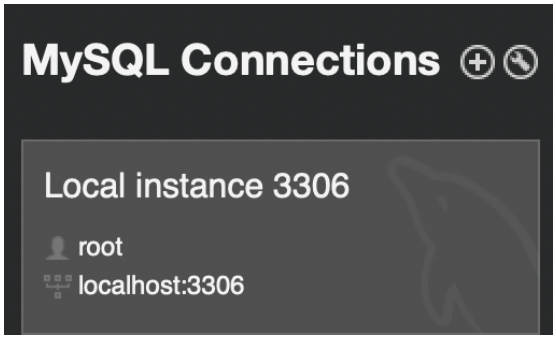
https://canvas.illinois.edu/courses/38800/assignments/746915
https://www.kaggle.com/datasets/usdot/flight-delays?resource=download&select=airlines.csv

---

Showing Database Connection:



Tables and the Total Number of Records:

SELECT COUNT(*) FROM Aircrafts;

COUNT(*)
1023

	tailNumber	company	aircraftModel
		Lockheed Martin	F-16 Fighting Falcon
	N008AA	Northrop Grumman	E-2D Advanced Hawkeye
	N012AA	Raytheon	Bonanza
	N019AA	Northrop Grumman	E-2D Advanced Hawkeye
	N021AA	Leonardo	AW245
	N023AA	Leonardo	AW298
	N025AA	Boeing	Boeing 787
	N107SY	Leonardo	AW207
	N11107	Safran	?LEAP-1A
	N11121	Leonardo	AW234
	N11140	Raytheon	Bonanza
	N11150	Leonardo	AW202
	N11206	Boeing	Boeing 801
	N114SY	Boeing	Boeing 759
	N118SY	Leonardo	AW252
	N12122	Leonardo	AW225
	N12142	Boeing	Boeing 726
	N124SY	Northrop Grumman	E-2D Advanced Hawkeye
	N12552	Boeing	Boeing 774
	N12563	Safran	?LEAP-1A
	N128UW	Safran	?LEAP-1A
	N12921	Boeing	Boeing 754
	N12967	Airbus	Airbus A392
	N12996	General Electric	Electric Aircraft

SELECT COUNT(*) FROM Airlines;

COUNT(*)
14

IATAcodeAirline	airline
AS	Alaska Airlines Inc.
AA	American Airlines Inc.
MQ	American Eagle Airlines Inc.
EV	Atlantic Southeast Airlines
DL	Delta Air Lines Inc.
F9	Frontier Airlines Inc.
HA	Hawaiian Airlines Inc.
B6	JetBlue Airways
OO	Skywest Airlines Inc.
WN	Southwest Airlines Co.
NK	Spirit Air Lines
UA	United Air Lines Inc.
US	US Airways Inc.
VX	Virgin America

SELECT COUNT(*) FROM Airports;

COUNT(*)
1099

IATAcodeAirport	airport	city	state	country
AAA	Anaa Airport	Anaa		French Polynesia
ABD	Abadan Airport	Abadan		Iran
ABE	Lehigh Valley International Airport	Allentown	PA	United States of America
ABI	Abilene Regional Airport	Abilene	TX	United States of America
ABQ	Albuquerque International Sunport	Albuquerque	NM	United States of America
ABR	Aberdeen Regional Airport	Aberdeen	SD	United States of America
ABT	Al Baha Airport	El-baha		Saudi Arabia
ABY	Southwest Georgia Regional Airport	Albany	GA	United States of America
ACA	General Juan N Alvarez International Airport	Acapulco		Mexico
ACH	St Gallen Altenrhein Airport	Altenrhein		Switzerland
ACK	Nantucket Memorial Airport	Nantucket	MA	United States of America
ACT	Waco Regional Airport	Waco	TX	United States of America
ACV	Arcata Airport	Arcata/Eur…	CA	United States of America
ACY	Atlantic City International Airport	Atlantic City	NJ	United States of America
ACZ	Zabol Airport	Zabol		Iran
ADA	Adana Airport	Adana		Turkey
ADB	Adnan Menderes International Airport	Izmir		Turkey
ADJ	Amman-Marka International Airport	Amman		Jordan
ADK	Adak Airport	Adak	AK	United States of America
ADQ	Kodiak Airport	Kodiak	AK	United States of America
AEP	Jorge Newbery Airpark	Buenos Aires		Argentina
AEU	Abu Musa Island Airport	Abumusa I.		Iran
AEX	Alexandria International Airport	Alexandria	LA	United States of America

---

SELECT COUNT(*) FROM Flights;

COUNT(*)
1104

flightNumb...	airline	tailNumber	year	month	day	dayOfTheWeek	originAirp...	destinationAirp...	departureDelay	cancelled
3	OO	N812SK	2015	6	14	5	SEA	PDX	-13	0
8	AS	N464AS	2015	8	13	3	SEA	EWR	9	0
9	WN	N268WN	2015	2	1	2	PHL	FLL	-4	0
10	HA	N389HA	2015	4	9	1	HNL	LAX	10	0
12	AS	N403AS	2015	9	20	2	SLC	LAX	17	0
15	AA	N477AA	2015	4	25	3	MEM	DFW	-9	0
16	B6	N273JB	2015	5	19	1	IAD	JFK	-6	0
17	NK	N514NK	2015	7	23	3	LAS	IAH	-4	0
19	DL	N933DN	2015	7	29	6	MSP	TUS	-2	0
22	WN	N8302F	2015	6	29	6	MDW	MCO	0	0
23	MQ	N836MQ	2015	10	26	6	LEX	DFW		1
24	UA	N37466	2015	7	11	5	EWR	FLL	-8	0
30	AS	N514AS	2015	10	2	6	PDX	BOS	-9	0
39	MQ	N511MQ	2015	8	18	6	ELP	ORD		1
46	WN	N962WN	2015	11	4	6	SAT	BNA	-5	0
52	DL	N812DN	2015	7	26	2	LAX	DTW	-7	0
55	MQ	N667MQ	2015	8	9	5	DFW	DSM	4	0
57	UA	N76522	2015	6	30	2	SFO	IAH	-5	0
60	AA	N3DUAA	2015	11	28	1	SFO	DFW	-7	0
61	OO	N824AS	2015	4	23	5	SFO	BOI	29	0
62	AS	N792AS	2015	1	16	1	ANC	JNU	-12	0
68	AA	N857AA	2015	1	8	3	MIA	DEN	4	0
70	EV	N695CA	2015	11	1	1	BTR	ATL	0	0

SELECT COUNT(*) FROM userRating;

