
ORACLE®

Histogram Support in MySQL 8.0

Øystein Grøvlen
Senior Principal Software Engineer

MySQL Optimizer Team, Oracle
February 2018

Program Agenda

1. Motivating example
2. Quick start guide
3. How are histograms used?
4. Query example
5. Some advice

Program Agenda

1. Motivating example
2. Quick start guide
3. How are histograms used?
4. Query example
5. Some advice

Motivating Example

JOIN Query

EXPLAIN SELECT *
FROM orders JOIN customer ON o_custkey = c_custkey
WHERE o_orderdate < '1993-01-01' AND c_acctbal < -1000;

id	select type	table	type	possible keys	key	key len	ref	rows	filtered	extra
1	SIMPLE	orders	ALL	i_o_orderdate, i_o_custkey	NULL	NULL	NULL	15000000	31.19	Using where
1	SIMPLE	customer	eq_ref	PRIMARY	PRIMARY	4	dbt3.orders.o_custkey	1	33.33	Using where

Motivating Example

Reverse join order

EXPLAIN SELECT /*+ JOIN_ORDER(customer, orders) */ *
FROM orders JOIN customer ON o_custkey = c_custkey
WHERE o_orderdate < '1993-01-01' AND c_acctbal < -1000;

id	select type	table	type	possible keys	key	key len	ref	rows	filtered	extra
1	SIMPLE	customer	ALL	PRIMARY	NULL	NULL	NULL	1500000	33.33	Using where
1	SIMPLE	orders	ref	i_o_orderdate, i_o_custkey	i_o_custkey	5	dbt3. customer. c_custkey	15	31.19	Using where

Comparing Join Order

Performance

Histograms

Create histogram to get a better plan

ANALYZE TABLE customer UPDATE HISTOGRAM ON c_acctbal WITH 1024 BUCKETS;

```
EXPLAIN SELECT *
FROM orders JOIN customer ON o_custkey = c_custkey
WHERE o_orderdate < '1993-01-01' AND c_acctbal < -1000;
```

id	select type	table	type	possible keys	key	key len	ref	rows	filtered	extra
1	SIMPLE	customer	ALL	PRIMARY	NULL	NULL	NULL	1500000	0.00	Using where
1	SIMPLE	orders	ref	i_o_orderdate, i_o_custkey	i_o_custkey	5	dbt3. customer. c_custkey	15	31.19	Using where

