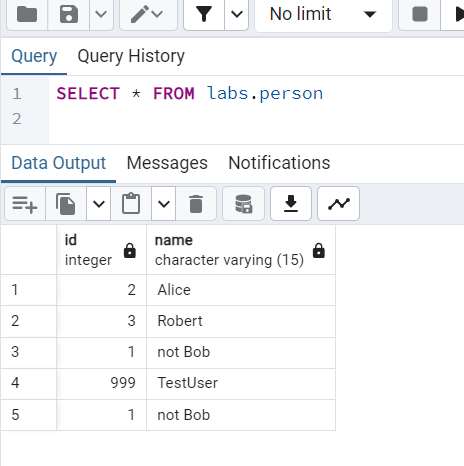
**Task 2.**

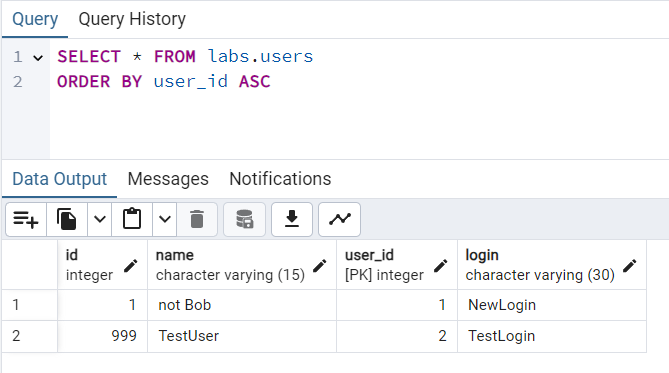
1-3.





4.





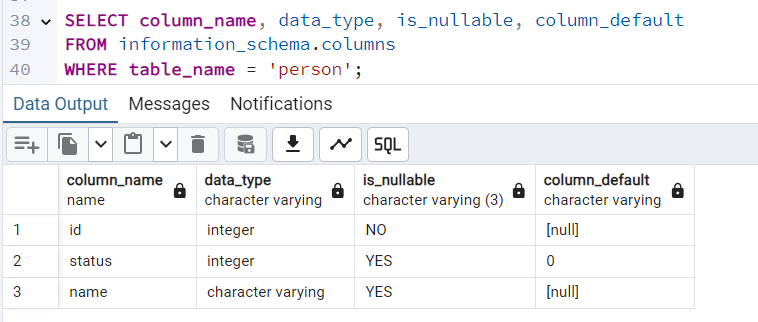
Rewrite UPDATE to change name for row where id=1 for person table only.

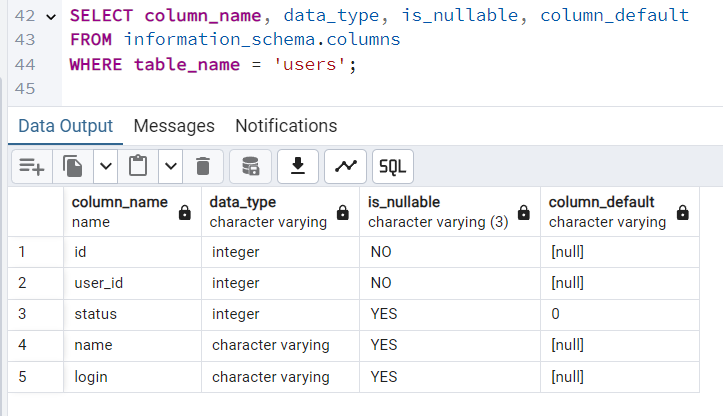
UPDATE ONLY labs.person

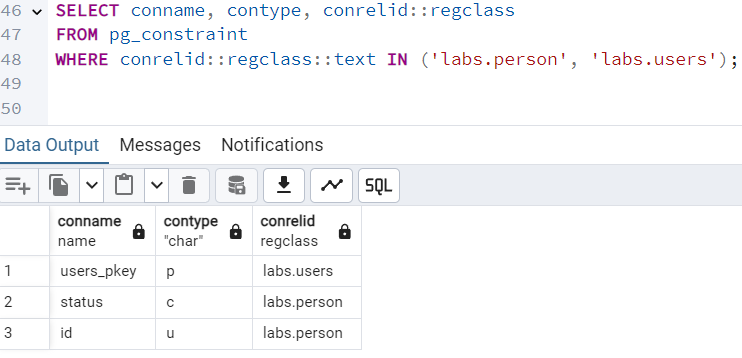
SET name = 'not Bob'

WHERE id = 1;

5.