COUNT(*)
25

	userID	airline	userRating
	1270	B6	3
	1756	EV	4
	3321	OO	2
	3455	US	2
	3653	F9	4
	3994	MQ	4
	4200	UA	4
	4262	F9	5
	5065	WN	4
	5220	EV	4
	5491	OO	5
	5772	VX	5
	5780	NK	5
	6348	US	2
	6460	AS	1
	6549	NK	3
	6676	WN	2
	6844	HA	2
	6966	B6	4
	8178	AS	4
	8412	AA	3
	8900	DL	2
	9290	AA	1

SELECT COUNT(*) FROM Users;

COUNT(*)
197

userID	username	password	phoneNumber
9	twojxlev	3kaah89i	51944624
110	xodpwnqf	7swfdg83	879864839
118	rbrojjmr	jlcxenp4	606874999
198	nywqxayu	yxdeemfv	858229897
234	yrkwrivl	iaxfaymq	267972132
276	sbggiaxo	oz530gad	899798011
286	ldcfojcs	2a6dbeiy	355520527
315	ibwluhyj	4xhybqq7	802789714
445	rmvyjeui	w3yalil1	763439028
554	nlasxdpj	eta8lk93	803187577
613	smvkdgzq	r8u2uckq	164375562
632	povjoyqq	c215eihp	348774115
838	xipvujgb	0n7g53fz	225485445
879	hbayvhgj	231lilas	749468507
952	nygzjbyv	d8k00ord	274830691
979	itenyvkd	0svcool6	89469771
1055	kbhyytkk	8nyclii5	643712144
1071	xcctliue	mwbe6y4o	893592659
1072	vzvjcrdn	m23iyz1y	123356087
1093	bknxzuli	c27r1h4a	810933516
1094	nnodvbbr	0jb1zlrx	174175378
1101	wctrambq	3vu3zmm7	52187194
1129	ylcewuyw	lywg180q	143921094

DDL Commands:

```
DROP DATABASE flightsdatabase2;
CREATE DATABASE flightsdatabase2;
USE flightsdatabase2;

DROP TABLE IF EXISTS Users;
CREATE TABLE Users (
        userID INT PRIMARY KEY,
        username VARCHAR(100),
        password VARCHAR(100),
        phoneNumber VARCHAR(15)
);

DROP TABLE IF EXISTS Airlines;
CREATE TABLE Airlines (
```

```
	IATAcodeAirline VARCHAR(3) PRIMARY KEY,
	airline VARCHAR(100)
);

DROP TABLE IF EXISTS Airports;
CREATE TABLE Airports (
	IATAcodeAirport VARCHAR(3) PRIMARY KEY,
	airport VARCHAR(200),
	city VARCHAR(100),
	state VARCHAR(100),
	country VARCHAR(100)
);

DROP TABLE IF EXISTS Aircrafts;
CREATE TABLE Aircrafts (
	tailNumber VARCHAR(100) PRIMARY KEY,
	company VARCHAR(100),
	aircraftModel VARCHAR(100)
);

DROP TABLE IF EXISTS Flights;
CREATE TABLE Flights (
	flightNumber INT PRIMARY KEY,
	airline VARCHAR(3),
	tailNumber VARCHAR(100),
	year INT,
	month VARCHAR(15),
	day VARCHAR(2),
	dayOfTheWeek VARCHAR(15),
	originAirport VARCHAR(3),
	destinationAirport VARCHAR(3),
	departureDelay VARCHAR(10),
	cancelled INT,
	FOREIGN KEY (airline) REFERENCES Airlines(IATAcodeAirline),
	FOREIGN KEY (originAirport) REFERENCES Airports(IATAcodeAirport),
	FOREIGN KEY (destinationAirport) REFERENCES Airports(IATAcodeAirport),
	FOREIGN KEY (tailNumber) REFERENCES Aircrafts(tailNumber)
);
```

```
DROP TABLE IF EXISTS userRating;
CREATE TABLE userRating (
        userID INT,
        airline VARCHAR(3),
        userRating INT,
        PRIMARY KEY (userID, airline),
        FOREIGN KEY (userID) REFERENCES Users(userID),
        FOREIGN KEY (airline) REFERENCES Airlines(IATAcodeAirline)
);
```

Advanced Queries:

This advanced query gives the total flights for each airline in the month of December. There are 14 airlines, so we have a total of 14 rows in the output.

```
SELECT A.airline, COUNT(F.flightNumber) AS TotalFlights
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.month = '12'
GROUP BY A.airline
ORDER BY TotalFlights DESC;
```

airline	TotalFlights
Skywest Airlines Inc.	12
Delta Air Lines Inc.	11
American Airlines Inc.	11
United Air Lines Inc.	11
JetBlue Airways	8
Southwest Airlines Co.	7
Atlantic Southeast Airlines	6
US Airways Inc.	6
Spirit Air Lines	5
Hawaiian Airlines Inc.	3
American Eagle Airlines…	3
Alaska Airlines Inc.	3
Virgin America	1

This advanced query gives airlines with their departure delays in descending order. There are 14 airlines, so we have a total of 14 rows in the output. You can see which airline experiences the most departure delays from this result.

```
SELECT A.IATAcodeAirline, COUNT(F.departureDelay) AS NumDepartureDelay
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
GROUP BY A.IATAcodeAirline
ORDER BY NumDepartureDelay DESC;
```

	IATAcodeAirline	NumDepartureDel...
	AA	154
	DL	132
	OO	131
	UA	107
	B6	104
	EV	83
	MQ	80
	WN	79
	AS	67
	US	64
	NK	41
	F9	34
	HA	24
	VX	4

<u>Indexing:</u>

Changed Advanced Query to give total flights for each airline on Christmas Day, in order to use multiple non-PK or FK index options

```
SELECT A.airline, COUNT(F.flightNumber) AS TotalFlights
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015' AND F.month = '12' AND F.day = '25'
GROUP BY A.airline
ORDER BY TotalFlights DESC;
```

airline	TotalFlights
JetBlue Airways	1
Delta Air Lines Inc.	1
US Airways Inc.	1

Changed query to include cancellations as well as departure delays in 2015 for every airline

```
SELECT A.IATAcodeAirline, COUNT(F.departureDelay) AS NumDepartureDelay,
COUNT(F.cancelled) AS NumCancellations
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015'
GROUP BY A.IATAcodeAirline
ORDER BY NumDepartureDelay DESC, NumCancellations DESC;
```

