|  |  |  |
| --- | --- | --- |
| **№** | **ВОПРОС** | **ОТВЕТ** |
| **1** | **Оптимизируйте:**  **select \* from SPACE\_VEHICLE order by NAME\_SPACE\_VEHICLE;** | **SELECT name\_space\_vehicle FROM space\_vehicle ORDER BY name\_space\_vehicle** |
| **2** | **Оптимизируйте:**  **select \* from SPACE\_VEHICLE join COSMODROME on SPACE\_VEHICLE.num\_cosmodrome=COSMODROME.num\_cosmodrome order by NUM\_COUNTRY  (простой)** | CREATE INDEX idx\_space\_vehicle\_num\_cosmodrome ON SPACE\_VEHICLE (num\_cosmodrome);  CREATE INDEX idx\_cosmodrome\_num\_cosmodrome ON COSMODROME (num\_cosmodrome);  CREATE INDEX idx\_cosmodrome\_num\_country ON COSMODROME (NUM\_COUNTRY);  SELECT sv.NAME\_SPACE\_VEHICLE, c.name\_cosmodrom, c.NUM\_COUNTRY  FROM SPACE\_VEHICLE sv  INNER JOIN COSMODROME c ON sv.num\_cosmodrome = c.num\_cosmodrome  ORDER BY c.NUM\_COUNTRY; |
| **3** | **Оптимизируйте:**  **select \* from HISTORY\_SPACE\_VEHICLE S1, HISTORY\_SPACE\_VEHICLE S2 where extract(month from S1.DATE\_HISTORY\_SPACE\_VEHICLE) = extract(month from S2.DATE\_HISTORY\_SPACE\_VEHICLE)  (стандартный)** | CREATE INDEX idx\_history\_month ON HISTORY\_SPACE\_VEHICLE (EXTRACT(MONTH FROM DATE\_HISTORY\_SPACE\_VEHICLE));  EXPLAIN  WITH MonthlyData AS (  SELECT DISTINCT EXTRACT(MONTH FROM DATE\_HISTORY\_SPACE\_VEHICLE) AS month  FROM HISTORY\_SPACE\_VEHICLE  )  SELECT  hs1.\*,  hs2.\*  FROM HISTORY\_SPACE\_VEHICLE hs1  JOIN MonthlyData md ON EXTRACT(MONTH FROM hs1.DATE\_HISTORY\_SPACE\_VEHICLE) = md.month  JOIN HISTORY\_SPACE\_VEHICLE hs2 ON EXTRACT(MONTH FROM hs2.DATE\_HISTORY\_SPACE\_VEHICLE) = md.month  WHERE hs1.date\_history\_space\_vehicle != hs2.date\_history\_space\_vehicle; |
| **4** | **Создайте таблицу с одним ключевым полем строкового типа длиной 20, заполните ее произвольными строками (подумайте над механизмом их формирования).**  **Первая порция данных 100 записей, выполните запрос:**  **select \* from таблица order by поле**  **в двух вариантах с использованием индекса и без (узнайте имя построенного индекса) выполните:**  **SET STATISTICS INDEX имя индекса первичного ключа**  **Прокомментируйте результат**  **Добавьте еще 100 записей**  **И повторите предыдущие шаги, прокомментируйте результат.**  **(стандартный)** | CREATE TABLE test\_table (  key\_field VARCHAR(255) PRIMARY KEY  );  INSERT INTO test\_table (key\_field)  SELECT md5(random()::text || i::text)  FROM generate\_series(1, 100) AS i;  EXPLAIN ANALYZE select \* from test\_table order by key\_field;  *-- "QUERY PLAN"*  *"Sort (cost=5.32..5.57 rows=100 width=33) (actual time=0.916..0.927 rows=100 loops=1)"*  *" Sort Key: key\_field"*  *" Sort Method: quicksort Memory: 25kB"*  *" -> Seq Scan on test\_table (cost=0.00..2.00 rows=100 width=33) (actual time=0.037..0.049 rows=100 loops=1)"*  *"Planning Time: 3.179 ms"*  *"Execution Time: 0.970 ms"*  CREATE INDEX idx\_test\_table\_key\_field ON test\_table (key\_field);  -- ***Разницы между индексом и без него нет, поскольку выборка данных небольшая, ускорение только за счёт кэширования***  *-- "QUERY PLAN"*  *"Sort (cost=5.32..5.57 rows=100 width=33) (actual time=0.568..0.579 rows=100 loops=1)"*  *" Sort Key: key\_field"*  *" Sort Method: quicksort Memory: 25kB"*  *" -> Seq Scan on test\_table (cost=0.00..2.00 rows=100 width=33) (actual time=0.029..0.042 rows=100 loops=1)"*  *"Planning Time: 0.144 ms"*  *"Execution Time: 0.616 ms"*  INSERT INTO test\_table (key\_field)  SELECT md5(random()::text || i::text)  FROM generate\_series(101, 200) AS i;  EXPLAIN ANALYZE select \* from test\_table order by key\_field;  -- *"QUERY PLAN"*  *"Sort (cost=11.64..12.14 rows=200 width=33) (actual time=0.983..1.003 rows=200 loops=1)"*  *" Sort Key: key\_field"*  *" Sort Method: quicksort Memory: 25kB"*  *" -> Seq Scan on test\_table (cost=0.00..4.00 rows=200 width=33) (actual time=0.047..0.076 rows=200 loops=1)"*  *"Planning Time: 0.144 ms"*  *"Execution Time: 1.048 ms"*  INSERT INTO test\_table (key\_field)  SELECT md5(random()::text || i::text)  FROM generate\_series(101, 100000) AS i;  EXPLAIN ANALYZE select \* from test\_table order by key\_field;  -- ***На этом этапе начинает работать индекс (Index Only Scan), потому что объём данных очень большой, и последовательное сканирование уже ресурсоёмкое***  *-- "QUERY PLAN"*  *"Index Only Scan using idx\_test\_table\_key\_field on test\_table (cost=0.42..5429.92 rows=100100 width=33) (actual time=1.113..25.882 rows=100100 loops=1)"*  *" Heap Fetches: 0"*  *"Planning Time: 5.618 ms"*  *"Execution Time: 36.984 ms"*  SELECT relname FROM pg\_class WHERE oid = 'public.idx\_test\_table\_key\_field'::regclass;  -- |