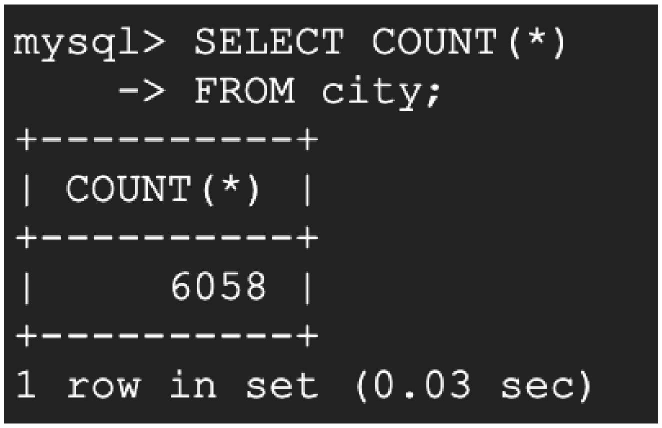
Stage 3 Report

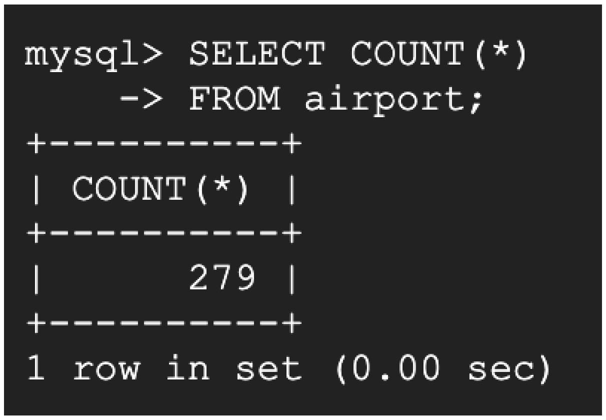
1. Data Definition Language

```
USE `flight031`;
DROP TABLE IF EXISTS `city`;
CREATE TABLE `city`(
   `Name` VARCHAR(30) NOT NULL,
   `Country` VARCHAR(30) NOT NULL,
   `Latitude` DECIMAL(8,5) NOT NULL,
   `Longitude` DECIMAL(8,5) NOT NULL,
   PRIMARY KEY (`Name`)
);
// After that we inserted 6058 cites into the database:
```



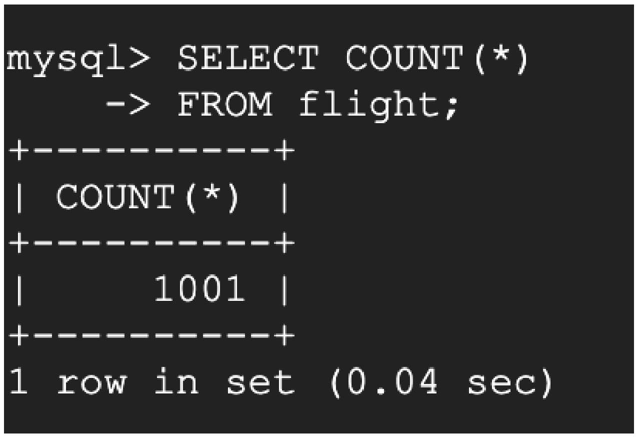
```
mysql> SELECT COUNT(*)
    -> FROM city;
+----------+
| COUNT(*) |
+----------+
|     6058 |
+----------+
1 row in set (0.03 sec)
```

```
DROP TABLE IF EXISTS `airport`;
CREATE TABLE `airport` (
`IATA_CODE` VARCHAR(3) NOT NULL,
`Name` VARCHAR(30) NOT NULL,
`City` VARCHAR(30) NOT NULL,
`State` VARCHAR(30) NOT NULL,
`Latitude` DECIMAL(8,5) NOT NULL,
`Longitude` DECIMAL(8,5) NOT NULL,
PRIMARY KEY(`IATA_CODE`),
CONSTRAINT `flight_rf_2` FOREIGN KEY (`City`) REFERENCES  `city` (`Name`)
);
// After that we inserted 279 airports into the database
// Note that there are not as more as 1000 airports in the US, so only 279 in the
dataset
```