IATAcodeAirline	NumDepartureDel...	NumCancellatio...
AA	154	154
DL	132	132
OO	131	131
UA	107	107
B6	104	104
EV	83	83
MQ	80	80
WN	79	79
AS	67	67
US	64	64
NK	41	41
F9	34	34
HA	24	24
VX	4	4

Indexing:

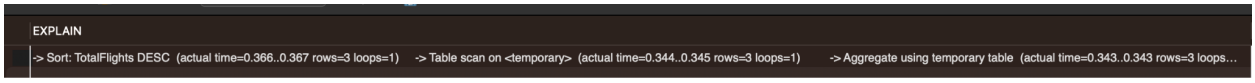
FIRST ADVANCED QUERY

SHOW INDEX FROM Flights;

EXPLAIN ANALYZE

```
SELECT A.airline, COUNT(F.flightNumber) AS TotalFlights
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015' AND F.month = '12' AND F.day = '25'
GROUP BY A.airline
ORDER BY TotalFlights DESC;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
Flights	0	PRIMARY	1	flightNumber	A	0	NULL	NULL		BTREE			YES	NULL
Flights	1	airline	1	airline	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	originAirport	1	originAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	destinationAirport	1	destinationAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	tailNumber	1	tailNumber	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_month	1	month	A	12	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_departureDelay	1	departureDelay	A	98	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_year	1	year	A	1	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_cancel	1	cancelled	A	2	NULL	NULL	YES	BTREE			YES	NULL



'-> Sort: TotalFlights DESC (actual time=0.593..0.593 rows=3 loops=1)\n

-> Table scan on <temporary> (actual time=0.587..0.588 rows=3 loops=1)\n

-> Aggregate using temporary table (actual time=0.587..0.587 rows=3 loops=1)\n

-> Nested loop inner join (**cost=2.77** rows=3) (actual time=0.502..0.562 rows=3 loops=1)\n

-> Filter: ((f.`day` = \'25\') and (f.`month` = \'12\') and (f.`year` = 2015) and (f.airline is not null)) (**cost=1.72** rows=3) (actual time=0.47..0.526 rows=3 loops=1)\n

-> Intersect rows sorted by row ID (**cost=1.72** rows=3.55) (actual time=0.468..0.523 rows=3 loops=1)\n

-> Index range scan on F using idx_day over (day = \'25\') (**cost=0.27** rows=45) (actual time=0.252..0.262 rows=45 loops=1)\n

-> Index range scan on F using idx_month over (month = \'12\') (**cost=0.425** rows=87) (actual time=0.0123..0.0334 rows=87 loops=1)\n

-> Single-row index lookup on A using PRIMARY (IATAcodeAirline=f.airline) (**cost=0.283** rows=1) (actual time=0.00692..0.00694 rows=1 loops=3)\n'

CREATE INDEX idx_year ON Flights(year);
SHOW INDEX FROM Flights;
EXPLAIN ANALYZE
SELECT A.airline, COUNT(F.flightNumber) AS TotalFlights
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015' AND F.month = '12' AND F.day = '25'
GROUP BY A.airline
ORDER BY TotalFlights DESC;

DROP INDEX idx_year ON Flights;

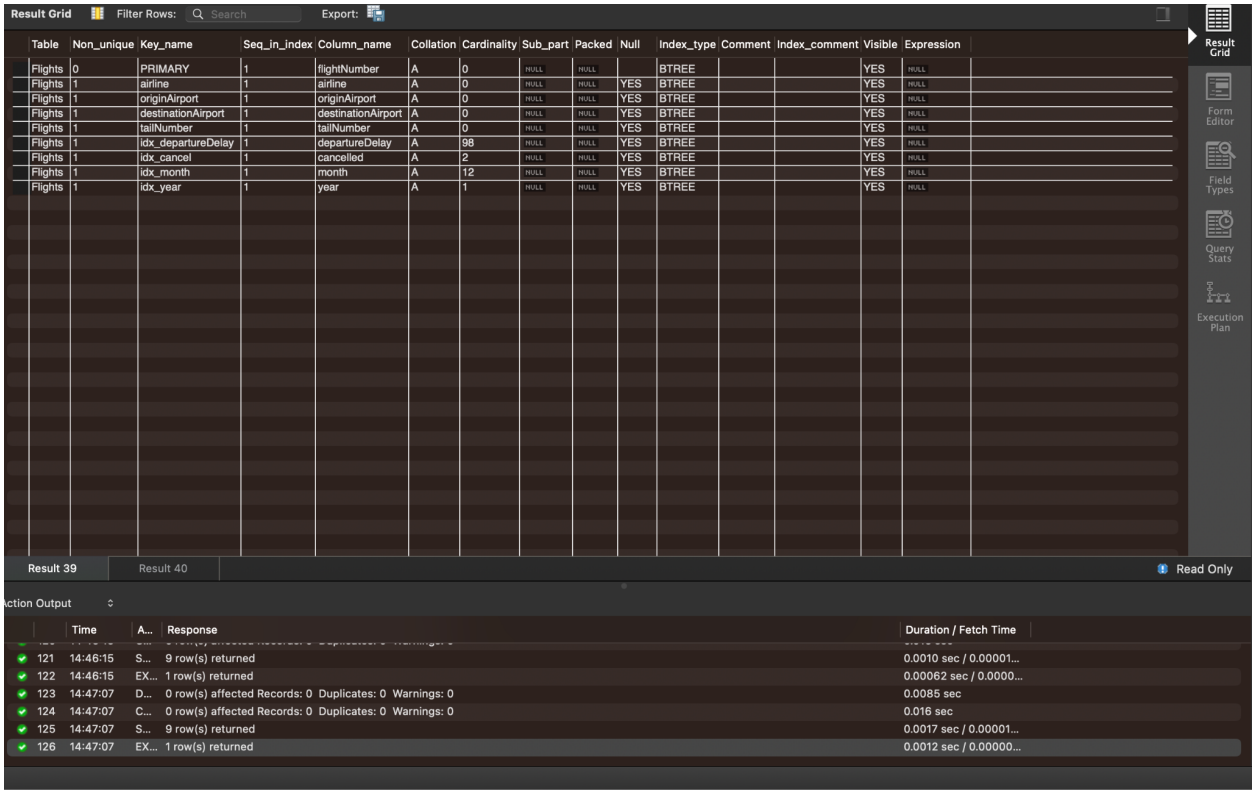


Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
Flights	0	PRIMARY	1	flightNumber	A	0	NULL	NULL		BTREE			YES	NULL
Flights	1	airline	1	airline	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	originAirport	1	originAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	destinationAirport	1	destinationAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	tailNumber	1	tailNumber	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_departureDelay	1	departureDelay	A	98	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_cancel	1	cancelled	A	2	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_month	1	month	A	12	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_year	1	year	A	1	NULL	NULL	YES	BTREE			YES	NULL