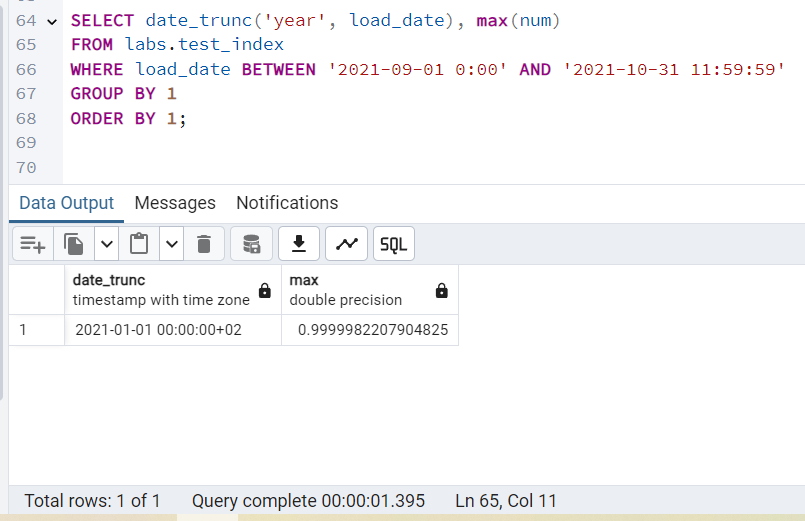


**Task 3.**

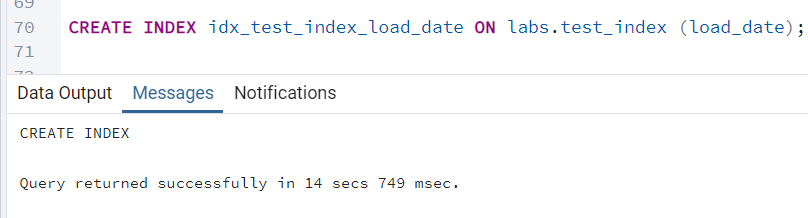
1-2.

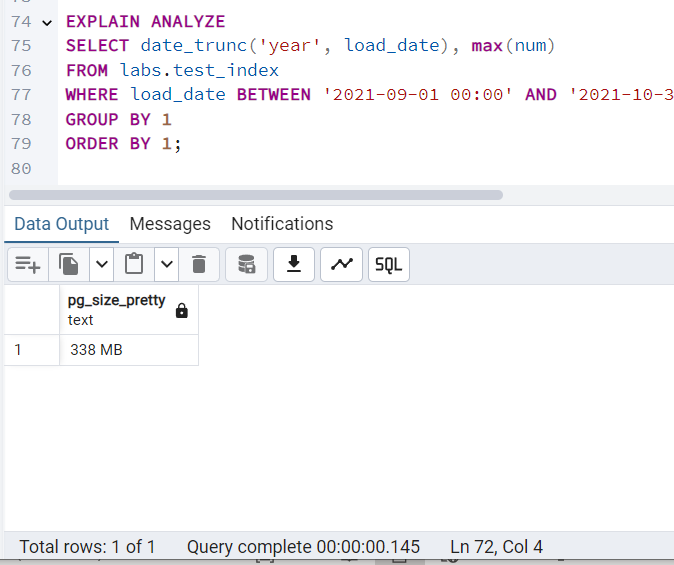






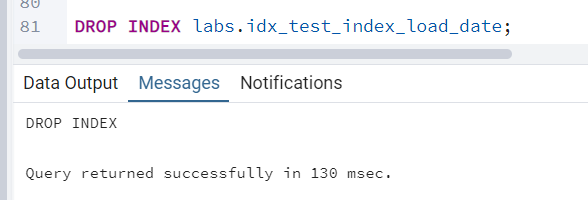
4.