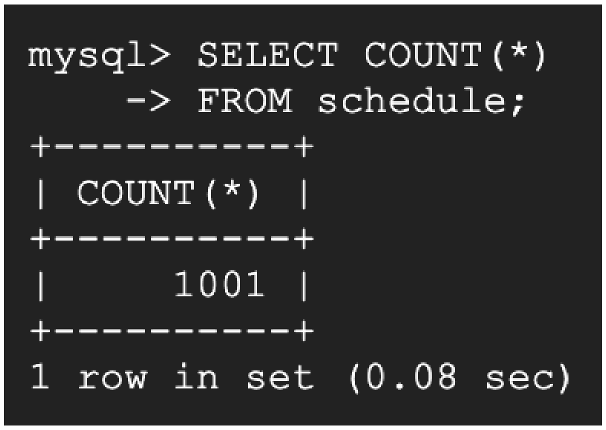
```
mysql> SELECT COUNT(*)
    -> FROM airport;
+----------+
| COUNT(*) |
+----------+
|      279 |
+----------+
1 row in set (0.00 sec)
```

```
DROP TABLE IF EXISTS `flight`;
CREATE TABLE `flight`(
   `Year` INT NOT NULL,
   `Month` INT NOT NULL,
   `Day` INT NOT NULL,
   `TailNumber` VARCHAR(6) NOT NULL,
   `PlannedDepartureTime` VARCHAR(4) NOT NULL,
   `DayOfWeek` INT,
   `Airline` VARCHAR(2) NOT NULL,
   `OriginAirport` VARCHAR(3) NOT NULL,
   `DestinationAirport` VARCHAR(3) NOT NULL,
   PRIMARY KEY (`Year`, `Month`, `Day`, `TailNumber`, `PlannedDepartureTime`),
   CONSTRAINT `flight_rf_1_1` FOREIGN KEY(`OriginAirport`)
REFERENCES`airport`(`IATA_CODE`),
   CONSTRAINT `flight_rf_1_2` FOREIGN KEY(`DestinationAirport`)
REFERENCES`airport`(`IATA_CODE`)
);
// Insert 1001 flights into the database;
```



```
mysql> SELECT COUNT(*)
    -> FROM flight;
+----------+
| COUNT(*) |
+----------+
|     1001 |
+----------+
1 row in set (0.04 sec)
```

```
DROP TABLE IF EXISTS `schedule`;
CREATE TABLE `schedule` (
`Year` INT NOT NULL,
`Month` INT NOT NULL,
`Day` INT NOT NULL,
`TailNumber` VARCHAR(6) NOT NULL,
`PlannedDepartureTime` VARCHAR(4) NOT NULL,
`IATA_CODE` VARCHAR(3) NOT NULL,
PRIMARY KEY (`Year`, `Month`, `Day`, `TailNumber`, `PlannedDepartureTime`),
CONSTRAINT `flight_rf_3_1` FOREIGN KEY (`Year`, `Month`, `Day`, `TailNumber`,
`PlannedDepartureTime`) REFERENCES `flight` (`Year`, `Month`, `Day`, `TailNumber`,
`PlannedDepartureTime`),
CONSTRAINT `flight_rf_4` FOREIGN KEY (`IATA_CODE`) REFERENCES `airport` (`IATA_CODE`)
);
// Insert 1001 rows in schedule table;
```

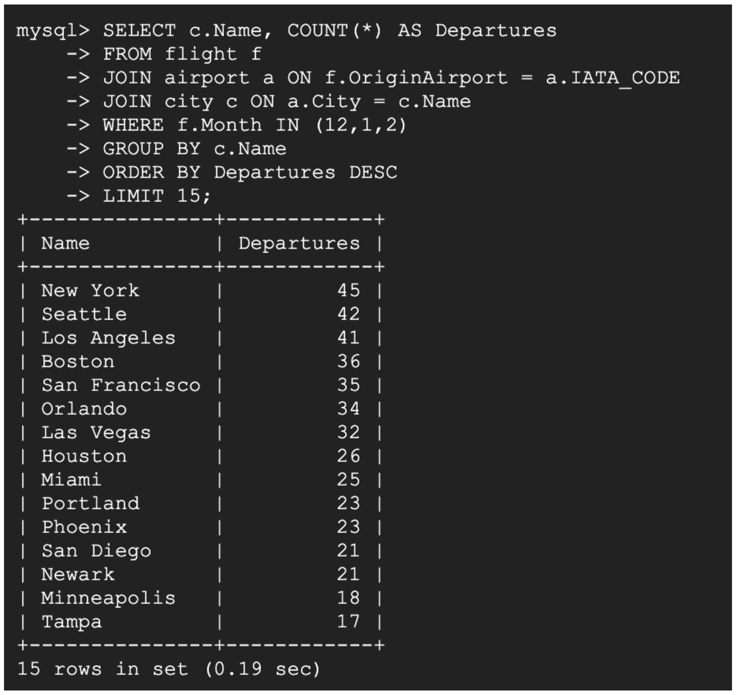


```
mysql> SELECT COUNT(*)
    -> FROM schedule;
+----------+
| COUNT(*) |
+----------+
|     1001 |
+----------+
1 row in set (0.08 sec)
```

2. Advanced Queries

1. Find the cities with the highest number of flight departures in winter months.

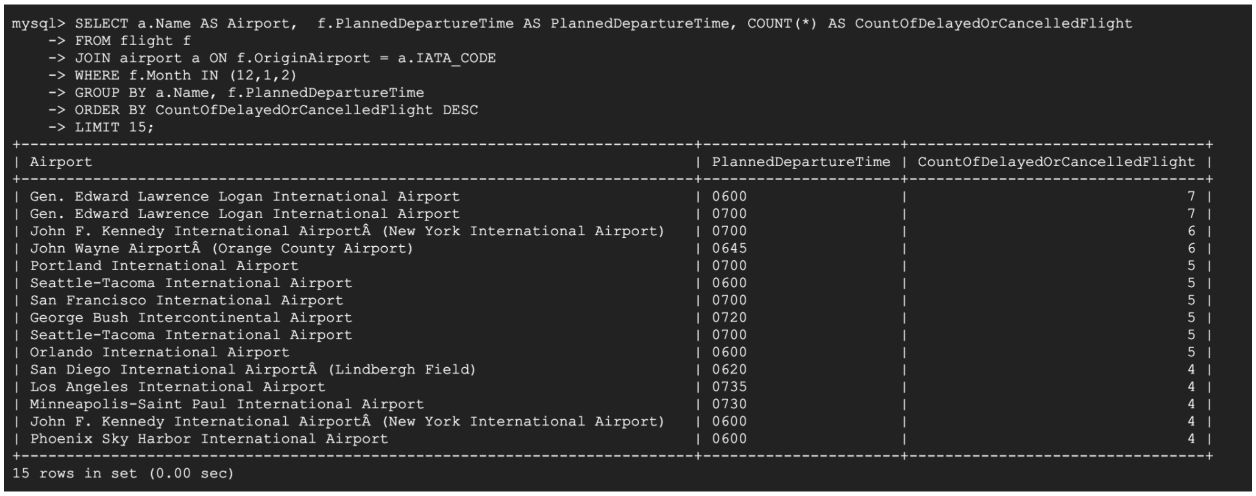
   SELECT c.Name, COUNT(*) AS Departures
   FROM flight f
   JOIN airport a ON f.OriginAirport = a.IATA_CODE
   JOIN city c ON a.City = c.Name
   WHERE f.Month IN (12,1,2)
   GROUP BY c.Name
   ORDER BY Departures DESC
   LIMIT 15;



```
mysql> SELECT c.Name, COUNT(*) AS Departures
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> JOIN city c ON a.City = c.Name
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY c.Name
    -> ORDER BY Departures DESC
    -> LIMIT 15;
+---------------+------------+
| Name          | Departures |
+---------------+------------+
| New York      |         45 |
| Seattle       |         42 |
| Los Angeles   |         41 |
| Boston        |         36 |
| San Francisco |         35 |
| Orlando       |         34 |
| Las Vegas     |         32 |
| Houston       |         26 |
| Miami         |         25 |
| Portland      |         23 |
| Phoenix       |         23 |
| San Diego     |         21 |
| Newark        |         21 |
| Minneapolis   |         18 |
| Tampa         |         17 |
+---------------+------------+
15 rows in set (0.19 sec)
```

