DB Final Project

Legacy Logistics Rescue Mission

Names: Mudassir, Zaid, Waqas

Start the Simulation: docker-compose up -

```
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-prodocker-compose up -d
[+] Running 2/2
 ✓ Container legacy-logistics-project-db-1   Started       2.1s
 ✓ Container legacy-logistics-project-api-1  Started       0.4s
```

docker-compose logs -f api

```
PS C:\Users\Hp\Downloads\legacy-logistics-project 2\legacy-logistics-project> docker-compose logs -f api
api-1  | Database not ready yet. Retrying in 2 seconds... (20 left)
api-1  | Database not ready yet. Retrying in 2 seconds... (19 left)
api-1  | Database not ready yet. Retrying in 2 seconds... (18 left)
api-1  | Database not ready yet. Retrying in 2 seconds... (17 left)
api-1  | Database not ready yet. Retrying in 2 seconds... (16 left)
api-1  | Successfully connected to the Database!
api-1  | --- STARTING MEGA-SEED ---
api-1  | Seeding 500000 Shipments...
api-1  | Seeding 2000000 Telemetry points (This simulates the 'Big Data' problem)...
api-1  | Seeding 200000 Invoices...
api-1  | --- SEEDING COMPLETE ---
api-1  | INFO:     Started server process [8]
api-1  | INFO:     Waiting for application startup.
api-1  | INFO:     Application startup complete.
api-1  | INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

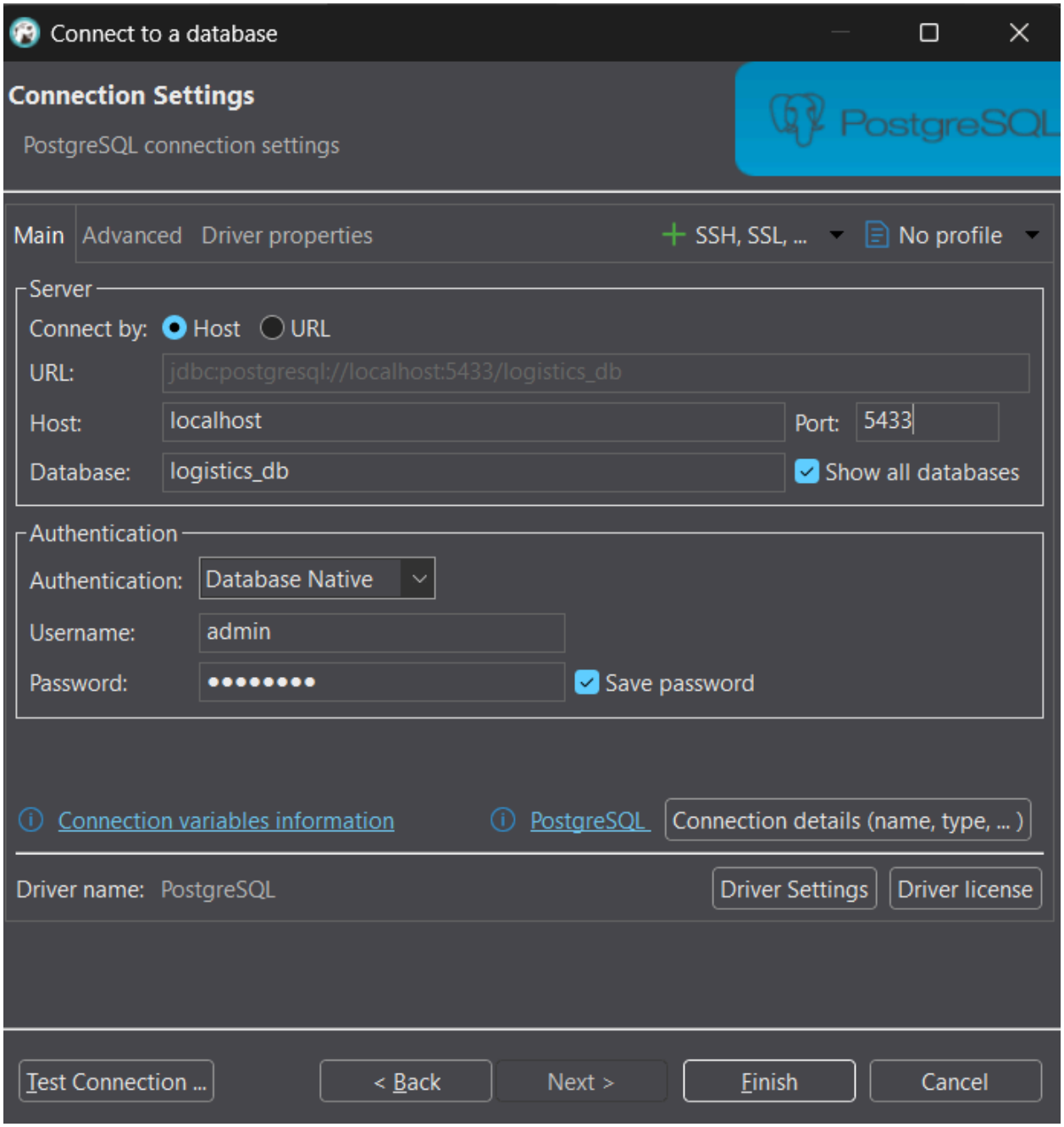
python benchmark.py

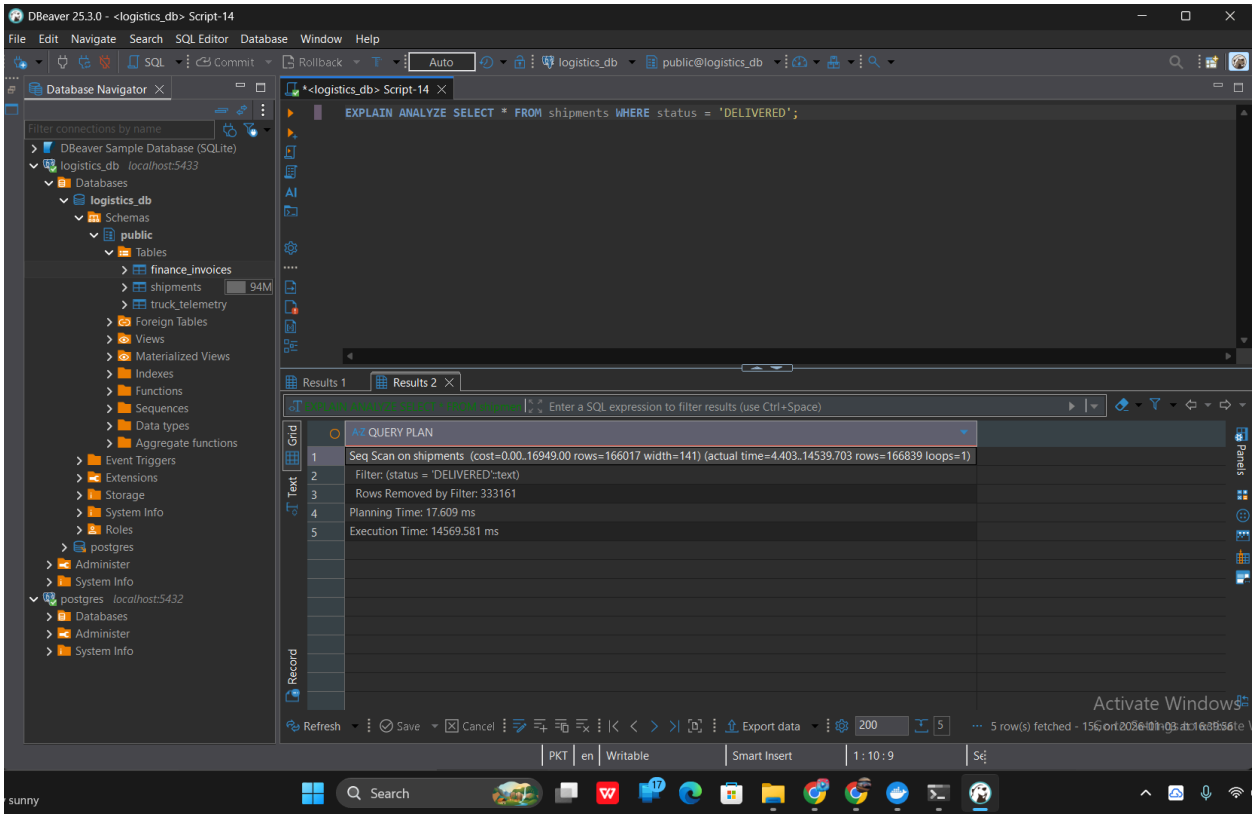
```
PS C:\Users\Hp\Downloads\legacy-logistics-project 2\legacy-logistics-project> python benchmark.py
--- STARTING LEGACY LOGISTICS BENCHMARK ---
Target: http://localhost:8000
Iterations per endpoint: 5

