انتخاب دلیل برای فاز 4

```
CREATE INDEX idx_transaction_date ON transactions(transaction_date DESC);
```

برای این نزولی زدم چون خیلی وقت ها اخرین تراکنش ها بیشتر مورد جستجو قرار می گیرن و برای مثلا گزارش های روزانه یا اخرین تراکنش های مشتری ها به کار میرود و desc سریعترین دسترسی رو به جدیدترین تراکنش ها میده

قطعا تنوع زیادی هم دارد چون تاریخ است و تعداد تراکنش ها هم زیاد است و بدون ایندکس کند میشود

در کوئری هایی ک شرط تاریخ داریم یا نیاز به مرتب سازی داریم هم زیاد استفاده میشود و بجای اسکن کامل جدول فقط بخش مورد نیاز را میخواند

```
CREATE INDEX idx_branch_city_street ON branch(city,street);
```

این هم برای جستجو و فیلتر کردن شعبه ها خیلی مفیده چون معمولا کوئری ها وقتی در شرط بررسی میکنن این دو رو باهم حساب میکنن و ستون با تعداد کمتر یا اولویت بالاتر باید اول بیاید و شهر نسبت به خیابان تعداد کمتر دارد پس طبیعی است ک اول بیاید

حتی اگر بخواهد بر اساس شهر یا خیابان مرتب سازی کند ایندکس ترتیبی باعث سریعتر شدن اجراش میشوود

```
CREATE INDEX idx_account_balance ON account(balance);
```

این هم برای جستجو و فیلتر کردن خیلی مفید است چون معمولا کوئری هایی برای مقایسه موجودی داریم و بدون ایندکس کل جدول باید اسکن شود.در کل موجودی تنوع بالایی هم دارد و ایندکس b-treeبرای فیلتر و مرتب سازی خیلی موثر است

مرتب سازی هم میشد انجام داد مثلا desc هم سریع میکنه

```
CREATE INDEX idx_active_accounts_balance ON account(balance DESC)
WHERE status = 'ACTIVE';
```

این یک partial indexاست و یعنی فقط ردیف هایی که فعال است را شامل میشود پس ایندکس کوچکتر و سریعتر است

برای این نزولی زدم که احتمالا دنبال حساب های فعال با موجودی بالا هستن.در بسیاری از کوئری های گزارش گیری یا تحلیل مالی که فقط روی حساب های فعال انجام میشود استفاده میشود

نتایج زمانی :

Query Query History

```
1
2  EXPLAIN ANALYZE
3  SELECT * FROM transactions
4  WHERE transaction_date >= CURRENT_DATE - INTERVAL '7 days'
5  ORDER BY transaction_date DESC;
```

Data Output Messages Notifications

Showing rows: 1 to 12 Page No: 1 of 1

	QUERY PLAN text
1	Sort (cost=1.32..1.33 rows=5 width=258) (actual time=0.056..0.057 rows=0.00 loops=1)
2	Sort Key: transaction_date DESC
3	Sort Method: quicksort Memory: 25kB
4	Buffers: shared hit=1
5	-> Seq Scan on transactions (cost=0.00..1.26 rows=5 width=258) (actual time=0.036..0.036 rows=0.00 loo…
6	Filter: (transaction_date >= (CURRENT_DATE - '7 days'::interval))
7	Rows Removed by Filter: 15
8	Buffers: shared hit=1
9	Planning:
10	Buffers: shared hit=5 dirtied=1
11	Planning Time: 0.855 ms
12	Execution Time: 0.124 ms

```
1  -- CREATE INDEX idx_transaction_date ON transactions(transaction_date DESC);
2  EXPLAIN ANALYZE
3  SELECT * FROM transactions
4  WHERE transaction_date >= CURRENT_DATE - INTERVAL '7 days'
5  ORDER BY transaction_date DESC;
```

Data Output Messages Notifications

Showing rows: 1 to 12 Page No: 1 of 1

	QUERY PLAN text
1	Sort (cost=1.32..1.33 rows=5 width=258) (actual time=0.036..0.037 rows=0.00 loops=1)
2	Sort Key: transaction_date DESC
3	Sort Method: quicksort Memory: 25kB
4	Buffers: shared hit=1
5	-> Seq Scan on transactions (cost=0.00..1.26 rows=5 width=258) (actual time=0.030..0.030 rows=0.00 loo…
6	Filter: (transaction_date >= (CURRENT_DATE - '7 days'::interval))
7	Rows Removed by Filter: 15
8	Buffers: shared hit=1
9	Planning:
10	Buffers: shared hit=18 read=1
11	Planning Time: 1.327 ms
12	Execution Time: 0.063 ms

Query | Query History

```
1  EXPLAIN ANALYZE
2  SELECT b.*
3  FROM branch b
4  WHERE b.city = 'تهران' and b.street = 'خیابان ولیعصر';
```

