Создание таблицы и вставка значений

```
CREATE TABLE products (
  id integer,
  created_at timestamp NOT NULL,
  active boolean NOT NULL,
  sort integer NOT NULL,
  price numeric(10,2) NOT NULL,
  code char(20) NOT NULL,
  name char(100) NOT NULL,
  description char(255) NOT NULL
);

INSERT INTO products (id, created_at, active, sort, price, code, name,
description)
SELECT
  gs AS id,
  timestamp '2016-01-01' + (random() * (365*8) * interval '1 day') AS
created_at,
  (random() < 0.70) AS active,
  (floor(random()*10000))::int AS sort,
  round((random()*10000)::numeric, 2) AS price,
  (array[
    'electronics','books','clothes','tools','sport',
    'beauty','games','home','kids','auto'
  ])[floor(random()*10)+1]::char(20) AS code,
  left(md5(random()::text), 60)::char(100) AS name,
  left(md5(random()::text) || md5(random()::text), 200)::char(255) AS
description
FROM generate_series(1, 10000) AS gs;
```

Запрос №1

```
-- EXPLAIN (ANALYZE, BUFFERS)
SELECT * FROM products
WHERE id = 765432;
```

Data Output | Messages | Notifications

	id integer	created_at timestamp without time zone	active boolean	sort integer	price numeric (10,2)	code character (20)	name character (100)	description character (255)
1	765432	2021-09-17 17:39:05.408671	false	271	8842.69	auto	3e53128c27b4c489184903b53d48e3...	7ebccdb19ed5d81bd11695dc75...

```
EXPLAIN (ANALYZE, BUFFERS)
SELECT * FROM products
WHERE id = 765432;
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Gather (cost=1000.00..61764.47 rows=1 width=404) (actual time=65.162..77.545 rows=1 loops=1)
2	Workers Planned: 2
3	Workers Launched: 2
4	Buffers: shared hit=16118 read=39438
5	-> Parallel Seq Scan on products (cost=0.00..60764.38 rows=1 width=404) (actual time=51.778..54.461 rows=0 loops=3)
6	Filter: (id = 765432)
7	Rows Removed by Filter: 333333
8	Buffers: shared hit=16118 read=39438
9	Planning:
10	Buffers: shared hit=30 dirtied=3
11	Planning Time: 0.379 ms
12	Execution Time: 77.559 ms

B-tree index (primary key)

ALTER TABLE products
ADD CONSTRAINT products_pkey PRIMARY KEY (id);

```
1  EXPLAIN (ANALYZE, BUFFERS)
2  SELECT * FROM products
3  WHERE id = 765432;
4
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Index Scan using products_pkey on products (cost=0.42..8.44 rows=1 width=404) (actual time=0.024..0.024 rows=1 loops=1)
2	Index Cond: (id = 765432)
3	Buffers: shared hit=1 read=3
4	Planning:
5	Buffers: shared hit=15 read=1
6	Planning Time: 0.791 ms
7	Execution Time: 0.033 ms

Запрос №2

```
1  -- EXPLAIN (ANALYZE, BUFFERS)
2  SELECT id, code, name, price
3  FROM products
4  WHERE active = true
5  ORDER BY sort
6  LIMIT 100
7
```