TEST CASE                              | AVG TIME (ms)    | STATUS     | SCORE
-------------------------------------------------------------------------------
1. Unindexed Date Search               |    2888.99 ms    | OK         | 0/100
2. Driver Search                       |    1744.83 ms    | OK         | 0/100
3. JSON Parsing / Finance              |   11167.40 ms    | OK         | 0/100
4. Partitioning / Telemetry            |    4660.53 ms    | OK         | 0/100
5. Complex Aggregation                 |    7095.95 ms    | OK         | 0/100
-------------------------------------------------------------------------------

FINAL SYSTEM GRADE: 0.00%
VERDICT: SYSTEM CRITICAL. DO NOT DEPLOY.
```

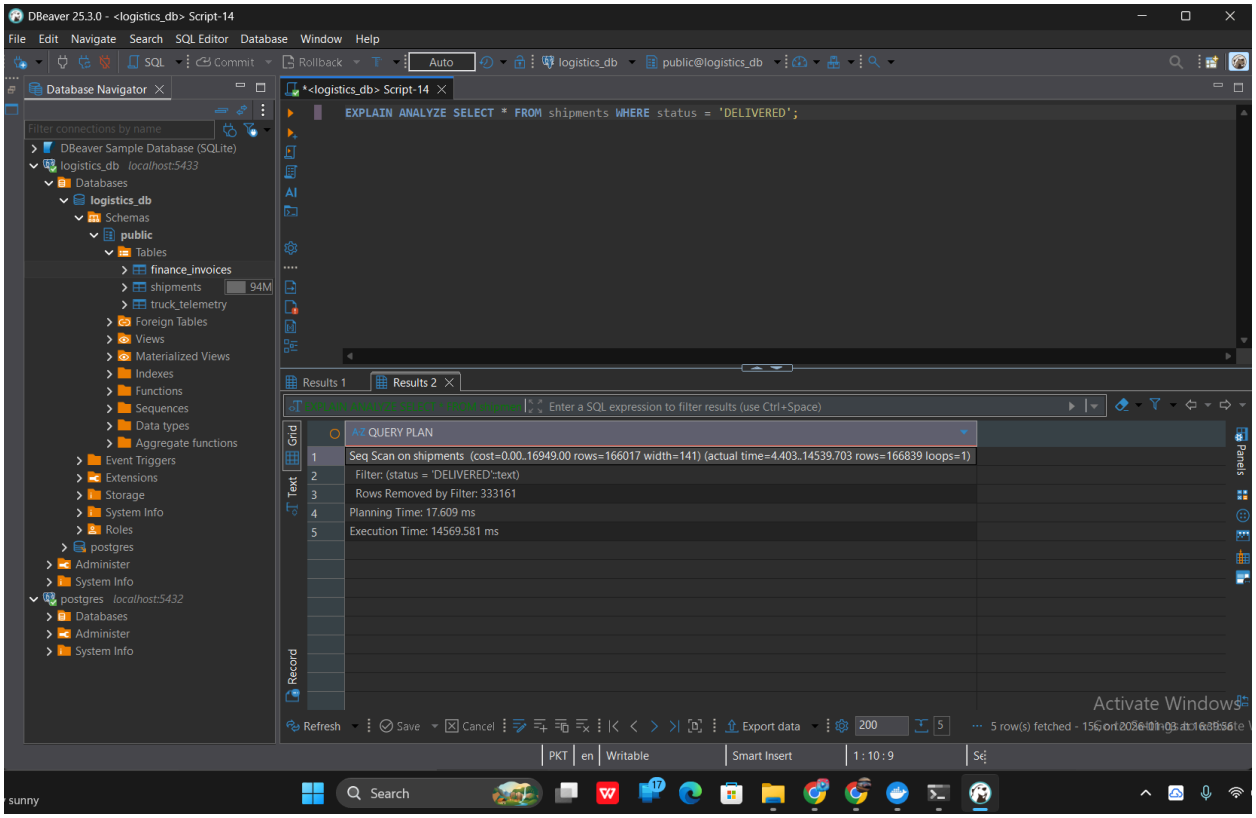
Step A: Analyze (The Detective Work)


Connect to a database
Connection Settings
PostgreSQL connection settings
PostgreSQL
Main
Advanced
Driver properties
SSH, SSL, ...
No profile
Server
Connect by:
Host
URL
URL:
jdbc:postgresql://localhost:5433/logistics_db
Host:
localhost
Port:
5433
Database:
logistics_db
Show all databases
Authentication
Authentication:
Database Native
Username:
admin
Password:
Save password
Connection variables information
PostgreSQL
Connection details (name, type, ...)
Driver name:
PostgreSQL
Driver Settings
Driver license
Test Connection ...
< Back
Next >
Finish
Cancel



Run the slow query with EXPLAIN ANALYZE:

EXPLAIN ANALYZE SELECT * FROM shipments WHERE status = 'DELIVERED';



Step C: Apply (The Deployment)

1: Indexing