2. Find the average delay time (in minutes) of flights departing from a given airport to all destinations in winter months.

   SELECT a.Name AS Airport, f.PlannedDepartureTime AS PlannedDepartureTime, COUNT(*) AS CountOfDelayedOrCancelledFlight
   FROM flight f
   JOIN airport a ON f.OriginAirport = a.IATA_CODE
   WHERE f.Month IN (12,1,2)
   GROUP BY a.Name, f.PlannedDepartureTime
   ORDER BY CountOfDelayedOrCancelledFlight DESC
   LIMIT 15;

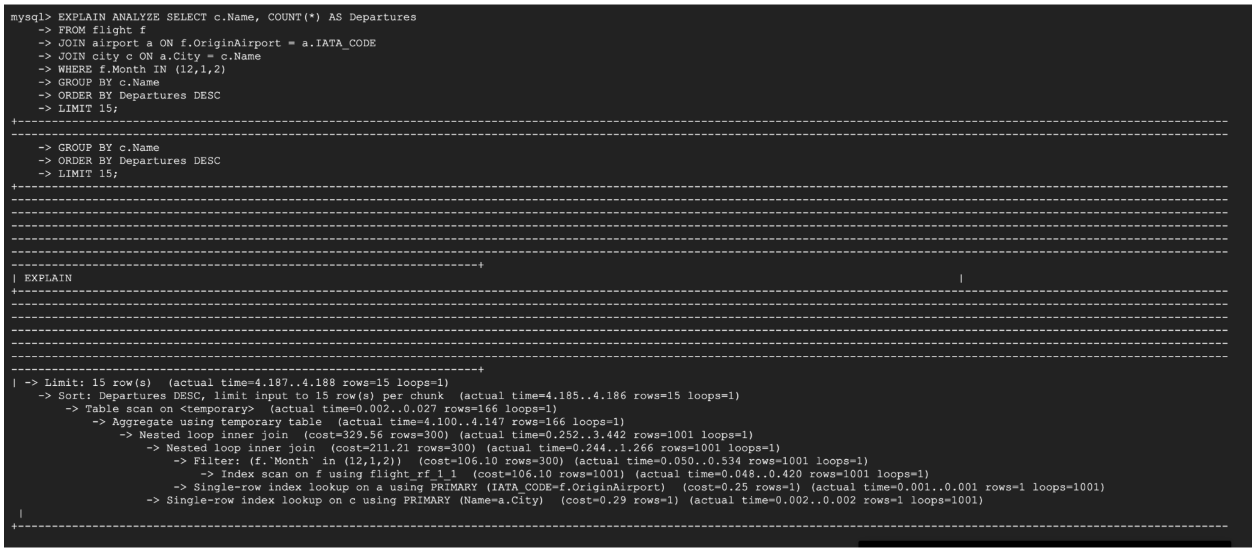


```
mysql> SELECT a.Name AS Airport,  f.PlannedDepartureTime AS PlannedDepartureTime, COUNT(*) AS CountOfDelayedOrCancelledFlight
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY a.Name, f.PlannedDepartureTime
    -> ORDER BY CountOfDelayedOrCancelledFlight DESC
    -> LIMIT 15;
+----------------------------------------------------------------------------+----------------------+---------------------------------+
| Airport                                                                    | PlannedDepartureTime | CountOfDelayedOrCancelledFlight |
+----------------------------------------------------------------------------+----------------------+---------------------------------+
| Gen. Edward Lawrence Logan International Airport                           | 0600                 |                               7 |
| Gen. Edward Lawrence Logan International Airport                           | 0700                 |                               7 |
| John F. Kennedy International AirportÂ (New York International Airport)    | 0700                 |                               6 |
| John Wayne AirportÂ (Orange County Airport)                                | 0645                 |                               6 |
| Portland International Airport                                             | 0700                 |                               5 |
| Seattle-Tacoma International Airport                                       | 0600                 |                               5 |
| San Francisco International Airport                                        | 0700                 |                               5 |
| George Bush Intercontinental Airport                                       | 0720                 |                               5 |
| Seattle-Tacoma International Airport                                       | 0700                 |                               5 |
| Orlando International Airport                                              | 0600                 |                               5 |
| San Diego International AirportÂ (Lindbergh Field)                         | 0620                 |                               4 |
| Los Angeles International Airport                                          | 0735                 |                               4 |
| Minneapolis-Saint Paul International Airport                               | 0730                 |                               4 |
| John F. Kennedy International AirportÂ (New York International Airport)    | 0600                 |                               4 |
| Phoenix Sky Harbor International Airport                                   | 0600                 |                               4 |
+----------------------------------------------------------------------------+----------------------+---------------------------------+
15 rows in set (0.00 sec)
```