Data Output | Messages | Notifications

	id [PK] integer	code character (20)	name character (100)	price numeric (10,2)
1	35018	kids	e1df00e82e8db26880b4ca4488fbcd3...	8046.82
2	27970	home	c823f83daab9c86507f167f624b59ea...	9579.26
3	553339	clothes	8c6b4ebc67a2464a3cd675f83ec1d2b...	9011.02
4	13142	tools	202c91d85a63382601a4180cfe097bf...	3197.09
5	815662	sport	3115938da0312e04cda830d4f2c1b14...	931.85

```
EXPLAIN (ANALYZE, BUFFERS)
SELECT id, code, name, price
FROM products
WHERE active = true
ORDER BY sort
LIMIT 100
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=71808.38..71820.05 rows=100 width=136) (actual time=127.013..131.447 rows=100 loops=1)
2	Buffers: shared hit=16232 read=39436
3	-> Gather Merge (cost=71808.38..139492.61 rows=580110 width=136) (actual time=127.012..131.442 rows=100 loops=1)
4	Workers Planned: 2
5	Workers Launched: 2
6	Buffers: shared hit=16232 read=39436
7	-> Sort (cost=70808.36..71533.50 rows=290055 width=136) (actual time=107.505..107.508 rows=85 loops=3)
8	Sort Key: sort
9	Sort Method: top-N heapsort Memory: 69kB
10	Buffers: shared hit=16232 read=39436
11	Worker 0: Sort Method: top-N heapsort Memory: 68kB
12	Worker 1: Sort Method: top-N heapsort Memory: 68kB
13	-> Parallel Seq Scan on products (cost=0.00..59722.67 rows=290055 width=136) (actual time=0.124..81.807 rows=233396 loops=3)
14	Filter: active
15	Rows Removed by Filter: 99937
16	Buffers: shared hit=16125 read=39431
17	Planning Time: 0.075 ms
18	Execution Time: 131.469 ms

Частичный индекс

CREATE INDEX idx_products_active_sort ON products (sort) WHERE active = true;

```
EXPLAIN (ANALYZE, BUFFERS)
SELECT id, code, name, price
FROM products
WHERE active = true
ORDER BY sort
LIMIT 100
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=0.42..34.19 rows=100 width=136) (actual time=0.017..0.077 rows=100 loops=1)
2	Buffers: shared hit=103
3	-> Index Scan using idx_products_active_sort on products (cost=0.42..235041.13 rows=696133 width=136) (actual time=0.016..0.071 rows=100 loops=1)
4	Buffers: shared hit=103
5	Planning Time: 0.066 ms
6	Execution Time: 0.088 ms

Запрос №3

```sql
-- EXPLAIN (ANALYZE, BUFFERS)
SELECT id, code, name, price
FROM products
WHERE active = true
ORDER BY price DESC
LIMIT 100
```

Data Output | Messages | Notifications

	id [PK] integer	code character (20)	name character (100)	price numeric (10,2)
1	18445	beauty	e2eaf985b4c7d8e96300484e16f166...	10000.00
2	424868	tools	91c459b639634b94045ddcfc13546a...	9999.97
3	384558	books	568f63831292eefc21fb3e922de7b56...	9999.95
4	528362	clothes	36f95f31929a42e628921e5ce1c278...	9999.95
5	148632	auto	d94e2ae3c87b144260164e51290d81...	9999.95

```sql
EXPLAIN (ANALYZE, BUFFERS)
SELECT id, code, name, price
FROM products
WHERE active = true
ORDER BY price DESC
LIMIT 100
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=71808.38..71820.05 rows=100 width=132) (actual time=111.748..116.244 rows=100 loops=1)
2	Buffers: shared hit=16258 read=39410
3	-> Gather Merge (cost=71808.38..139492.61 rows=580110 width=132) (actual time=111.747..116.239 rows=100 loops=1)
4	Workers Planned: 2
5	Workers Launched: 2
6	Buffers: shared hit=16258 read=39410
7	-> Sort (cost=70808.36..71533.50 rows=290055 width=132) (actual time=92.256..92.262 rows=84 loops=3)
8	Sort Key: price DESC
9	Sort Method: top-N heapsort Memory: 71kB
10	Buffers: shared hit=16258 read=39410
11	Worker 0: Sort Method: top-N heapsort Memory: 68kB
12	Worker 1: Sort Method: top-N heapsort Memory: 68kB
13	-> Parallel Seq Scan on products (cost=0.00..59722.67 rows=290055 width=132) (actual time=0.194..62.533 rows=233396 loops=3)
14	Filter: active
15	Rows Removed by Filter: 99937
16	Buffers: shared hit=16146 read=39410
17	Planning Time: 0.062 ms
18	Execution Time: 116.276 ms

Частичный индекс

CREATE INDEX idx_products_active_price ON products (price DESC) WHERE active = true;