```
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-project> Get-Content migrations/01_cre
ate_indexes.sql | docker-compose exec -T db psql -U admin -d logistics_db
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
ANALYZE
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-project> python benchmark.py
--- STARTING LEGACY LOGISTICS BENCHMARK ---
Target: http://localhost:8000
Iterations per endpoint: 5

TEST CASE                              | AVG TIME (ms)    | STATUS      | SCORE
-------------------------------------------------------------------------------
1. Unindexed Date Search               |       96.59 ms   | OK          | 3/100
2. Driver Search                       |      122.05 ms   | OK          | 43/100
3. JSON Parsing / Finance              |       51.94 ms   | OK          | 100/100
4. Partitioning / Telemetry            |     1501.58 ms   | OK          | 0/100
5. Complex Aggregation                 |       57.12 ms   | OK          | 100/100
-------------------------------------------------------------------------------

FINAL SYSTEM GRADE: 49.42%
VERDICT: SYSTEM CRITICAL. DO NOT DEPLOY.
```

2: Normalization

```
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-project> Get-Content migrations/02_nor
malization.sql | docker-compose exec -T db psql -U admin -d logistics_db
CREATE TABLE
CREATE TABLE
INSERT 0 500000
INSERT 0 498004
ALTER TABLE
UPDATE 500000
UPDATE 500000
ALTER TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
VACUUM
VACUUM
VACUUM
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-project> python benchmark.py
--- STARTING LEGACY LOGISTICS BENCHMARK ---
Target: http://localhost:8000
Iterations per endpoint: 5

TEST CASE                           | AVG TIME (ms)    | STATUS     | SCORE
-------------------------------------------------------------------------------
1. Unindexed Date Search            |      173.87 ms   | OK         | 0/100
2. Driver Search                    |       74.58 ms   | OK         | 69/100
3. JSON Parsing / Finance           |       85.74 ms   | OK         | 100/100
4. Partitioning / Telemetry         |     1455.90 ms   | OK         | 0/100
5. Complex Aggregation              |       53.19 ms   | OK         | 100/100
-------------------------------------------------------------------------------