Program Agenda

1. Motivating example
2. Quick start guide
3. How are histograms used?
4. Query example
5. Some advice

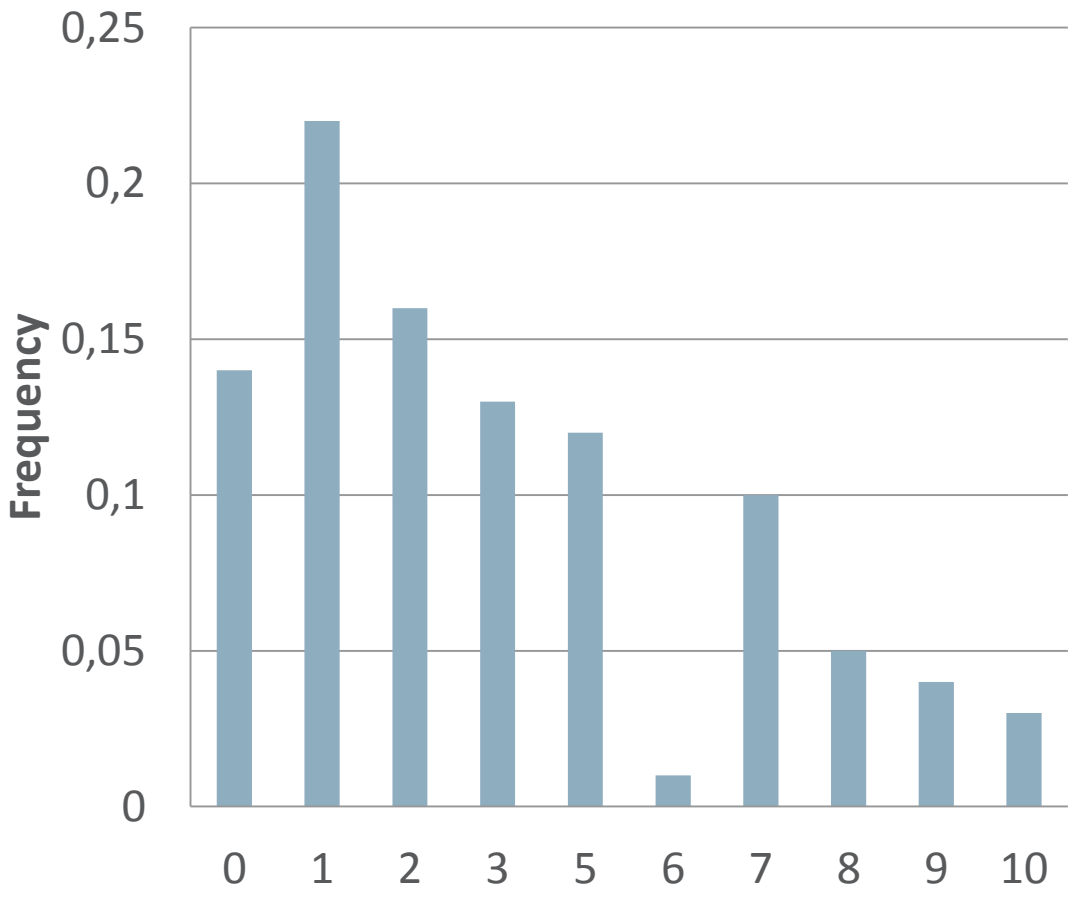
Histograms

Column statistics

- Information about value distribution for a column
- Data values group in buckets
  - Frequency calculated for each bucket
  - Maximum 1024 buckets
- May use sampling to build histogram
  - Sample rate depends on available memory
- Automatically chooses between two histogram types:
  - Singleton: One value per bucket
  - Equi-height: Multiple values per bucket