```
EXPLAIN (ANALYZE, BUFFERS)
SELECT id, code, name, price
FROM products
WHERE active = true
ORDER BY price DESC
LIMIT 100
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=0.42..34.95 rows=100 width=132) (actual time=0.038..0.443 rows=100 loops=1)
2	Buffers: shared hit=23 read=80
3	-> Index Scan using idx_products_active_price on products (cost=0.42..240316.98 rows=696133 width=132) (actual time=0.038..0.435 rows=100 loops=1)
4	Buffers: shared hit=23 read=80
5	Planning:
6	Buffers: shared hit=15 read=1
7	Planning Time: 0.718 ms
8	Execution Time: 0.453 ms

Запрос №4

```
-- EXPLAIN (ANALYZE, BUFFERS)
SELECT id, code, name, price
FROM products
WHERE active = true and id in (1,500,70000,800000,10230,345234)
ORDER BY sort
LIMIT 100
```

Data Output | Messages | Notifications

	id [PK] integer	code character (20)	name character (100)	price numeric (10,2)
1	10230	home	4552d1f9368fccfd52118ae71fe17c6...	4118.68
2	800000	clothes	f3f5dccd088168228a4c3b63183cd3...	1463.94
3	1	tools	af77991928106d769a41becc3056cd...	25.52
4	70000	home	06d5eafcdc7ae062d117adbd3dcc96...	3017.89
5	345234	sport	05ef122b802d1a1ea045a0cd80bafb...	5897.75

```
1  EXPLAIN (ANALYZE, BUFFERS)
2  SELECT id, code, name, price
3  FROM products
4  WHERE active = true and id in (1,500,70000,800000,10230,345234)
5  ORDER BY sort
6  LIMIT 100
7
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=30.70..30.71 rows=4 width=136) (actual time=0.048..0.049 rows=5 loops=1)
2	Buffers: shared hit=24
3	-> Sort (cost=30.70..30.71 rows=4 width=136) (actual time=0.048..0.048 rows=5 loops=1)
4	Sort Key: sort
5	Sort Method: quicksort Memory: 26kB
6	Buffers: shared hit=24
7	-> Index Scan using products_pkey on products (cost=0.42..30.66 rows=4 width=136) (actual time=0.015..0.038 rows=5 loops=1)
8	Index Cond: (id = ANY ('{1,500,70000,800000,10230,345234}'::integer[]))
9	Filter: active
10	Rows Removed by Filter: 1
11	Buffers: shared hit=24
12	Planning Time: 0.083 ms
13	Execution Time: 0.061 ms

Дополнительные индексы не требуются

Запрос №5

```
1  -- EXPLAIN (ANALYZE, BUFFERS)
2  SELECT id, code, name, price
3  FROM products
4  WHERE active = true and name LIKE '1f%'
5  LIMIT 100
6
```

Data Output | Messages | Notifications

	id [PK] integer	code character (20)	name character (100)	price numeric (10,2)
1	45179	auto	1fa8a0afed682973e42323a0900097...	4299.49
2	45663	clothes	1fc2ac330f0b8571e973008a37c881...	9784.11
3	46913	home	1fdfe5341d4af5c4d8b43af1f41a1f5...	2840.55
4	47031	books	1f774f4ed35becf0eb3c24420f92ad...	2603.88