	Time	A...	Response	Duration / Fetch Time
121	14:46:15	S...	9 row(s) returned	0.0010 sec / 0.00001...
122	14:46:15	EX...	1 row(s) returned	0.00062 sec / 0.0000...
123	14:47:07	D...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.0085 sec
124	14:47:07	C...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
125	14:47:07	S...	9 row(s) returned	0.0017 sec / 0.00001...
126	14:47:07	EX...	1 row(s) returned	0.0012 sec / 0.00000...

EXPLAIN
-> Sort: TotalFlights DESC (actual time=0.317..0.317 rows=3 loops=1) -> Table scan on <temporary> (actual time=0.306..0.307 rows=3 loops=1) -> Aggregate using temporary table (actual time=0.305..0.305 rows=3 loops...

'-> Sort: TotalFlights DESC (actual time=0.378..0.378 rows=3 loops=1)\n

 -> Table scan on <temporary> (actual time=0.37..0.371 rows=3 loops=1)\n

 -> Aggregate using temporary table (actual time=0.37..0.37 rows=3 loops=1)\n

 -> Nested loop inner join (cost=2.77 rows=3) (actual time=0.284..0.342 rows=3 loops=1)\n

-> Filter: ((f.`day` = \'25\') and (f.`month` = \'12\') and (f.`year` = 2015) and (f.airline is not null)) (cost=1.72 rows=3) (actual time=0.0574..0.109 rows=3 loops=1)\n

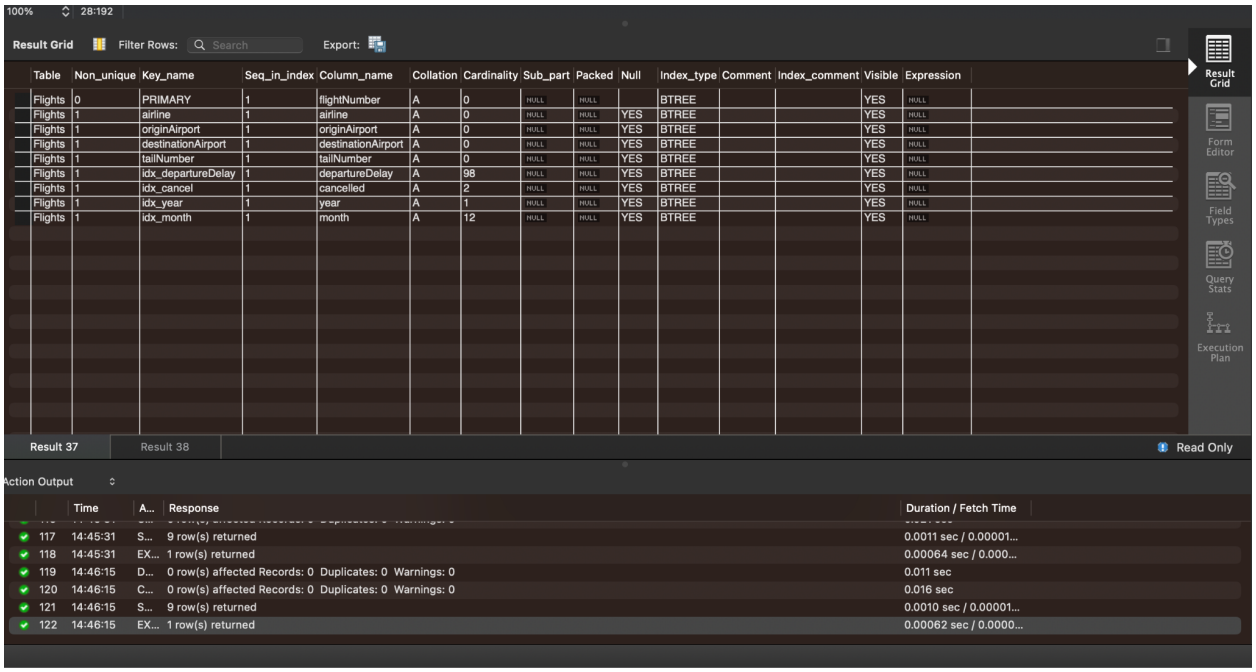
 -> Intersect rows sorted by row ID (cost=1.72 rows=3.55) (actual time=0.0557..0.106 rows=3 loops=1)\n

 -> Index range scan on F using idx_day over (day = \'25\') (cost=0.27 rows=45) (actual time=0.0289..0.0405 rows=45 loops=1)\n

-> Index range scan on F using idx_month over (month = \'12\') (cost=0.425 rows=87) (actual time=0.0095..0.0316 rows=87 loops=1)\n

 -> Single-row index lookup on A using PRIMARY (IATAcodeAirline=f.airline) (cost=0.283 rows=1) (actual time=0.0249..0.025 rows=1 loops=3)\n'

DROP INDEX idx_month ON Flights;
CREATE INDEX idx_month ON Flights(month);
SHOW INDEX FROM Flights;
EXPLAIN ANALYZE
SELECT A.airline, COUNT(F.flightNumber) AS TotalFlights
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015' AND F.month = '12' AND F.day = '25'
GROUP BY A.airline
ORDER BY TotalFlights DESC;



'-> Sort: TotalFlights DESC (actual time=0.13..0.13 rows=3 loops=1)\n

-> Table scan on <temporary> (actual time=0.125..0.126 rows=3 loops=1)\n

-> Aggregate using temporary table (actual time=0.124..0.124 rows=3 loops=1)\n

-> Nested loop inner join (cost=2.77 rows=3) (actual time=0.0643..0.116 rows=3 loops=1)\n

-> Filter: ((f.`day` = \'25\') and (f.`month` = \'12\') and (f.`year` = 2015) and (f.airline is not null)) (cost=1.72 rows=3) (actual time=0.0482..0.0954 rows=3 loops=1)\n

-> Intersect rows sorted by row ID (cost=1.72 rows=3.55) (actual time=0.047..0.0935 rows=3 loops=1)\n

-> Index range scan on F using idx_day over (day = \'25\') (cost=0.27 rows=45) (actual time=0.0219..0.032 rows=45 loops=1)\n

-> Index range scan on F using idx_month over (month = \'12\') (cost=0.425 rows=87) (actual time=0.0107..0.0309 rows=87 loops=1)\n

-> Single-row index lookup on A using PRIMARY (IATAcodeAirline=f.airline) (cost=0.283 rows=1) (actual time=0.0061..0.00612 rows=1 loops=3)\n'

DROP INDEX idx_day ON Flights;
CREATE INDEX idx_day ON Flights(day);
SHOW INDEX FROM Flights;
EXPLAIN ANALYZE
SELECT A.airline, COUNT(F.flightNumber) AS TotalFlights
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015' AND F.month = '12' AND F.day = '25'