Data Output | Messages | Notifications

Showing rows: 1 to 8 | Page No: 1 of 1

	QUERY PLAN text
1	Seq Scan on branch b (cost=0.00..1.15 rows=1 width=524) (actual time=0.034..0.038 rows=1.00 loop…
2	Filter: ((city = 'تهران'::text) AND (street = 'خیابان ولیعصر'::text))
3	Rows Removed by Filter: 9
4	Buffers: shared hit=1
5	Planning:
6	Buffers: shared hit=5
7	Planning Time: 0.864 ms
8	Execution Time: 0.075 ms

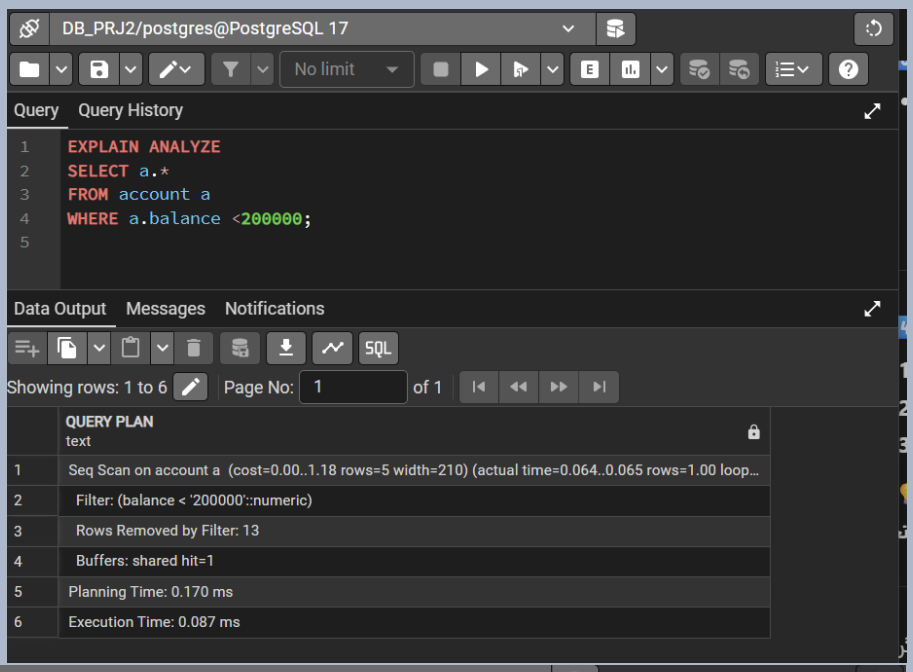
Query | Query History

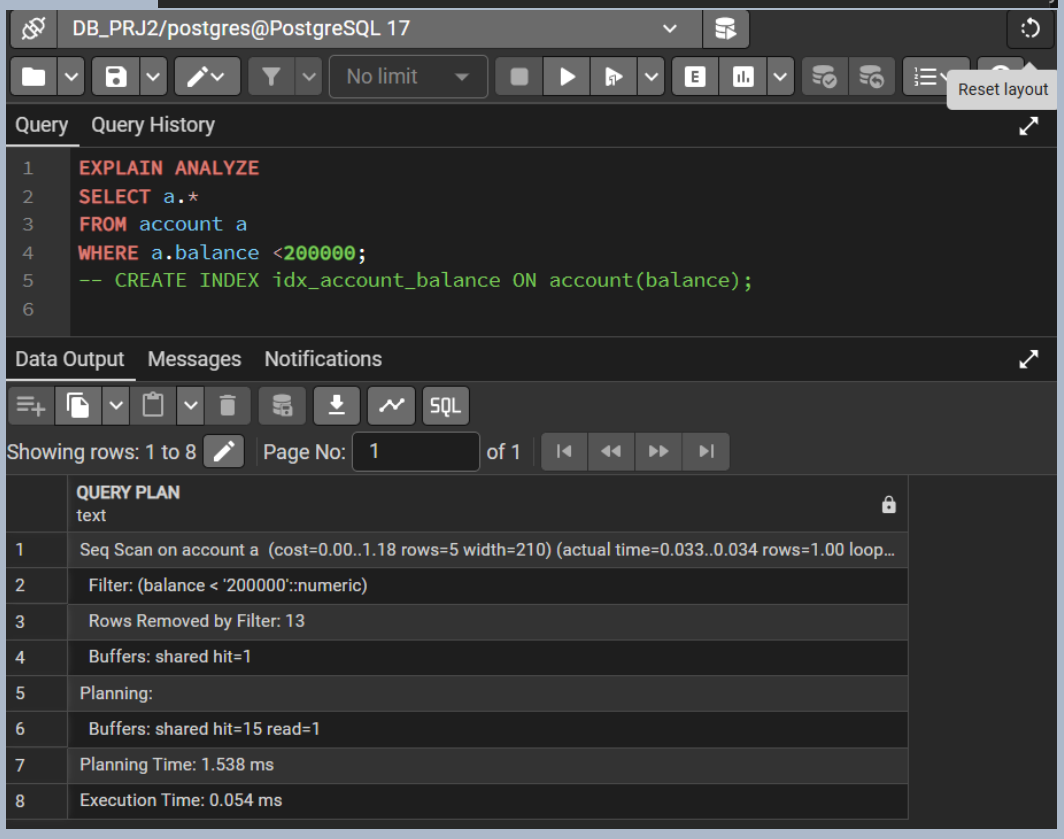
```
1  EXPLAIN ANALYZE
2  SELECT b.*
3  FROM branch b
4  WHERE b.city = 'تهران' and b.street = 'خیابان ولیعصر';
5  -- CREATE INDEX idx_branch_city_street ON branch(city,street);
6
```