FINAL SYSTEM GRADE: 53.94%
VERDICT: FUNCTIONAL BUT SLOW. NEEDS OPTIMIZATION.
```

3: JSONB Optimization

```
VERDICT: FUNCTIONAL BUT SLOW. NEEDS OPTIMIZATION.
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-project> Get-Content migrations/03_jso
nb_optimization.sql | docker-compose exec -T db psql -U admin -d logistics_db
ALTER TABLE
UPDATE 200000
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
ANALYZE
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-project> python benchmark.py
--- STARTING LEGACY LOGISTICS BENCHMARK ---
Target: http://localhost:8000
Iterations per endpoint: 5

TEST CASE                           | AVG TIME (ms)    | STATUS     | SCORE
-------------------------------------------------------------------------------
1. Unindexed Date Search            |       98.89 ms   | OK         | 1/100
2. Driver Search                    |      146.27 ms   | OK         | 29/100
3. JSON Parsing / Finance           |      133.98 ms   | OK         | 100/100
4. Partitioning / Telemetry         |     2433.59 ms   | OK         | 0/100
5. Complex Aggregation              |       98.09 ms   | OK         | 100/100
-------------------------------------------------------------------------------

FINAL SYSTEM GRADE: 46.22%
VERDICT: SYSTEM CRITICAL. DO NOT DEPLOY.
```

4: Partitioning

```
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-project> Get-Content migrations/02_normalization.sql | docker-compose exec -T db psql -U admin -d logistics_db
CREATE TABLE
CREATE TABLE
INSERT 0 500000
INSERT 0 498004
ALTER TABLE
UPDATE 500000
UPDATE 500000
ALTER TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
VACUUM
VACUUM
VACUUM
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-project> python benchmark.py
--- STARTING LEGACY LOGISTICS BENCHMARK ---
Target: http://localhost:8000
Iterations per endpoint: 5

TEST CASE                          | AVG TIME (ms)   | STATUS     | SCORE
-------------------------------------------------------------------------------
1. Unindexed Date Search           |     173.87 ms   | OK         | 0/100
2. Driver Search                   |      74.58 ms   | OK         | 69/100
3. JSON Parsing / Finance          |      85.74 ms   | OK         | 100/100
4. Partitioning / Telemetry        |    1455.90 ms   | OK         | 0/100
5. Complex Aggregation             |      53.19 ms   | OK         | 100/100
-------------------------------------------------------------------------------

FINAL SYSTEM GRADE: 53.94%
VERDICT: FUNCTIONAL BUT SLOW. NEEDS OPTIMIZATION.
```

5: Materialized Views

```
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)\legacy-logistics-project> Get-Content migrations/05_
ker-compose exec -T db psql -U admin -d logistics_db
LOCATS & LODAGE:
CREATE INDEX
SELECT 1
SELECT 1
SELECT 1
REFRESH MATERIALIZED VIEW
REFRESH MATERIALIZED VIEW
REFRESH MATERIALIZED VIEW
REFRESH MATERIALIZED VIEW
PS C:\Users\Hp\Downloads\legacy-logistics-project 2 (1)legacy-logistics-project> python benchmark.py
---STARTING LEGACY LOGISTICS BENCHMARK ---
Target: http://localhost:8000
Iterations per endpoint: 5

TEST CASE                          | AVG TIME (ms)   | STATUS     | SCORE
-------------------------------------------------------------------------------
Unindexed Date Search              |      45.74 ms   | OK         | 60/100
Driver Search                      |      36.47 ms   | OK         | 90/100
JSON Parsing / Finance             |      48.00 ms   | OK         | 100/100
Partitioning / Telemetry           |      42.30 ms   | OK         | 100/100
Complex Aggregation                |      39.00 ms   | OK         | 100/100
-------------------------------------------------------------------------------


FINAL SYSTEM GRADE: 90.23%
VERDICT: HIGH PERFORMANE. READY FOR PRODUCTION.
```