ex1:
1.
without indexes:

```
1   explain analyze select count(id) from customer;
```

Data Output   Explain   Messages   Notifications

	QUERY PLAN text	🔒
1	Aggregate (cost=3307.12..3307.13 rows=1 width=8) (actual time=20.033..20.034 rows=1 loops=1)	
2	[...] -> Index Only Scan using customer_pkey on customer (cost=0.29..3057.12 rows=100000 width=4)...	
3	[...] Heap Fetches: 14754	
4	Planning Time: 0.119 ms	
5	Execution Time: 20.082 ms	

b_tree id index:

```
1   explain analyze select count(id) from customer;
```

Data Output   Explain   Messages   Notifications

	QUERY PLAN text	🔒
1	Aggregate (cost=3307.12..3307.13 rows=1 width=8) (actual time=22.258..22.260 rows=1 loops=1)	
2	[...] -> Index Only Scan using id_ind on customer (cost=0.29..3057.12 rows=100000 width=4) (actual t...	
3	[...] Heap Fetches: 14754	
4	Planning Time: 0.156 ms	
5	Execution Time: 22.320 ms	

hash id index:

```
1    explain analyze select count(id) from customer;
```

Data Output   Explain   Messages   Notifications

	QUERY PLAN text
1	Aggregate (cost=3307.12..3307.13 rows=1 width=8) (actual time=21.347..21.348 rows=1 loops=1)
2	[...] -> Index Only Scan using customer_pkey on customer (cost=0.29..3057.12 rows=100000 width=4)...
3	[...] Heap Fetches: 14754
4	Planning Time: 0.132 ms
5	Execution Time: 21.394 ms

Cost does not change in this example.
2.
without indexes:

```
1    explain analyze select id from customer where id = 4 or id = 47;
```

Data Output   Explain   Messages   Notifications

	QUERY PLAN text
1	Bitmap Heap Scan on customer (cost=8.60..16.48 rows=2 width=4) (actual time=0.016..0.018 rows=2 loops=1)
2	[...] Recheck Cond: ((id = 4) OR (id = 47))
3	[...] Heap Blocks: exact=2
4	[...] -> BitmapOr (cost=8.60..8.60 rows=2 width=0) (actual time=0.012..0.012 rows=0 loops=1)
5	[...] -> Bitmap Index Scan on customer_pkey (cost=0.00..4.30 rows=1 width=0) (actual time=0.009..0.009 row...
6	[...] Index Cond: (id = 4)
7	[...] -> Bitmap Index Scan on customer_pkey (cost=0.00..4.30 rows=1 width=0) (actual time=0.003..0.003 row...
8	[...] Index Cond: (id = 47)
9	Planning Time: 0.067 ms
10	Execution Time: 0.037 ms

b_tree name index:

```
1    explain analyze select id from customer where id = 4 or id = 47;
```

Data Output   Explain   Messages   Notifications

	QUERY PLAN text
1	Bitmap Heap Scan on customer (cost=8.60..16.48 rows=2 width=4) (actual time=0.040..0.042 rows=2 loops=1)
2	[...] Recheck Cond: ((id = 15235) OR (id = 35665))
3	[...] Heap Blocks: exact=2
4	[...] -> BitmapOr (cost=8.60..8.60 rows=2 width=0) (actual time=0.035..0.036 rows=0 loops=1)
5	[...] -> Bitmap Index Scan on idind (cost=0.00..4.30 rows=1 width=0) (actual time=0.028..0.028 rows=1 loops=...
6	[...] Index Cond: (id = 15235)
7	[...] -> Bitmap Index Scan on idind (cost=0.00..4.30 rows=1 width=0) (actual time=0.006..0.006 rows=1 loops=...
8	[...] Index Cond: (id = 35665)
9	Planning Time: 0.130 ms
10	Execution Time: 0.069 ms

hash name index:

```
1    explain analyze select id from customer where id = 4 or id = 47;
```

Data Output   Explain   Messages   Notifications

	QUERY PLAN text
1	Bitmap Heap Scan on customer (cost=8.02..15.89 rows=2 width=4) (actual time=0.595..0.597 rows=2 loops=1)
2	[...] Recheck Cond: ((id = 4) OR (id = 47))
3	[...] Heap Blocks: exact=2
4	[...] -> BitmapOr (cost=8.02..8.02 rows=2 width=0) (actual time=0.583..0.583 rows=0 loops=1)
5	[...] -> Bitmap Index Scan on idind (cost=0.00..4.01 rows=1 width=0) (actual time=0.566..0.567 rows=1 loops=...
6	[...] Index Cond: (id = 4)
7	[...] -> Bitmap Index Scan on idind (cost=0.00..4.01 rows=1 width=0) (actual time=0.013..0.013 rows=1 loops=...
8	[...] Index Cond: (id = 47)
9	Planning Time: 0.921 ms
10	Execution Time: 0.642 ms

Cost without indexes is the same as the cost with b_tree index, but cost with hash index is less than the others.
3.
without indexes:

```
1   explain analyze select id from customer where address in ('Innopolis');
```

Data Output   Explain   Messages   Notifications

	QUERY PLAN text
1	Seq Scan on customer (cost=0.00..4283.00 rows=1 width=4) (actual time=12.909..12.910 rows=0 loops=1)
2	[...] Filter: (address = 'Innopolis'::text)
3	[...] Rows Removed by Filter: 100000
4	Planning Time: 0.521 ms
5	Execution Time: 12.932 ms

with b_tree index:

```
1   explain analyze select id from customer where address in ('Innopolis');
```

Data Output   Explain   Messages   Notifications

	QUERY PLAN text
1	Index Scan using adind on customer (cost=0.42..8.44 rows=1 width=4) (actual time=0.123..0.124 rows=0 loop...
2	[...] Index Cond: (address = 'Innopolis'::text)
3	Planning Time: 0.420 ms
4	Execution Time: 0.135 ms

with hash index:

```
1   explain analyze select id from customer where address in ('Innopolis');
```

Data Output   Explain   Messages   Notifications

	QUERY PLAN text
1	Index Scan using adind on customer (cost=0.00..8.02 rows=1 width=4) (actual time=0.009..0.010 rows=0 loop...
2	[...] Index Cond: (address = 'Innopolis'::text)
3	Planning Time: 0.453 ms
4	Execution Time: 0.022 ms

Cost with b_tree index is bigger than the cost with hash index and is much less than the cost without indexes.