3. Indexing Analysis

Advanced Query 1

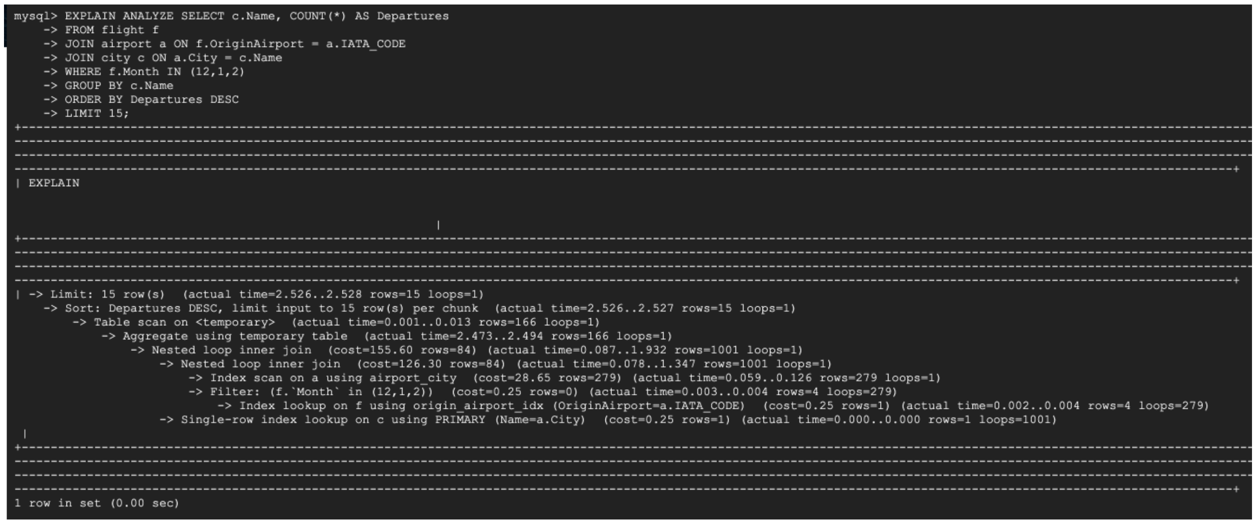
Original costs of the 1st query: 329.56



```
mysql> EXPLAIN ANALYZE SELECT c.Name, COUNT(*) AS Departures
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> JOIN city c ON a.City = c.Name
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY c.Name
    -> ORDER BY Departures DESC
    -> LIMIT 15;
+------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
    -> GROUP BY c.Name
    -> ORDER BY Departures DESC
    -> LIMIT 15;
+------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------+
| EXPLAIN                                                                                                      |
+------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=4.187..4.188 rows=15 loops=1)
    -> Sort: Departures DESC, limit input to 15 row(s) per chunk  (actual time=4.185..4.186 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=0.002..0.027 rows=166 loops=1)
            -> Aggregate using temporary table  (actual time=4.100..4.147 rows=166 loops=1)
                -> Nested loop inner join  (cost=329.56 rows=300) (actual time=0.252..3.442 rows=1001 loops=1)
                    -> Nested loop inner join  (cost=211.21 rows=300) (actual time=0.244..1.266 rows=1001 loops=1)
                        -> Filter: (f.`Month` in (12,1,2))  (cost=106.10 rows=300) (actual time=0.050..0.534 rows=1001 loops=1)
                            -> Index scan on f using flight_rf_1_1  (cost=106.10 rows=1001) (actual time=0.048..0.420 rows=1001 loops=1)
                        -> Single-row index lookup on a using PRIMARY (IATA_CODE=f.OriginAirport)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=1001)
                    -> Single-row index lookup on c using PRIMARY (Name=a.City)  (cost=0.29 rows=1) (actual time=0.002..0.002 rows=1 loops=1001)
 |
+------------------------------------------------------------------------------------------------------------------------
```

1. **Add index to OriginAirport in table flight -> cost: 155.60**

   Analyze: There are many flights with the same OriginAirport, therefore an index of that attribute will reduce the cost of the search for flights with certain OriginAirport and thus reduce the cost of joining table airport and flight on OriginAirport.