```
EXPLAIN (ANALYZE, BUFFERS)
SELECT id, code, name, price
FROM products
WHERE active = true and name LIKE '1f%'
LIMIT 100
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=1000.00..61771.33 rows=70 width=132) (actual time=0.235..33.008 rows=100 loops=1)
2	Buffers: shared hit=24 read=1994
3	-> Gather (cost=1000.00..61771.33 rows=70 width=132) (actual time=0.234..32.998 rows=100 loops=1)
4	Workers Planned: 2
5	Workers Launched: 2
6	Buffers: shared hit=24 read=1994
7	-> Parallel Seq Scan on products (cost=0.00..60764.33 rows=29 width=132) (actual time=0.307..2.479 rows=34 loops=3)
8	Filter: (active AND (name ~~ '1f%'::text))
9	Rows Removed by Filter: 12032
10	Buffers: shared hit=24 read=1994
11	Planning Time: 0.090 ms
12	Execution Time: 33.059 ms

GIN

Подключение расширения триграмм (разбиение на 3 всевозможных слога):
CREATE EXTENSION IF NOT EXISTS pg_trgm;

Создание GIN индекса по имени товара:
CREATE INDEX idx_products_active_name_trgm ON products USING gin ((name::text) gin_trgm_ops)
WHERE active = true;

```
EXPLAIN (ANALYZE, BUFFERS)
SELECT id, code, name, price
FROM products
WHERE active = true and name::text LIKE '1f%'
LIMIT 100
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=60.41..378.12 rows=100 width=132) (actual time=0.839..0.922 rows=100 loops=1)
2	Buffers: shared hit=123
3	-> Bitmap Heap Scan on products (cost=60.41..11119.96 rows=3481 width=132) (actual time=0.839..0.917 rows=100 loops=1)
4	Recheck Cond: (((name)::text ~~ '1f%'::text) AND active)
5	Heap Blocks: exact=97
6	Buffers: shared hit=123
7	-> Bitmap Index Scan on idx_products_active_name_trgm (cost=0.00..59.54 rows=3481 width=0) (actual time=0.620..0.621 rows=2716 loops=1)
8	Index Cond: ((name)::text ~~ '1f%'::text)
9	Buffers: shared hit=26
10	Planning:
11	Buffers: shared hit=1
12	Planning Time: 0.095 ms
13	Execution Time: 0.974 ms

Запрос №6

```
-- EXPLAIN (ANALYZE, BUFFERS)
SELECT *
FROM products
WHERE active = true and created_at > '2023-12-01 00:00:00'
LIMIT 100
```

Data Output | Messages | Notifications

	id [PK] integer	created_at timestamp without time zone	active boolean	sort integer	price numeric (10,2)	code character (20)	name character (100)	description character (255)
1	326036	2023-12-16 01:13:15.84696	true	78	5.96	electronics	e80edef857e1c17ead45216f05fbe0...	0f8bdb03c93d0aafee8a60d2f7b57...
2	326060	2023-12-09 11:09:38.130211	true	3511	7901.76	electronics	ebe7a15964abc7733b22ffc912bcaf...	043da48349300352ded7e4c2b540...
3	326081	2023-12-06 22:50:08.363776	true	2151	4567.32	books	a0d1920ebbdb3212b114730b13008...	ffb41c9ed3049212387113bdfb848...
4	326156	2023-12-21 20:25:20.006179	true	2642	9979.83	books	bc502e72866d289e7834283e10347...	8f7062664ec68b6cc2ad1980e80e...


```
EXPLAIN (ANALYZE, BUFFERS)
SELECT *
FROM products
WHERE active = true and created_at > '2023-12-01 00:00:00'
LIMIT 100
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=0.00..951.97 rows=100 width=404) (actual time=0.012..2.754 rows=100 loops=1)
2	Buffers: shared hit=15 read=784
3	-> Seq Scan on products (cost=0.00..68056.00 rows=7149 width=404) (actual time=0.011..2.749 rows=100 loops=1)
4	Filter: (active AND (created_at > '2023-12-01 00:00:00'::timestamp without time zone))
5	Rows Removed by Filter: 14179
6	Buffers: shared hit=15 read=784
7	Planning:
8	Buffers: shared hit=1
9	Planning Time: 0.078 ms
10	Execution Time: 2.792 ms