GROUP BY A.airline
ORDER BY TotalFlights DESC;

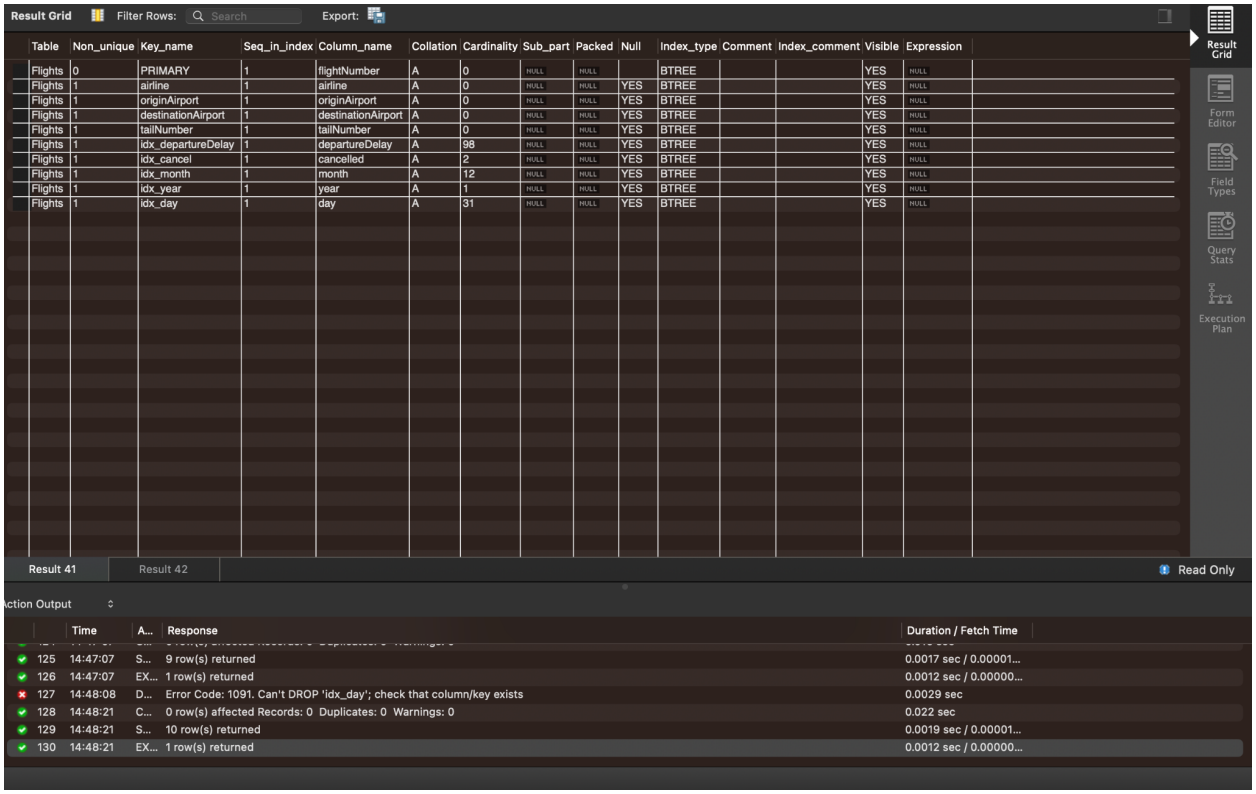


Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
Flights	0	PRIMARY	1	flightNumber	A	0	NULL	NULL		BTREE			YES	NULL
Flights	1	airline	1	airline	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	originAirport	1	originAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	destinationAirport	1	destinationAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	tailNumber	1	tailNumber	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_departureDelay	1	departureDelay	A	98	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_cancel	1	cancelled	A	2	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_month	1	month	A	12	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_year	1	year	A	1	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_day	1	day	A	31	NULL	NULL	YES	BTREE			YES	NULL

	Time	A...	Response	Duration / Fetch Time
125	14:47:07	S...	9 row(s) returned	0.0017 sec / 0.00001...
126	14:47:07	EX...	1 row(s) returned	0.0012 sec / 0.00000...
127	14:48:08	D...	Error Code: 1091. Can't DROP 'idx_day'; check that column/key exists	0.0029 sec
128	14:48:21	C...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.022 sec
129	14:48:21	S...	10 row(s) returned	0.0019 sec / 0.00001...
130	14:48:21	EX...	1 row(s) returned	0.0012 sec / 0.00000...

'-> Sort: TotalFlights DESC (actual time=0.121..0.122 rows=3 loops=1)\n

-> Table scan on <temporary> (actual time=0.117..0.117 rows=3 loops=1)\n

-> Aggregate using temporary table (actual time=0.116..0.116 rows=3 loops=1)\n

-> Nested loop inner join (cost=2.77 rows=3) (actual time=0.0578..0.109 rows=3 loops=1)\n

-> Filter: ((f.`day` = \'25\') and (f.`month` = \'12\') and (f.`year` = 2015) and (f.airline is not null)) (cost=1.72 rows=3) (actual time=0.0437..0.0896 rows=3 loops=1)\n

-> Intersect rows sorted by row ID (cost=1.72 rows=3.55) (actual time=0.0424..0.0876 rows=3 loops=1)\n

-> Index range scan on F using idx_day over (day = \'25\') (cost=0.27 rows=45) (actual time=0.0177..0.0279 rows=45 loops=1)\n

-> Index range scan on F using idx_month over (month = \'12\') (cost=0.425 rows=87) (actual time=0.012..0.032 rows=87 loops=1)\n

-> Single-row index lookup on A using PRIMARY (IATAcodeAirline=f.airline) (cost=0.283 rows=1) (actual time=0.00599..0.00601 rows=1 loops=3)\n'

Explanation:
Based on the actual time taken for the query, the most suitable indexing option would be doing an index range scan based on **idx_month**, since the original time taken for running the advanced query without any indexes runs at about 0.5, and with it, reduces the time range (for nested loop inner join and intersect rows by sorted by row ID) to a range around 0.05. While **idx_day** reduces the time range for the same procedures to near the same amount of time, the overall time taken for **idx_month** comes to about 0.00062, as opposed to 0.0012 for **idx_day**. Further, while the overall cost from **idx_month**'s index scan remains higher, it also has a greater number of rows than **idx_day**, and dividing, the cost per row is far lower than **idx_day**, making **idx_month** the more efficient choice for indexing for this query.

SECOND ADVANCED QUERY

SHOW INDEX FROM Flights;
EXPLAIN ANALYZE