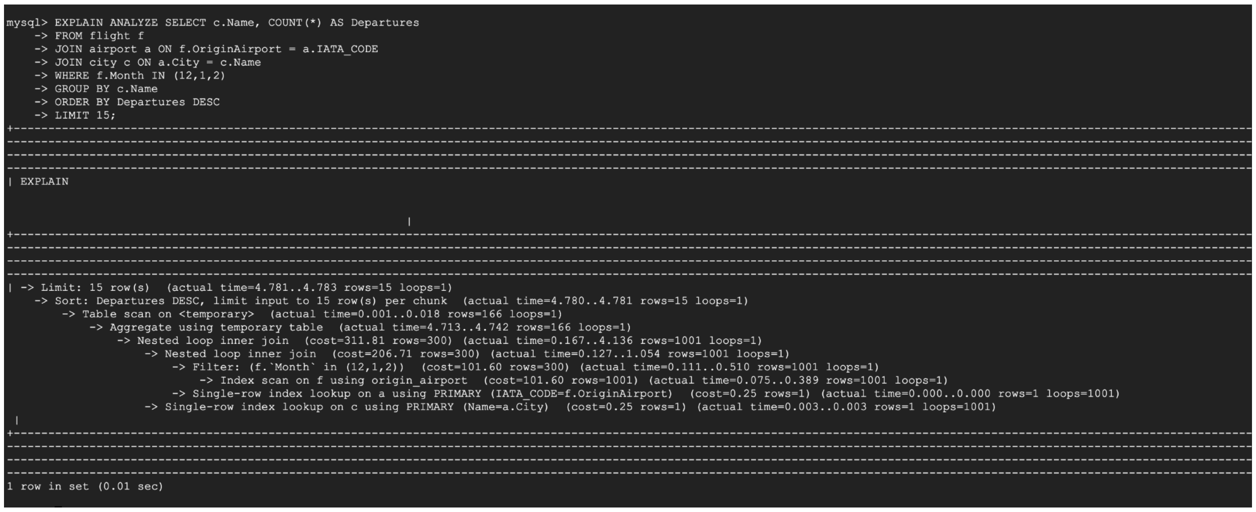


```
mysql> EXPLAIN ANALYZE SELECT c.Name, COUNT(*) AS Departures
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> JOIN city c ON a.City = c.Name
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY c.Name
    -> ORDER BY Departures DESC
    -> LIMIT 15;
+------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------+
| EXPLAIN


                                                        |
+------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=2.526..2.528 rows=15 loops=1)
    -> Sort: Departures DESC, limit input to 15 row(s) per chunk  (actual time=2.526..2.527 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=0.001..0.013 rows=166 loops=1)
            -> Aggregate using temporary table  (actual time=2.473..2.494 rows=166 loops=1)
                -> Nested loop inner join  (cost=155.60 rows=84) (actual time=0.087..1.932 rows=1001 loops=1)
                    -> Nested loop inner join  (cost=126.30 rows=84) (actual time=0.078..1.347 rows=1001 loops=1)
                        -> Index scan on a using airport_city  (cost=28.65 rows=279) (actual time=0.059..0.126 rows=279 loops=1)
                        -> Filter: (f.`Month` in (12,1,2))  (cost=0.25 rows=0) (actual time=0.003..0.004 rows=4 loops=279)
                            -> Index lookup on f using origin_airport_idx (OriginAirport=a.IATA_CODE)  (cost=0.25 rows=1) (actual time=0.002..0.004 rows=4 loops=279)
                    -> Single-row index lookup on c using PRIMARY (Name=a.City)  (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=1001)
 |
+------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

2. **Add index to City in table airport -> cost: 311.81**

   Analyze: Since most cities only have one airport, adding index to the City of airport will not make many differences on the cost.

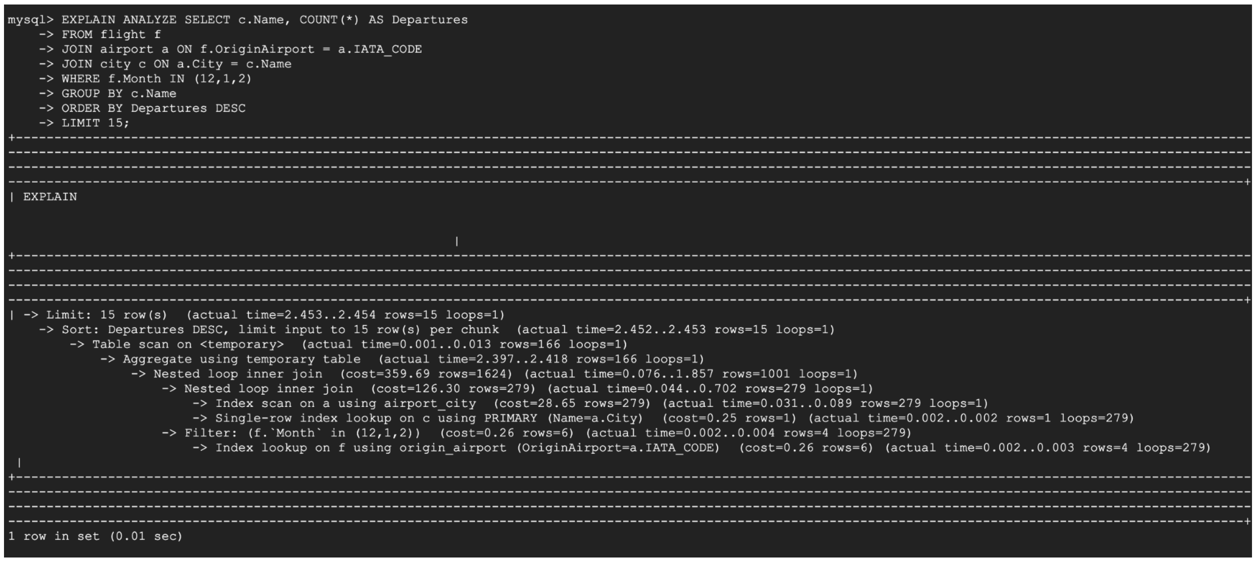


```
mysql> EXPLAIN ANALYZE SELECT c.Name, COUNT(*) AS Departures
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> JOIN city c ON a.City = c.Name
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY c.Name
    -> ORDER BY Departures DESC
    -> LIMIT 15;
+------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
| EXPLAIN