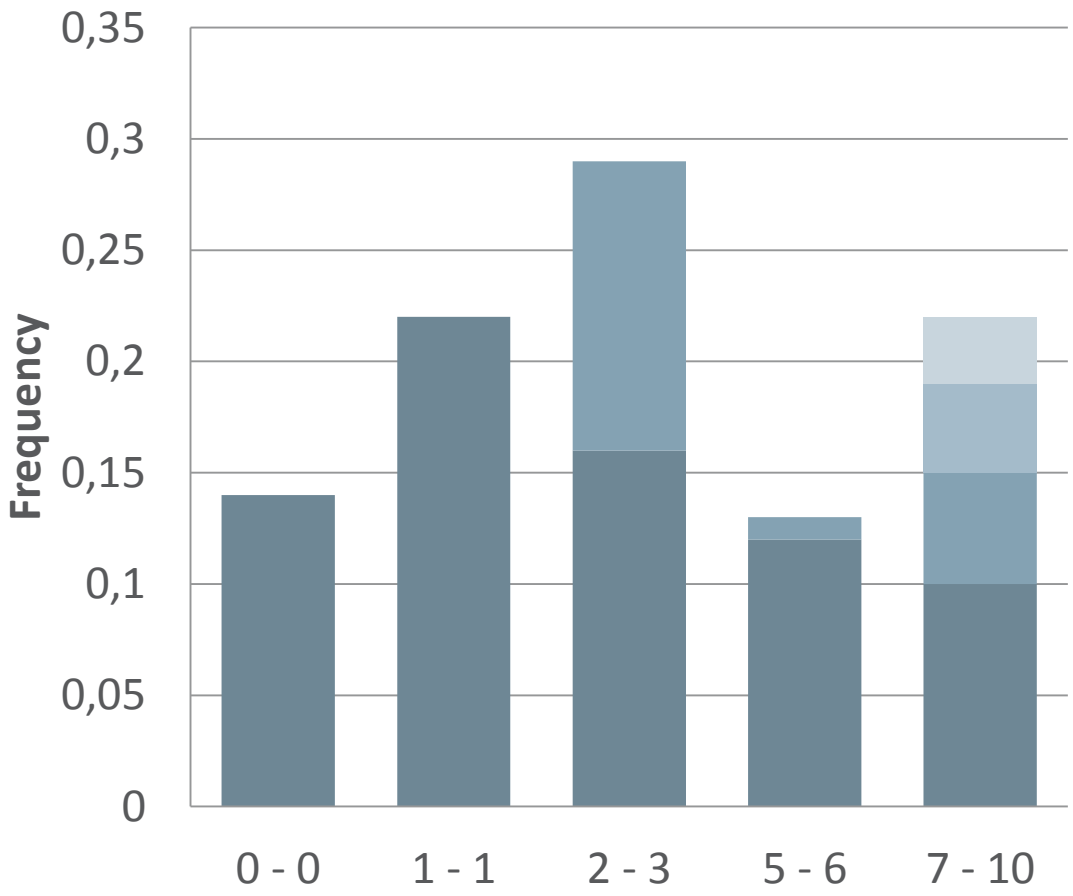
Singleton Histogram

- One value per bucket
- Each bucket stores:
  - Value
  - Cumulative frequency
- Well suited to estimate both equality and range predicates



Equi-Height Histogram

- Multiple values per bucket
- Not quite equi-height
  - Values are not split across buckets
    ⇒Frequent values in separate buckets
- Each bucket stores:
  - Minimum value
  - Maximum value
  - Cumulative frequency
  - Number of distinct values
- Best suited for range predicates



Usage

- Create or refresh histogram(s) for column(s):

  ANALYZE TABLE *table* UPDATE HISTOGRAM ON *column* [, *column*] WITH *n* BUCKETS;

  – Note: Will only update histogram, not other statistics

- Drop histogram:

  ANALYZE TABLE *table* DROP HISTOGRAM ON *column* [, *column*];

- Based on entire table or sampling:

  – Depends on avail. memory: **histogram_generation_max_mem_size** (default: 20 MB)

- New storage engine API for sampling

  – Default implementation: Full table scan even when sampling

  – Storage engines may implement more efficient sampling

Storage

- Stored in a JSON column in data dictionary
- Can be inspected in Information Schema table:

```
SELECT JSON_PRETTY(histogram)
FROM information_schema.column_statistics
WHERE schema_name = 'dbt3_sf1'
    AND table_name ='lineitem'
    AND column_name = 'l_linenumber';
```

Histogram content


```
{
 "buckets": [[1, 0.2499493852494869 8], [2, 0.4642106640072052 3],
  [3, 0.6427401784471978], [4, 0.7855470933802572],
  [5, 0.8927398868395817], [6, 0.9642370753255 8], [7, 1] ],
 "data-type": "int",
 "null-values": 0.0,
 "collation-id": 8,
 "last-updated": "2018-02-03 21:05:21.690872",
 "sampling-rate": 0.2082911543745725 2,
 "histogram-type": "singleton",
 "number-of-buckets-specified": 1024
}
```

Strings

- Max. 42 characters considered
- Base64 encoded

```
SELECT FROM_BASE64(SUBSTR(v, LOCATE(':', v, 10) + 1)) value, c cumulfreq
FROM information_schema.column_statistics,
        JSON_TABLE(histogram->'$.buckets', '$[*]'
                COLUMNS(v VARCHAR(60) PATH '$[0]',
                        c double PATH '$[1]')) hist
WHERE column_name = 'o_orderstatus';
```

```
+-------+--------------------+
| value | cumulfreq          |
+-------+--------------------+
| F     | 0.486252926438575 6 |
| O     |  0.974029654577566 |
| P     | 0.9999999999999999 |
+-------+--------------------+
```

Calculate Bucket Frequency

Use window function

```
SELECT FROM_BASE64(SUBSTR(v, LOCATE(':', v, 10) + 1)) value, c cumulfreq,
        c - LAG(c, 1, 0) over () freq
FROM information_schema.column_statistics,
     JSON_TABLE(histogram->'$.buckets', '$[*]'
                COLUMNS(v VARCHAR(60) PATH '$[0]',
                        c double PATH '$[1]')) hist
WHERE column_name = 'o_orderstatus';
```

```
+-------+--------------------+----------------------+
| value | cumulfreq          | freq                 |
+-------+--------------------+----------------------+
| F     | 0.4862529264385756 |   0.4862529264385756 |
| O     |  0.974029654577566 |  0.4877767281389903 |
| P     | 0.9999999999999999 | 0.02597034542243392 |
+-------+--------------------+----------------------+
```