```
SELECT A.IATAcodeAirline, COUNT(F.departureDelay) AS NumDepartureDelay,
COUNT(F.cancelled) AS NumCancellations
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015'
GROUP BY A.IATAcodeAirline
ORDER BY NumDepartureDelay DESC, NumCancellations DESC;
```

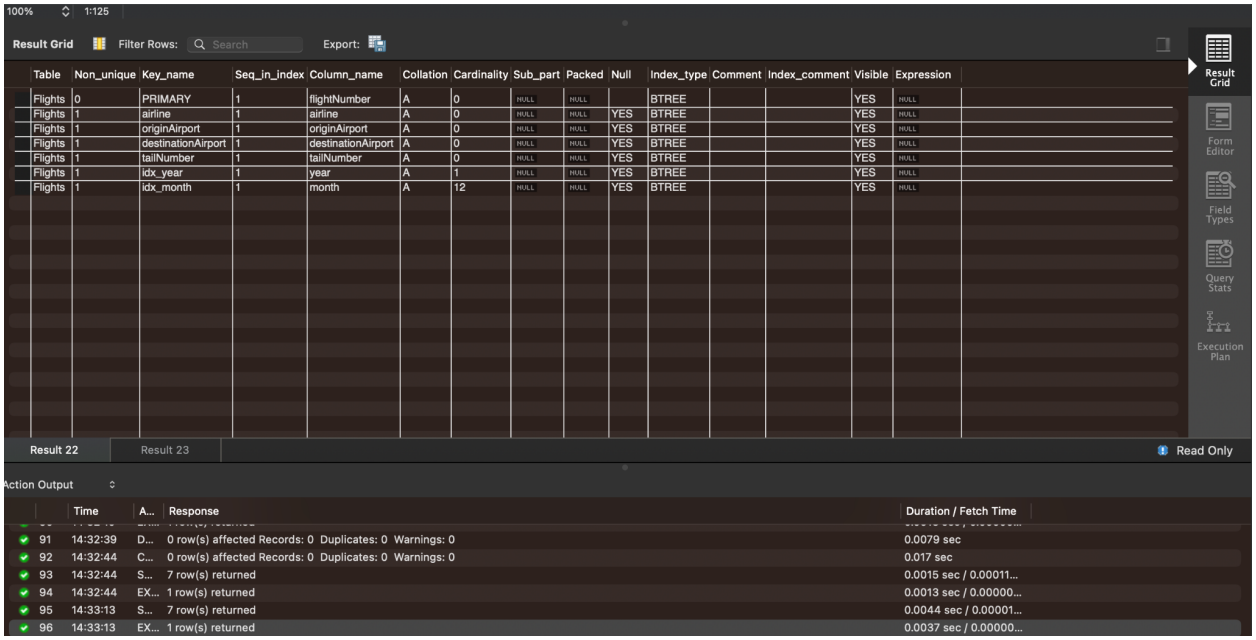
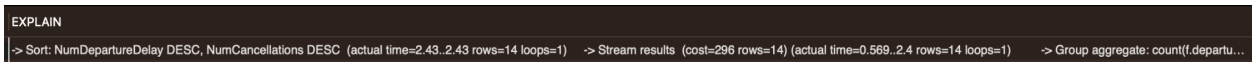


Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
Flights	0	PRIMARY	1	flightNumber	A	0	NULL	NULL		BTREE			YES	NULL
Flights	1	airline	1	airline	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	originAirport	1	originAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	destinationAirport	1	destinationAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	tailNumber	1	tailNumber	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_year	1	year	A	1	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_month	1	month	A	12	NULL	NULL	YES	BTREE			YES	NULL

Result 22 | Result 23 | Read Only

Action Output

	Time	A...	Response	Duration / Fetch Time
91	14:32:39	D...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.0079 sec
92	14:32:44	C...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.017 sec
93	14:32:44	S...	7 row(s) returned	0.0015 sec / 0.00011...
94	14:32:44	EX...	1 row(s) returned	0.0013 sec / 0.00000...
95	14:33:13	S...	7 row(s) returned	0.0044 sec / 0.00001...
96	14:33:13	EX...	1 row(s) returned	0.0037 sec / 0.00000...



DROP INDEX idx_departureDelay ON Flights;

CREATE INDEX idx_departureDelay ON Flights(departureDelay);

SHOW INDEX FROM Flights;

EXPLAIN ANALYZE

SELECT A.IATAcodeAirline, COUNT(F.departureDelay) AS NumDepartureDelay,
COUNT(F.cancelled) AS NumCancellations
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015'
GROUP BY A.IATAcodeAirline
ORDER BY NumDepartureDelay DESC, NumCancellations DESC;

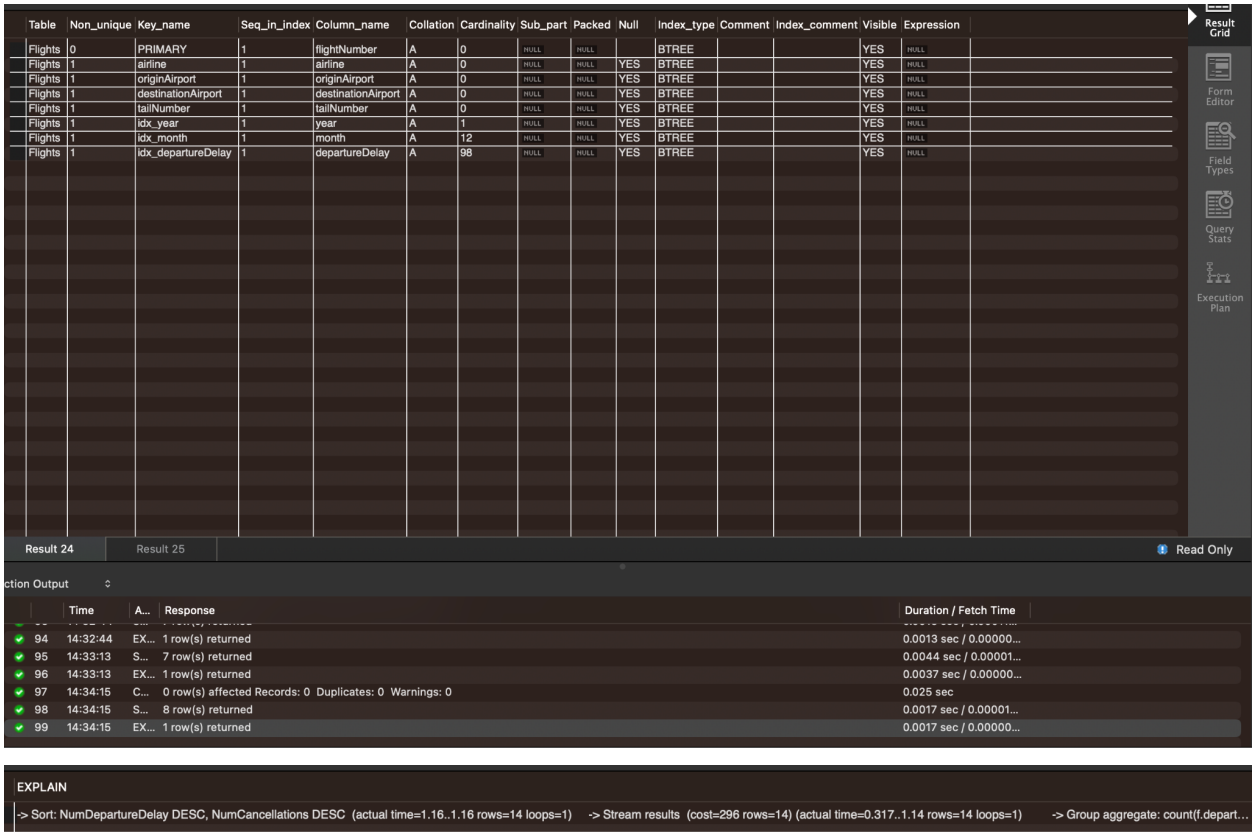


Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
Flights	0	PRIMARY	1	flightNumber	A	0	NULL	NULL		BTREE			YES	NULL