                                                            |
+------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
| -> Limit: 15 row(s)  (actual time=4.781..4.783 rows=15 loops=1)
    -> Sort: Departures DESC, limit input to 15 row(s) per chunk  (actual time=4.780..4.781 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=0.001..0.018 rows=166 loops=1)
            -> Aggregate using temporary table  (actual time=4.713..4.742 rows=166 loops=1)
                -> Nested loop inner join  (cost=311.81 rows=300) (actual time=0.167..4.136 rows=1001 loops=1)
                    -> Nested loop inner join  (cost=206.71 rows=300) (actual time=0.127..1.054 rows=1001 loops=1)
                        -> Filter: (f.`Month` in (12,1,2))  (cost=101.60 rows=300) (actual time=0.111..0.510 rows=1001 loops=1)
                            -> Index scan on f using origin_airport  (cost=101.60 rows=1001) (actual time=0.075..0.389 rows=1001 loops=1)
                        -> Single-row index lookup on a using PRIMARY (IATA_CODE=f.OriginAirport)  (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=1001)
                    -> Single-row index lookup on c using PRIMARY (Name=a.City)  (cost=0.25 rows=1) (actual time=0.003..0.003 rows=1 loops=1001)
 |
+------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
1 row in set (0.01 sec)
```

3. **Add index to Month in table flight -> cost: 359.69**

   Analyze: Since that these 1001 rows data are all from January, the index Month does not improve anything and even cost more.

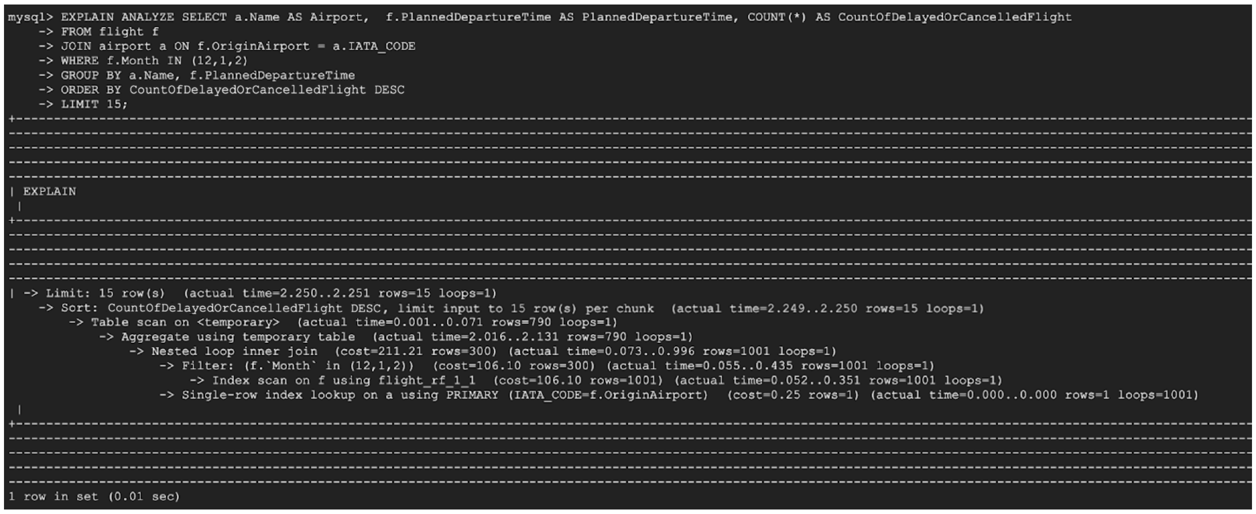


```
mysql> EXPLAIN ANALYZE SELECT c.Name, COUNT(*) AS Departures
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> JOIN city c ON a.City = c.Name
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY c.Name
    -> ORDER BY Departures DESC
    -> LIMIT 15;
+-------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------+
| EXPLAIN


                                                            |
+-------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=2.453..2.454 rows=15 loops=1)
    -> Sort: Departures DESC, limit input to 15 row(s) per chunk  (actual time=2.452..2.453 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=0.001..0.013 rows=166 loops=1)
            -> Aggregate using temporary table  (actual time=2.397..2.418 rows=166 loops=1)
                -> Nested loop inner join  (cost=359.69 rows=1624) (actual time=0.076..1.857 rows=1001 loops=1)
                    -> Nested loop inner join  (cost=126.30 rows=279) (actual time=0.044..0.702 rows=279 loops=1)
                        -> Index scan on a using airport_city  (cost=28.65 rows=279) (actual time=0.031..0.089 rows=279 loops=1)
                        -> Single-row index lookup on c using PRIMARY (Name=a.City)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=279)
                    -> Filter: (f.`Month` in (12,1,2))  (cost=0.26 rows=6) (actual time=0.002..0.004 rows=4 loops=279)
                        -> Index lookup on f using origin_airport (OriginAirport=a.IATA_CODE)  (cost=0.26 rows=6) (actual time=0.002..0.003 rows=4 loops=279)
 |
+-------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------+
1 row in set (0.01 sec)
```

Advanced Query 2

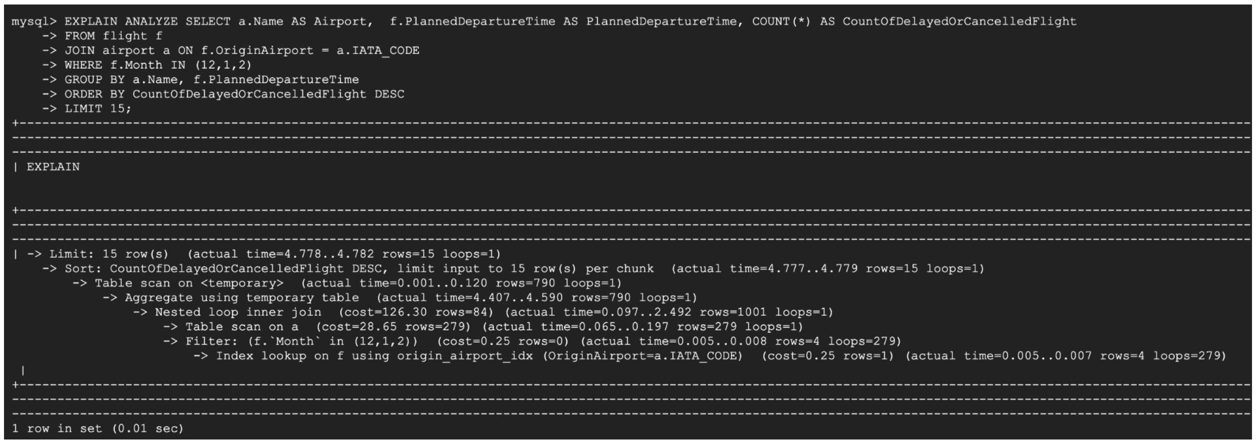
Original costs of the 2nd query: 211.21