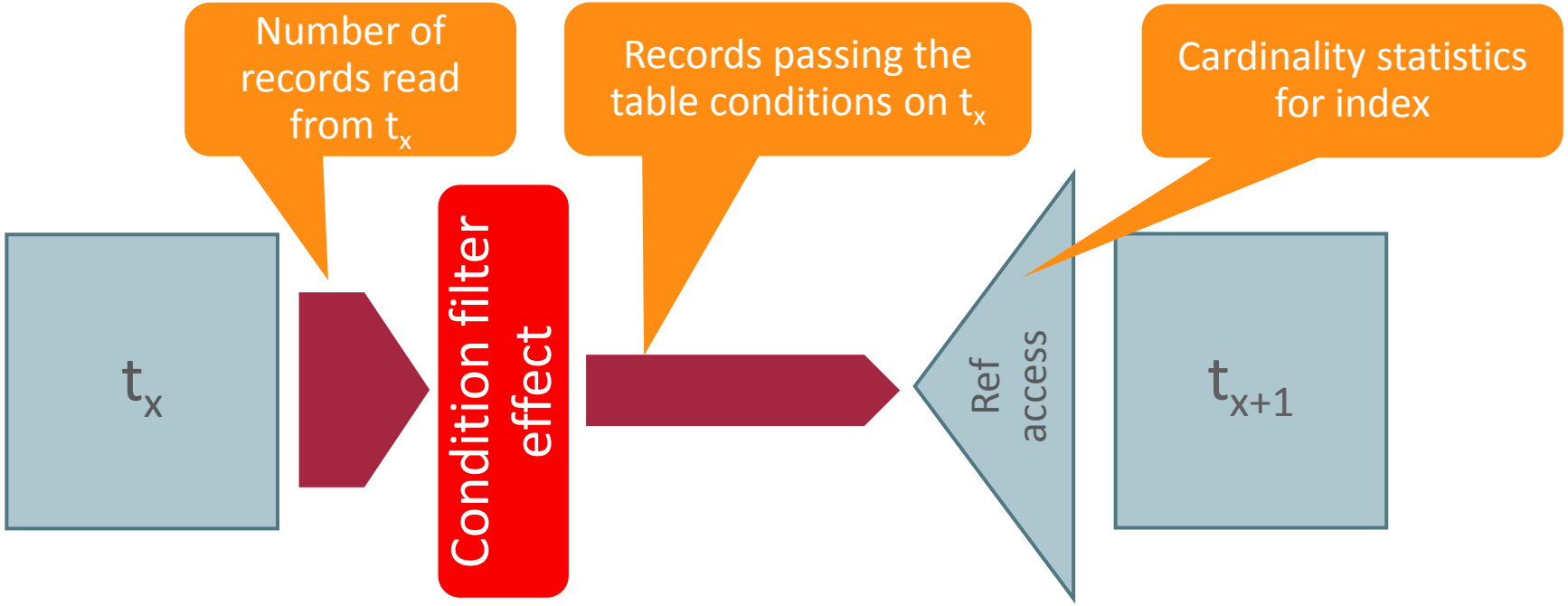
Program Agenda

1. Motivating example
2. Quick start guide
3. How are histograms used?
4. Query example
5. Some advice

When are Histograms useful?