Flights	1	airline	1	airline	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	originAirport	1	originAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	destinationAirport	1	destinationAirport	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	tailNumber	1	tailNumber	A	0	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_year	1	year	A	1	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_month	1	month	A	12	NULL	NULL	YES	BTREE			YES	NULL
Flights	1	idx_departureDelay	1	departureDelay	A	98	NULL	NULL	YES	BTREE			YES	NULL

Result Grid
Form Editor
Field Types
Query Stats
Execution Plan
Result 24
Result 25
Read Only
ction Output

	Time	A...	Response	Duration / Fetch Time
94	14:32:44	EX...	1 row(s) returned	0.0013 sec / 0.00000...
95	14:33:13	S...	7 row(s) returned	0.0044 sec / 0.00001...
96	14:33:13	EX...	1 row(s) returned	0.0037 sec / 0.00000...
97	14:34:15	C...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.025 sec
98	14:34:15	S...	8 row(s) returned	0.0017 sec / 0.00001...
99	14:34:15	EX...	1 row(s) returned	0.0017 sec / 0.00000...

EXPLAIN

-> Sort: NumDepartureDelay DESC, NumCancellations DESC (actual time=1.16..1.16 rows=14 loops=1) -> Stream results (cost=296 rows=14) (actual time=0.317..1.14 rows=14 loops=1) -> Group aggregate: count(f.depart...

'-> Sort: NumDepartureDelay DESC, NumCancellations DESC (actual time=1.76..1.76 rows=14 loops=1)\n

-> Stream results (cost=296 rows=14) (actual time=0.809..1.75 rows=14 loops=1)\n

-> Group aggregate: count(f.departureDelay), count(f.cancelled) (cost=296 rows=14) (actual time=0.808..1.75 rows=14 loops=1)\n

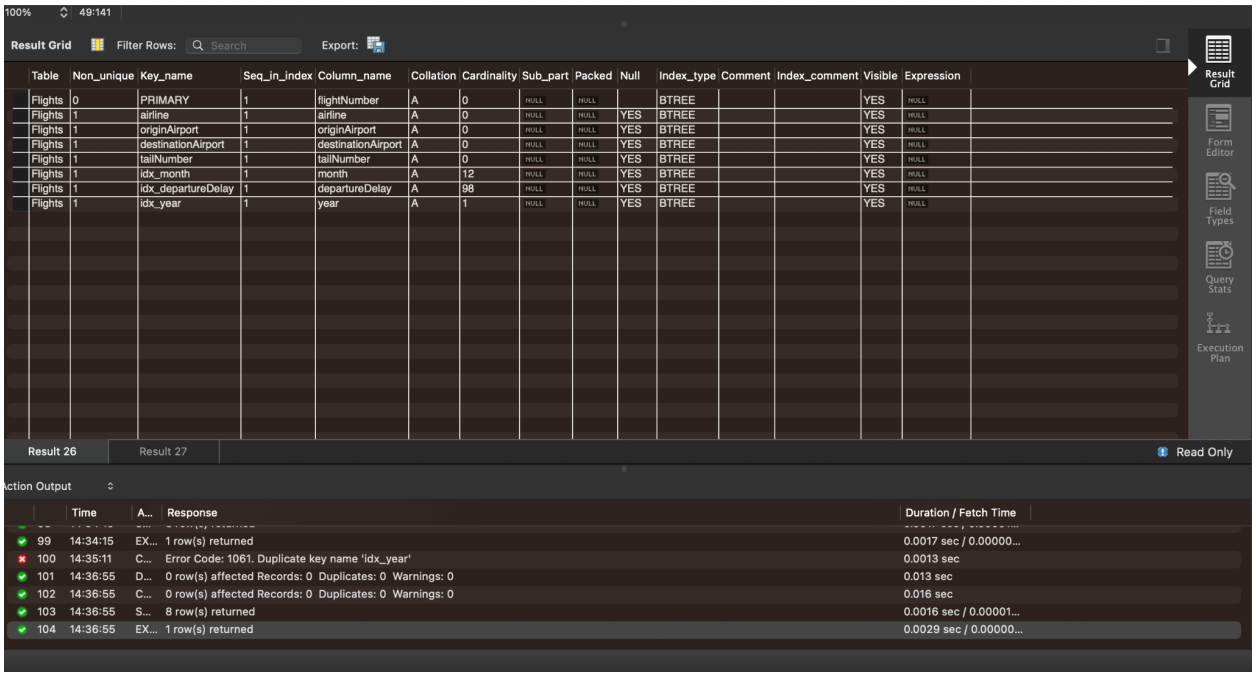
-> Nested loop inner join (cost=186 rows=1104) (actual time=0.652..1.66 rows=1104 loops=1)\n

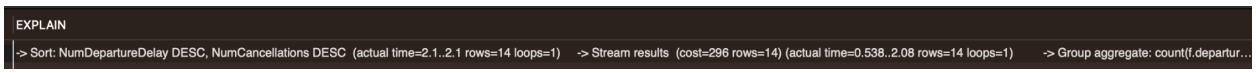
-> Covering index scan on A using PRIMARY (cost=1.65 rows=14) (actual time=0.031..0.0332 rows=14 loops=1)\n

-> Filter: (f.`year` = 2015) (cost=5.81 rows=78.9) (actual time=0.0849..0.11 rows=78.9 loops=14)\n