BRIN

CREATE INDEX idx_products_created_at_brin ON products USING brin (created_at);

```
1  EXPLAIN (ANALYZE, BUFFERS)
2  SELECT *
3  FROM products
4  WHERE active = true and created_at > '2022-12-01 00:00:00'
5  LIMIT 100
6
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Limit (cost=0.00..73.87 rows=100 width=404) (actual time=0.011..0.270 rows=100 loops=1)
2	Buffers: shared hit=17 read=46
3	-> Seq Scan on products (cost=0.00..68056.00 rows=92133 width=404) (actual time=0.009..0.264 rows=100 loop...
4	Filter: (active AND (created_at > '2022-12-01 00:00:00'::timestamp without time zone))
5	Rows Removed by Filter: 942
6	Buffers: shared hit=17 read=46
7	Planning:
8	Buffers: shared hit=2
9	Planning Time: 0.096 ms
10	Execution Time: 0.312 ms

Запрос №7 (hash)

```
1  -- EXPLAIN (ANALYZE, BUFFERS)
2  SELECT id, code, name, price
3  FROM products
4  where code = 'auto' and active = true
5
```

Data Output | Messages | Notifications

	id [PK] integer	code character (20)	name character (100)	price numeric (10,2)
1	516099	auto	f6d2bae2151a3cb402dd5e09ca08dd...	629.15
2	516100	auto	8dd1695e33fa74970a64e613f11822...	1450.53
3	516101	auto	a89a644f1bb8ad05702383079774b3...	2200.49

```
1  EXPLAIN (ANALYZE, BUFFERS)
2  SELECT id, code, name, price
3  FROM products
4  where code = 'auto' and active = true
5
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Seq Scan on products (cost=0.00..68056.00 rows=68523 width=132) (actual time=0.178..124.975 rows=70159 loops=...
2	Filter: (active AND (code = 'auto'::bpchar))
3	Rows Removed by Filter: 929841
4	Buffers: shared hit=16346 read=39210
5	Planning:
6	Buffers: shared hit=1
7	Planning Time: 0.087 ms
8	Execution Time: 126.390 ms

HASH

CREATE INDEX idx_products_category_hash ON products USING hash (code) WHERE active = true;

```
1  EXPLAIN (ANALYZE, BUFFERS)
2  SELECT id, code, name, price
3  FROM products
4  where code = 'auto' and active = true
5
```

Data Output | Messages | Notifications

	QUERY PLAN text
1	Bitmap Heap Scan on products (cost=2607.05..61939.40 rows=68523 width=132) (actual time=6.866..142.639 rows=70159 loops=1)
2	Recheck Cond: ((code = 'auto'::bpchar) AND active)
3	Heap Blocks: exact=40592
4	Buffers: shared read=40765
5	-> Bitmap Index Scan on idx_products_active_code_hash (cost=0.00..2589.92 rows=68523 width=0) (actual time=3.629..3.629 rows=70159 loop...
6	Index Cond: (code = 'auto'::bpchar)
7	Buffers: shared read=173
8	Planning:
9	Buffers: shared read=1
10	Planning Time: 0.109 ms
11	Execution Time: 144.433 ms

Запрос №8 (jsonb)

CREATE table api_responses (
  id bigserial primary key,
  payload jsonb NOT NULL
);

```
INSERT INTO api_responses(payload)
SELECT
  case when random() < 0.1
      then jsonb_build_object('error','timeout','code',504,'req_id',gs)
      else jsonb_build_object('status','ok','req_id',gs)
  end
FROM generate_series(1, 10000) gs;
```

```
1  -- EXPLAIN (ANALYZE, BUFFERS)
2  SELECT id, payload
3  FROM api_responses
4  WHERE payload ? 'error'
```