Estimate cost of join

- t_x JOIN t_{x+1}



- records(t_{x+1}) = records(t_x) * condition_filter_effect * records_per_key

How to Calculate Condition Filter Effect, MySQL 5.7


```
SELECT *
FROM office JOIN employee ON office.id = employee.office_id
WHERE office_name = 'San Francisco' AND
      employee.name = 'John' AND age > 21 AND
            hire_date BETWEEN '2014-01-01' AND '2014-06-01';
```

Filter estimate based on what is available:

1. Range estimate
2. Index statistics
3. Guesstimate

=	0.1
<=,<,>,>=	1/3
BETWEEN	1/9
NOT <op>	1 – SEL(<op>)
AND	P(A and B) = P(A) * P(B)
OR	P(A or B) = P(A) + P(B) – P(A and B)
…	…

How to Calculate Condition Filter Effect, MySQL 5.7


```
SELECT *
FROM office JOIN employee ON office.id = employee.office_id
WHERE office_name = 'San Francisco' AND
      employee.name = 'John' AND age > 21 AND
            hire_date BETWEEN '2014-01-01' AND '2014-06-01';
```

Filter estimate based on what is available:

1. Range estimate
2. Index statistics
3. Histograms
4. Guesstimate

=	0.1
<=,<,>,>=	1/3
BETWEEN	1/9
NOT <op>	1 – SEL(<op>)
AND	P(A and B) = P(A) * P(B)
OR	P(A or B) = P(A) + P(B) – P(A and B)
…	…

Calculating Condition Filter Effect for Tables

Example without histograms

0.03 (index)

```
SELECT *
FROM office JOIN employee ON office.id = employee.office_id
WHERE office_name = 'San Francisco' AND
      employee.name = 'John' AND age > 21 AND
           hire_date BETWEEN '2014-01-01' AND '2014-06-01';
```

0.29 (range)

0.1 (guesstimate)

0.33 (guesstimate)

Condition filter effect for tables:

- office: 0.03
- employee: 0.29 * 0.1 * 0.33 ≈ 0.01

Calculating Condition Filter Effect for Tables

Example with histogram

```
SELECT *
FROM office JOIN employee ON office.id = employee.office_id
WHERE office_name = 'San Francisco' AND
      employee.name = 'John' AND age > 21 AND
           hire_date BETWEEN '2014-01-01' AND '2014-06-01';
```

0.03 (index)

0.29 (range)

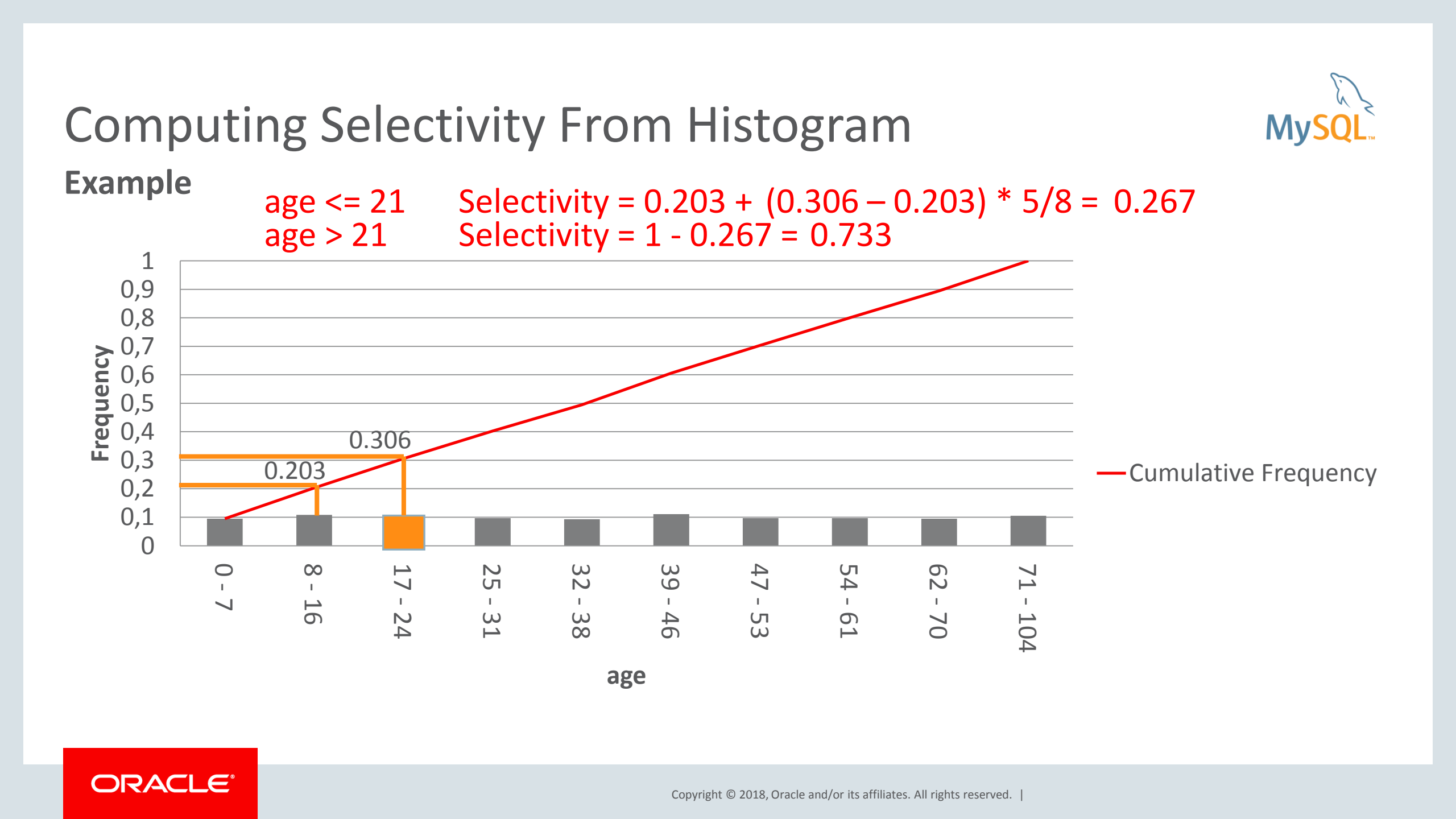
0.1 (guesstimate)