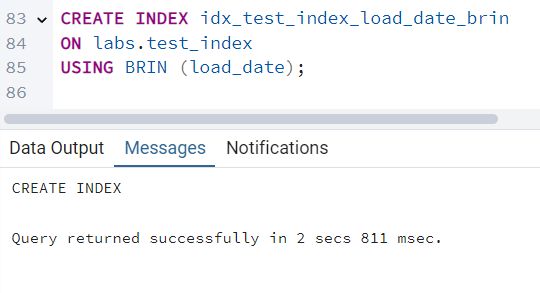


**Improvement:**

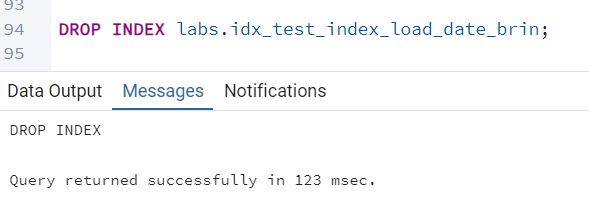
* PostgreSQL now uses an **Index Scan** instead of a **Sequential Scan**.
* Query execution time is **lower**.



5.

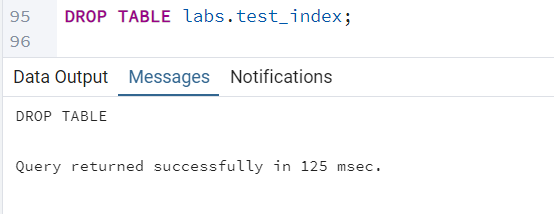




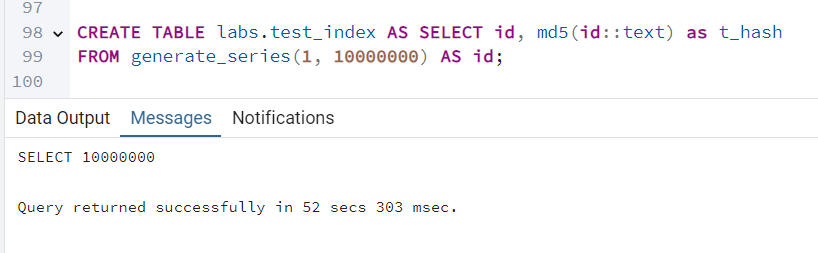


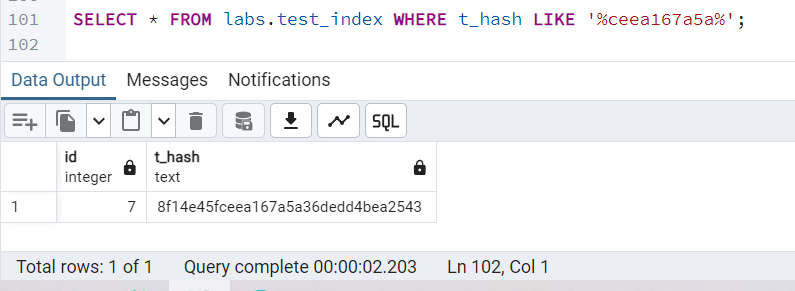
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index Type** | **Creation Time** | **Index Size** | **Query Speed** | **Best Use Case** |
| **B-Tree** | **Slow** (longer to build) | **Larger** | **Fastest** (Index Scan) | General purpose, small-to-medium tables |
| **BRIN** | **Fast** (very quick to create) | **Smaller** | **Faster than Seq Scan** but slower than B-Tree | Large, append-only, **sequential data** |

6.