Data Output | Messages | Notifications

	id [PK] bigint	payload jsonb
1	3012	{"code": 504, "error": "timeout", "req_id": 3012}
2	3039	{"code": 504, "error": "timeout", "req_id": 3039}
3	3050	{"code": 504, "error": "timeout", "req_id": 3050}
4	3055	{"code": 504, "error": "timeout", "req_id": 3055}
5	3060	{"code": 504, "error": "timeout", "req_id": 3060}

```
1  EXPLAIN (ANALYZE, BUFFERS)
2  SELECT id, payload
3  FROM api_responses
4  WHERE payload ? 'error'
```

Data Output | Messages | Notifications

Showing rows: 1 to 8

	QUERY PLAN text
1	Seq Scan on api_responses (cost=10000000000.00..10000046275.00 rows=202020 width=55) (actual time=0.023..118.163 rows=200643 loops=...
2	Filter: (payload ? 'error'::text)
3	Rows Removed by Filter: 1799357
4	Buffers: shared hit=16290 read=4985
5	Planning:
6	Buffers: shared hit=3 read=2 dirtied=1
7	Planning Time: 0.412 ms
8	Execution Time: 121.986 ms

```
CREATE INDEX idx_api_responses_payload_gin ON api_responses USING GIN (payload);
```

```
1  EXPLAIN (ANALYZE, BUFFERS)
2  SELECT id, payload
3  FROM api_responses
4  WHERE payload ? 'error'
```

Data Output | Messages | Notifications

Showing rows: 1 to 11 | Page

	QUERY PLAN text
1	Bitmap Heap Scan on api_responses (cost=1387.21..25187.46 rows=202020 width=55) (actual time=10.184..93.470 rows=200643 loops=1)
2	Recheck Cond: (payload ? 'error'::text)
3	Heap Blocks: exact=21275
4	Buffers: shared hit=1 read=21308
5	-> Bitmap Index Scan on idx_api_responses_payload_gin (cost=0.00..1336.70 rows=202020 width=0) (actual time=8.569..8.569 rows=200643 loop…
6	Index Cond: (payload ? 'error'::text)
7	Buffers: shared hit=1 read=33
8	Planning:
9	Buffers: shared hit=1
10	Planning Time: 0.060 ms
11	Execution Time: 97.575 ms

Вывод индексов

```
1  SELECT
2  t.relname AS table_name,
3  i.relname AS index_name,
4  am.amname AS index_type, -- btree, gin, gist, hash и т.д.
5  pg_get_indexdef(i.oid) AS indexdef
6  FROM pg_class t
7  JOIN pg_index ix ON ix.indrelid = t.oid
8  JOIN pg_class i ON i.oid = ix.indexrelid
9  JOIN pg_am am ON am.oid = i.relam
10 JOIN pg_namespace n ON n.oid = t.relnamespace
11 WHERE n.nspname = 'public'
12 AND t.relname IN ('products', 'api_responses')
13 ORDER BY t.relname, i.relname;
```

Data Output | Messages | Notifications

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

	table_name name	index_name name	index_type name	indexdef text
1	api_responses	api_responses_pkey	btree	CREATE UNIQUE INDEX api_responses_pkey ON public.api_responses USING btree (id)
2	api_responses	idx_api_responses_payload_gin	gin	CREATE INDEX idx_api_responses_payload_gin ON public.api_responses USING gin (payload)
3	products	idx_products_active_code_hash	hash	CREATE INDEX idx_products_active_code_hash ON public.products USING hash (code) WHERE (active = true)
4	products	idx_products_active_name_trgm	gin	CREATE INDEX idx_products_active_name_trgm ON public.products USING gin (((name)::text) gin_trgm_ops) WHERE (active = true)
5	products	idx_products_active_price	btree	CREATE INDEX idx_products_active_price ON public.products USING btree (price DESC) WHERE (active = true)
6	products	idx_products_active_sort	btree	CREATE INDEX idx_products_active_sort ON public.products USING btree (sort) WHERE (active = true)
7	products	idx_products_created_at_brin	brin	CREATE INDEX idx_products_created_at_brin ON public.products USING brin (created_at)
8	products	products_pkey	btree	CREATE UNIQUE INDEX products_pkey ON public.products USING btree (id)