0.95 (histogram)

Condition filter effect for tables:

- office: 0.03
- employee: 0.29 * 0.1 * 0.95 ≈ 0.03

Computing Selectivity From Histogram



Example

age <= 21 Selectivity = 0.203 + (0.306 – 0.203) * 5/8 = 0.267

age > 21 Selectivity = 1 - 0.267 = 0.733

Program Agenda

1. Motivating example
2. Quick start guide
3. How are histograms used?
4. Query example
5. Some advice

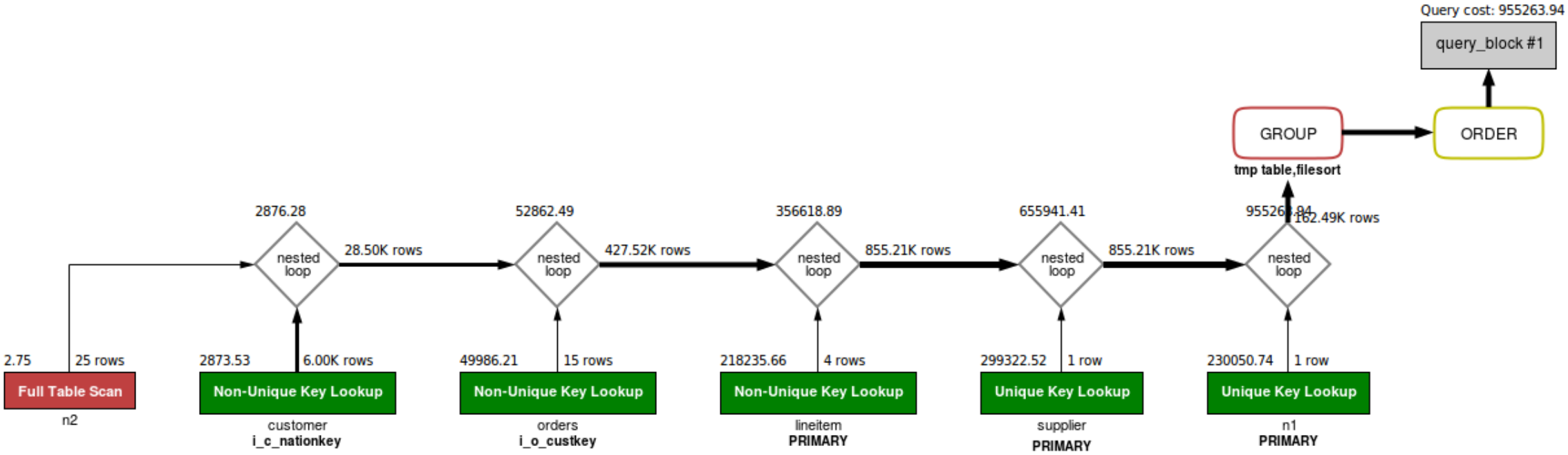
DBT-3 Query 7

Volume Shipping Query

```
SELECT supp_nation, cust_nation, l_year, SUM(volume) AS revenue
FROM    (SELECT  n1.n_name AS supp_nation,  n2.n_name AS cust_nation,
                 EXTRACT(YEAR FROM l_shipdate) AS l_year,
                 l_extendedprice * (1 - l_discount) AS volume
          FROM  supplier, lineitem, orders, customer, nation n1, nation n2
          WHERE  s_suppkey = l_suppkey AND o_orderkey = l_orderkey
              AND c_custkey = o_custkey AND s_nationkey = n1.n_nationkey
              AND c_nationkey = n2.n_nationkey
              AND ((n1.n_name = 'RUSSIA' AND n2.n_name = 'FRANCE')
                OR (n1.n_name = 'FRANCE' AND n2.n_name = 'RUSSIA'))
              AND l_shipdate BETWEEN '1995-01-01' AND '1996-12-31') AS shipping
GROUP BY supp_nation , cust_nation , l_year
ORDER BY supp_nation , cust_nation , l_year;
```