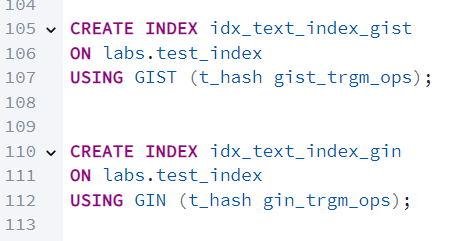


**Task 4.**





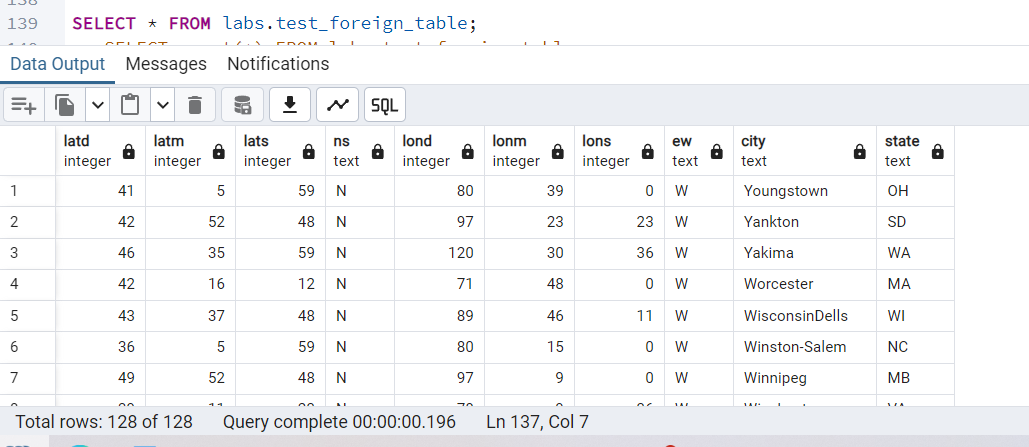
3-5. I was running these queries for 10 minutes and it did not give me results, I assume dataset with 10 M rows my laptop will execute very long time. Then expected result should be next:

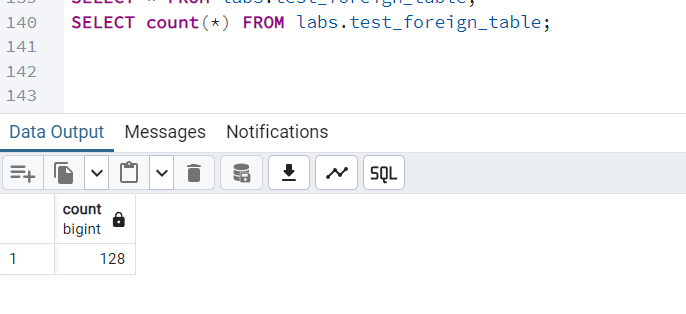


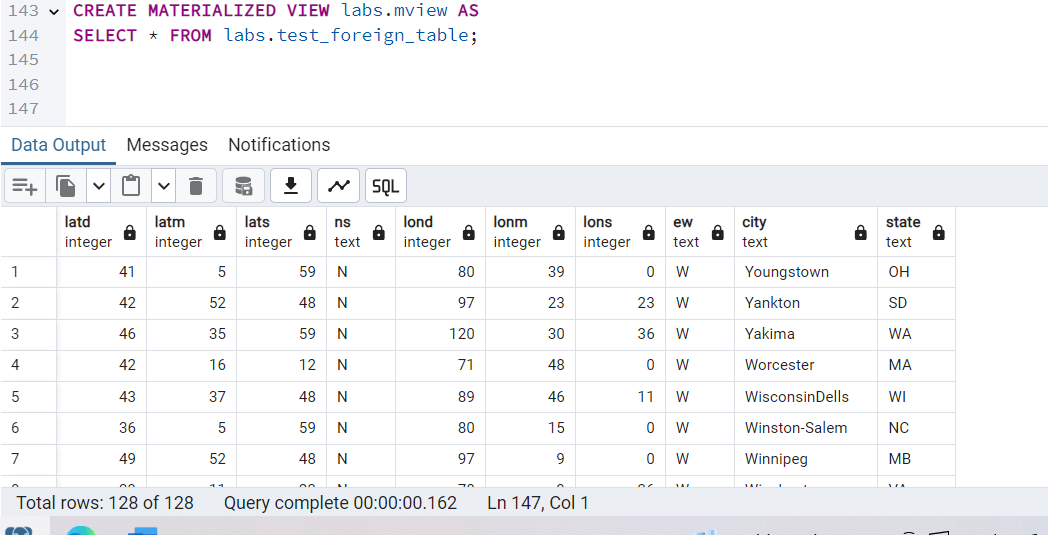
|  |  |
| --- | --- |
| **Index Type** | **Performance for LIKE '%abc%'** |
| No Index | Slow (Sequential Scan) |
| GiST (gist\_trgm\_ops) | Faster, but still scans many rows |
| GIN (gin\_trgm\_ops) | Fastest (Efficient Index Scan) |

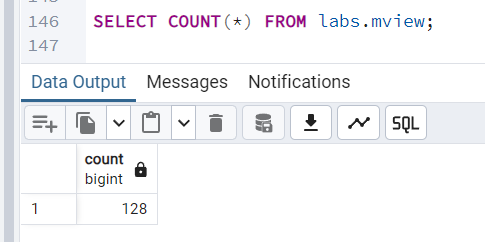
**Task 5.**

1-3.





4. 



5.