```
mysql> EXPLAIN ANALYZE SELECT a.Name AS Airport,  f.PlannedDepartureTime AS PlannedDepartureTime, COUNT(*) AS CountOfDelayedOrCancelledFlight
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY a.Name, f.PlannedDepartureTime
    -> ORDER BY CountOfDelayedOrCancelledFlight DESC
    -> LIMIT 15;
+----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
| EXPLAIN
 |
+----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
| -> Limit: 15 row(s)  (actual time=2.250..2.251 rows=15 loops=1)
    -> Sort: CountOfDelayedOrCancelledFlight DESC, limit input to 15 row(s) per chunk  (actual time=2.249..2.250 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=0.001..0.071 rows=790 loops=1)
            -> Aggregate using temporary table  (actual time=2.016..2.131 rows=790 loops=1)
                -> Nested loop inner join  (cost=211.21 rows=300) (actual time=0.073..0.996 rows=1001 loops=1)
                    -> Filter: (f.`Month` in (12,1,2))  (cost=106.10 rows=300) (actual time=0.055..0.435 rows=1001 loops=1)
                        -> Index scan on f using flight_rf_1_1  (cost=106.10 rows=1001) (actual time=0.052..0.351 rows=1001 loops=1)
                    -> Single-row index lookup on a using PRIMARY (IATA_CODE=f.OriginAirport)  (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=1001)
 |
+----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
1 row in set (0.01 sec)
```

1. **Add index to OriginAirport in table flight: 126.30**

   Analyze: Same as the situation in the analysis in the 1st advanced query. There are many flights with the same OriginAirport, therefore an index of that attribute will reduce the cost of the search for flights with certain OriginAirport and thus reduce the cost of joining table airport and flight on OriginAirport.

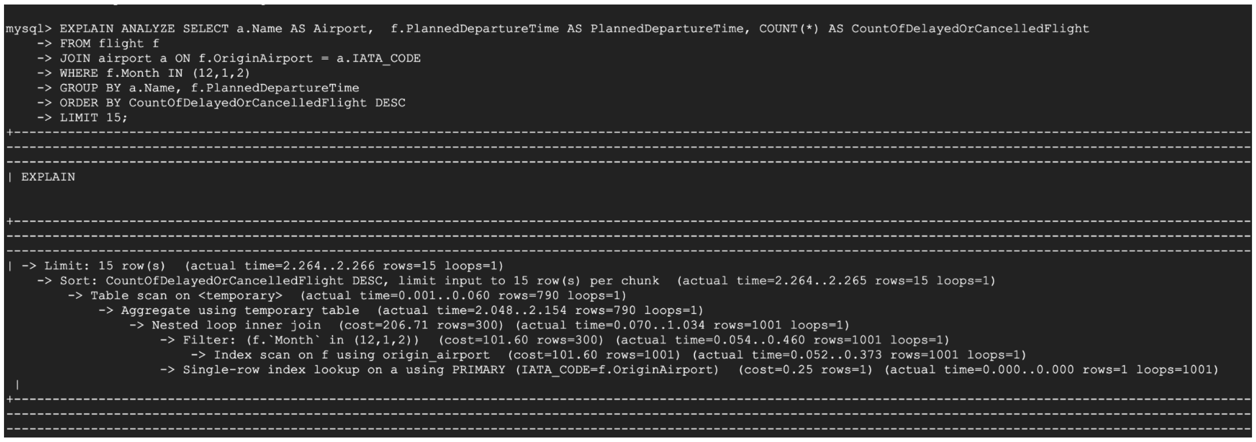


```
mysql> EXPLAIN ANALYZE SELECT a.Name AS Airport,  f.PlannedDepartureTime AS PlannedDepartureTime, COUNT(*) AS CountOfDelayedOrCancelledFlight
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY a.Name, f.PlannedDepartureTime
    -> ORDER BY CountOfDelayedOrCancelledFlight DESC
    -> LIMIT 15;
+----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
| EXPLAIN