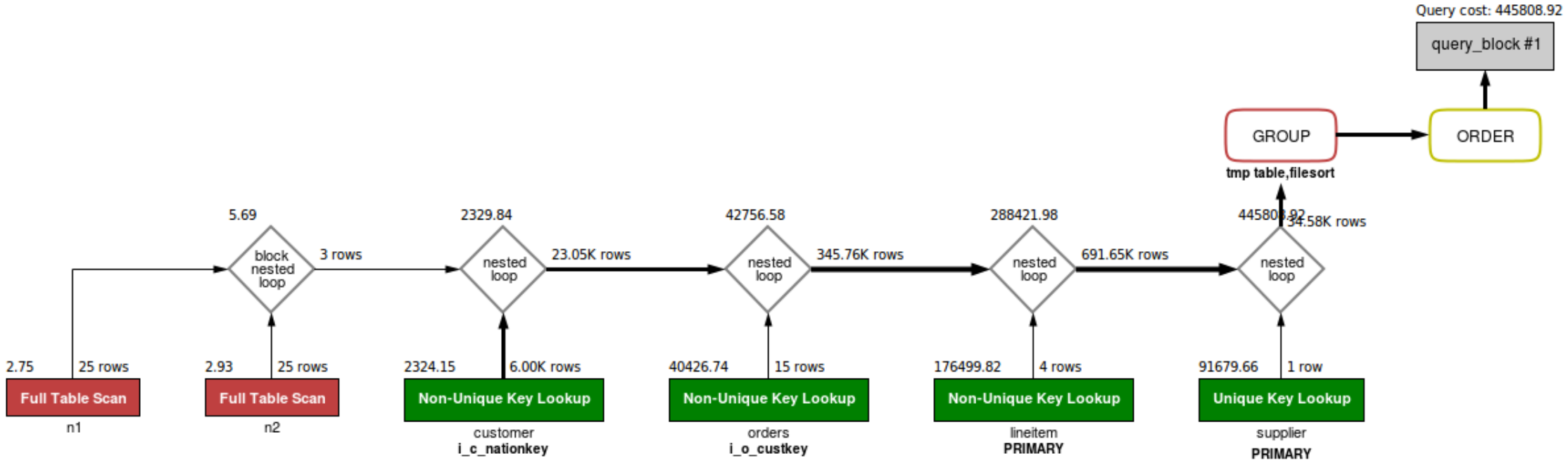
DBT-3 Query 7

Query plan without histogram



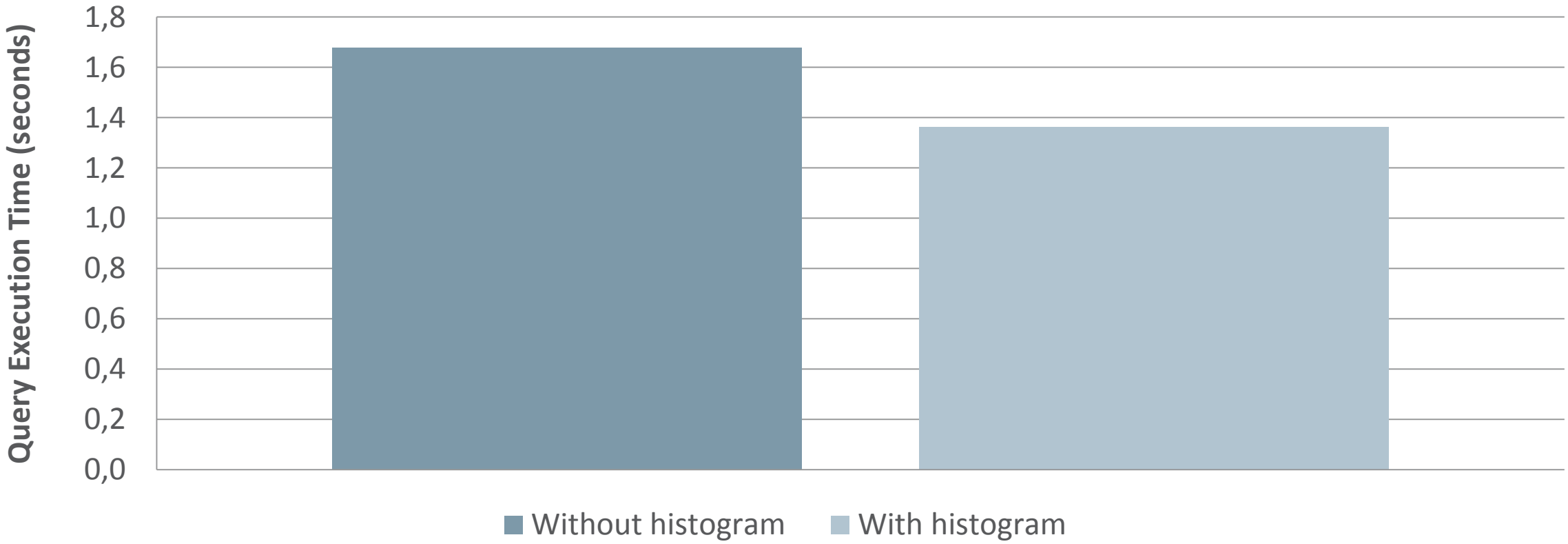
DBT-3 Query 7

Query plan with histogram



DBT-3 Query 7

Performance



Program Agenda

1. Motivating example
2. Quick start guide
3. How is histograms used?
4. Query example
5. Some advice

Some advice

Which columns to create histograms for?

- Histograms are useful for columns that are
  - not the first column of any index, and
  - used in WHERE conditions of
    - JOIN queries
    - Queries with IN-subqueries
    - ORDER BY ... LIMIT queries
- Best fit
  - Low cardinality columns (e.g., gender, orderStatus, dayOfWeek, enums)
  - Columns with uneven distribution (skew)
  - Stable distribution (do not change much over time)

Some more advice

- When not to create histograms:
  - First column of an index
  - Never used in WHERE clause
  - Monotonically increasing column values (e.g. date columns)
    - Histogram will need frequent updates to be accurate
    - Consider to create index
- How many buckets?
  - If possible, enough to get a singleton histogram
  - For equi-height, 100 buckets should be enough

More information

- MySQL Server Team blog
  - http://mysqlserverteam.com/
  - https://mysqlserverteam.com/histogram-statistics-in-mysql/ (Erik Frøseth)
- My blog:
  - http://oysteing.blogspot.com/
- MySQL forums:
  - Optimizer & Parser: http://forums.mysql.com/list.php?115
  - Performance: http://forums.mysql.com/list.php?24

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Integrated Cloud

Applications & Platform Services


ORACLE®