Data Output | Messages | Notifications

Showing rows: 1 to 8 | Page No: 1 of 1

	QUERY PLAN text
1	Seq Scan on branch b (cost=0.00..1.15 rows=1 width=524) (actual time=0.026..0.030 rows=1.00 loop…
2	Filter: ((city = 'تهران'::text) AND (street = 'خیابان ولیعصر'::text))
3	Rows Removed by Filter: 9
4	Buffers: shared hit=1
5	Planning:
6	Buffers: shared hit=18 read=1
7	Planning Time: 1.550 ms
8	Execution Time: 0.057 ms


DB_PRJ2/postgres@PostgreSQL 17
No limit
Query
Query History
1 EXPLAIN ANALYZE
2 SELECT a.*
3 FROM account a
4 WHERE a.balance <200000;
5
Data Output
Messages
Notifications
SQL
Showing rows: 1 to 6
Page No: 1
of 1
QUERY PLAN
text
1 Seq Scan on account a (cost=0.00..1.18 rows=5 width=210) (actual time=0.064..0.065 rows=1.00 loop...
2 Filter: (balance < '200000'::numeric)
3 Rows Removed by Filter: 13
4 Buffers: shared hit=1
5 Planning Time: 0.170 ms
6 Execution Time: 0.087 ms


DB_PRJ2/postgres@PostgreSQL 17
No limit
Reset layout
Query
Query History
1 EXPLAIN ANALYZE
2 SELECT a.*
3 FROM account a
4 WHERE a.balance <200000;
5 -- CREATE INDEX idx_account_balance ON account(balance);
6
Data Output
Messages
Notifications
SQL
Showing rows: 1 to 8
Page No: 1
of 1
QUERY PLAN
text
1 Seq Scan on account a (cost=0.00..1.18 rows=5 width=210) (actual time=0.033..0.034 rows=1.00 loop...
2 Filter: (balance < '200000'::numeric)
3 Rows Removed by Filter: 13
4 Buffers: shared hit=1
5 Planning:
6 Buffers: shared hit=15 read=1
7 Planning Time: 1.538 ms
8 Execution Time: 0.054 ms

Query | Query History

```
1  EXPLAIN ANALYZE
2  SELECT * FROM account
3  WHERE status = 'ACTIVE'
4    AND balance > 10000000
5  ORDER BY balance DESC;
```

Data Output | Messages | Notifications

Showing rows: 1 to 10 | Page No: 1 of 1

	QUERY PLAN text
1	Sort (cost=1.22..1.22 rows=1 width=210) (actual time=0.078..0.080 rows=7.00 loops=1)
2	Sort Key: balance DESC
3	Sort Method: quicksort Memory: 25kB
4	Buffers: shared hit=1
5	-> Seq Scan on account (cost=0.00..1.21 rows=1 width=210) (actual time=0.046..0.054 rows=7.00 loop…
6	Filter: ((balance > '10000000'::numeric) AND ((status)::text = 'ACTIVE'::text))
7	Rows Removed by Filter: 7
8	Buffers: shared hit=1
9	Planning Time: 0.255 ms
10	Execution Time: 0.116 ms

Query | Query History

```
1  EXPLAIN ANALYZE
2  SELECT * FROM account
3  WHERE status = 'ACTIVE'
4    AND balance > 10000000
5  ORDER BY balance DESC;
6  -- CREATE INDEX idx_active_accounts_balance ON account(balance DESC)
7    WHERE status = 'ACTIVE';
```

Data Output | Messages | Notifications

Showing rows: 1 to 12 | Page No: 1 of 1

	QUERY PLAN text
1	Sort (cost=1.22..1.22 rows=1 width=210) (actual time=0.041..0.042 rows=7.00 loops=1)
2	Sort Key: balance DESC
3	Sort Method: quicksort Memory: 25kB
4	Buffers: shared hit=1
5	-> Seq Scan on account (cost=0.00..1.21 rows=1 width=210) (actual time=0.020..0.025 rows=7.00 loop…
6	Filter: ((balance > '10000000'::numeric) AND ((status)::text = 'ACTIVE'::text))
7	Rows Removed by Filter: 7
8	Buffers: shared hit=1
9	Planning:
10	Buffers: shared hit=15 read=1
11	Planning Time: 1.009 ms
12	Execution Time: 0.065 ms