+----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
| -> Limit: 15 row(s)  (actual time=4.778..4.782 rows=15 loops=1)
    -> Sort: CountOfDelayedOrCancelledFlight DESC, limit input to 15 row(s) per chunk  (actual time=4.777..4.779 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=0.001..0.120 rows=790 loops=1)
            -> Aggregate using temporary table  (actual time=4.407..4.590 rows=790 loops=1)
                -> Nested loop inner join  (cost=126.30 rows=84) (actual time=0.097..2.492 rows=1001 loops=1)
                    -> Table scan on a  (cost=28.65 rows=279) (actual time=0.065..0.197 rows=279 loops=1)
                    -> Filter: (f.`Month` in (12,1,2))  (cost=0.25 rows=0) (actual time=0.005..0.008 rows=4 loops=279)
                        -> Index lookup on f using origin_airport_idx (OriginAirport=a.IATA_CODE)  (cost=0.25 rows=1) (actual time=0.005..0.007 rows=4 loops=279)
 |
+----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------------------
1 row in set (0.01 sec)
```

2. **Add index to PlannedDepartureTime in table flight: 206.71**

   Analyze: When a query involves grouping and ordering, the database needs to perform sorting and aggregation operations. In this case, the database needs to group flights by both Airport and PlannedDepartureTime, count the number of delayed or canceled flights for each group, and then sort the results based on the count. The index on PlannedDepartureTime may not be sufficient to optimize the grouping and counting operations.

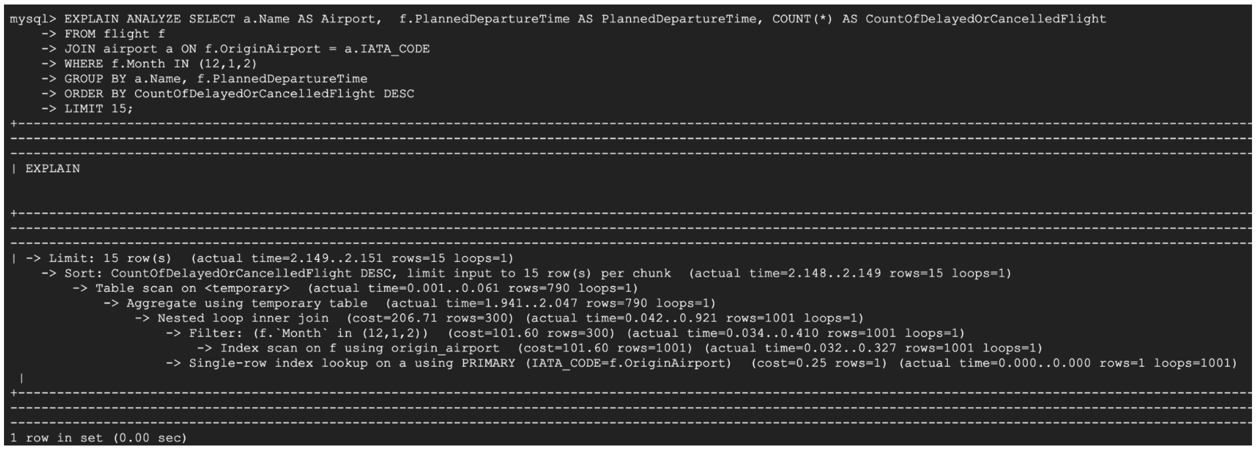


```
mysql> EXPLAIN ANALYZE SELECT a.Name AS Airport,  f.PlannedDepartureTime AS PlannedDepartureTime, COUNT(*) AS CountOfDelayedOrCancelledFlight
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY a.Name, f.PlannedDepartureTime
    -> ORDER BY CountOfDelayedOrCancelledFlight DESC
    -> LIMIT 15;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
| EXPLAIN


+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
| -> Limit: 15 row(s)  (actual time=2.264..2.266 rows=15 loops=1)
    -> Sort: CountOfDelayedOrCancelledFlight DESC, limit input to 15 row(s) per chunk  (actual time=2.264..2.265 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=0.001..0.060 rows=790 loops=1)
            -> Aggregate using temporary table  (actual time=2.048..2.154 rows=790 loops=1)
                -> Nested loop inner join  (cost=206.71 rows=300) (actual time=0.070..1.034 rows=1001 loops=1)
                    -> Filter: (f.`Month` in (12,1,2))  (cost=101.60 rows=300) (actual time=0.054..0.460 rows=1001 loops=1)
                        -> Index scan on f using origin_airport  (cost=101.60 rows=1001) (actual time=0.052..0.373 rows=1001 loops=1)
                    -> Single-row index lookup on a using PRIMARY (IATA_CODE=f.OriginAirport)  (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=1001)
 |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
```

3. **Add index to Name in table airport: 206.71**

   Analyze: Since nearly each airport has its unique name, this index might not improve anything at all.



```
mysql> EXPLAIN ANALYZE SELECT a.Name AS Airport,  f.PlannedDepartureTime AS PlannedDepartureTime, COUNT(*) AS CountOfDelayedOrCancelledFlight
    -> FROM flight f
    -> JOIN airport a ON f.OriginAirport = a.IATA_CODE
    -> WHERE f.Month IN (12,1,2)
    -> GROUP BY a.Name, f.PlannedDepartureTime
    -> ORDER BY CountOfDelayedOrCancelledFlight DESC
    -> LIMIT 15;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
| EXPLAIN


+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
| -> Limit: 15 row(s)  (actual time=2.149..2.151 rows=15 loops=1)
    -> Sort: CountOfDelayedOrCancelledFlight DESC, limit input to 15 row(s) per chunk  (actual time=2.148..2.149 rows=15 loops=1)
        -> Table scan on <temporary>  (actual time=0.001..0.061 rows=790 loops=1)
            -> Aggregate using temporary table  (actual time=1.941..2.047 rows=790 loops=1)
                -> Nested loop inner join  (cost=206.71 rows=300) (actual time=0.042..0.921 rows=1001 loops=1)
                    -> Filter: (f.`Month` in (12,1,2))  (cost=101.60 rows=300) (actual time=0.034..0.410 rows=1001 loops=1)
                        -> Index scan on f using origin_airport  (cost=101.60 rows=1001) (actual time=0.032..0.327 rows=1001 loops=1)
                    -> Single-row index lookup on a using PRIMARY (IATA_CODE=f.OriginAirport)  (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=1001)
 |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------
1 row in set (0.00 sec)
```

Conclusion: Based on experiments and analysis above, adding an index to *OriginAirport* in table *flight* is a good design, improving both advanced queries.