-> Index lookup on F using airline (airline=a.IATAcodeAirline) (cost=5.81 rows=78.9) (actual time=0.0847..0.105 rows=78.9 loops=14)\n'

DROP INDEX idx_year ON Flights;
CREATE INDEX idx_year ON Flights(year);
SHOW INDEX FROM Flights;
EXPLAIN ANALYZE
SELECT A.IATAcodeAirline, COUNT(F.departureDelay) AS NumDepartureDelay,
COUNT(F.cancelled) AS NumCancellations
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015'
GROUP BY A.IATAcodeAirline
ORDER BY NumDepartureDelay DESC, NumCancellations DESC;





'-> Sort: NumDepartureDelay DESC, NumCancellations DESC (actual time=1.15..1.15 rows=14 loops=1)\n

-> Stream results (cost=296 rows=14) (actual time=0.317..1.14 rows=14 loops=1)\n

-> Group aggregate: count(f.departureDelay), count(f.cancelled) (cost=296 rows=14) (actual time=0.316..1.14 rows=14 loops=1)\n

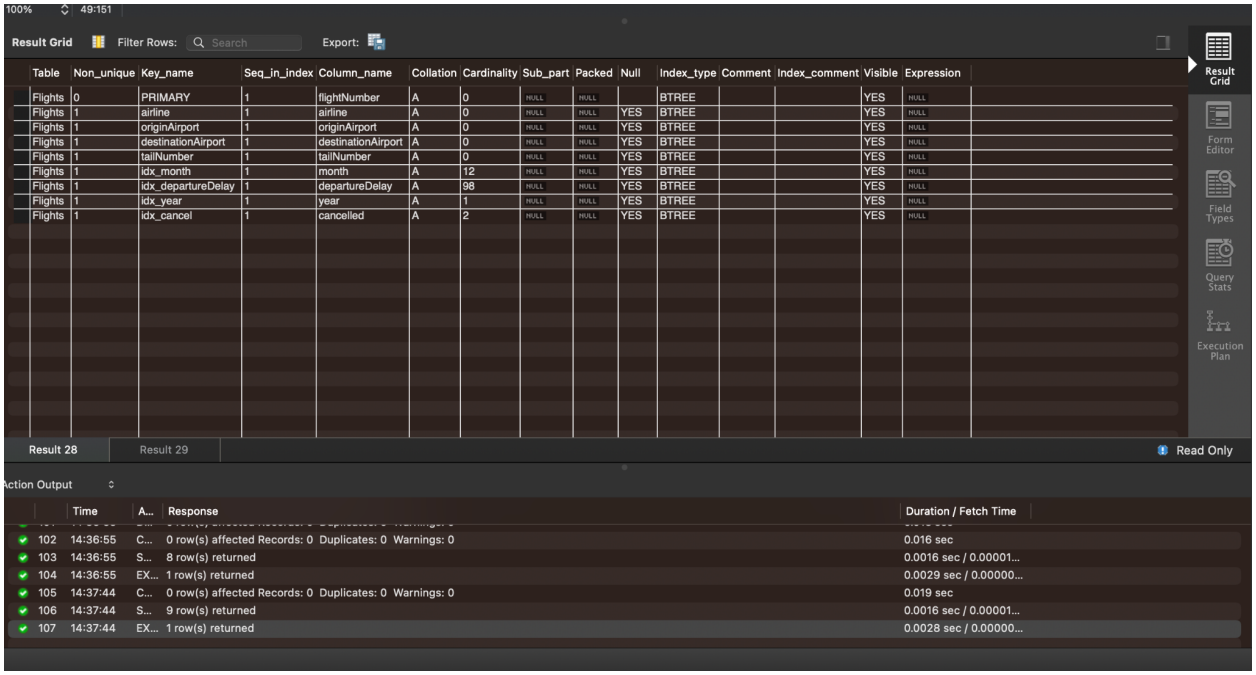
-> Nested loop inner join (cost=186 rows=1104) (actual time=0.167..1.06 rows=1104 loops=1)\n

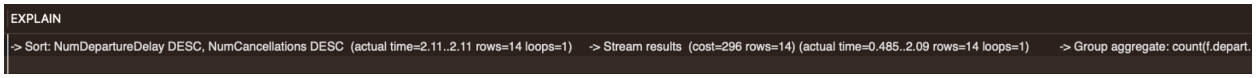
-> Covering index scan on A using PRIMARY (cost=1.65 rows=14) (actual time=0.0267..0.0286 rows=14 loops=1)\n

-> Filter: (f.`year` = 2015) (cost=5.81 rows=78.9) (actual time=0.0463..0.0688 rows=78.9 loops=14)\n

-> Index lookup on F using airline (airline=a.IATAcodeAirline) (cost=5.81 rows=78.9) (actual time=0.0461..0.0637 rows=78.9 loops=14)\n'

DROP INDEX idx_cancel ON Flights;
CREATE INDEX idx_cancel ON Flights(cancelled);
SHOW INDEX FROM Flights;
EXPLAIN ANALYZE
SELECT A.IATAcodeAirline, COUNT(F.departureDelay) AS NumDepartureDelay,
COUNT(F.cancelled) AS NumCancellations
FROM Airlines A
JOIN Flights F ON A.IATAcodeAirline = F.airline
WHERE F.year = '2015'
GROUP BY A.IATAcodeAirline
ORDER BY NumDepartureDelay DESC, NumCancellations DESC;





'-> Sort: NumDepartureDelay DESC, NumCancellations DESC (actual time=0.984..0.984 rows=14 loops=1)\n

-> Stream results (cost=296 rows=14) (actual time=0.332..0.972 rows=14 loops=1)\n

-> Group aggregate: count(f.departureDelay), count(f.cancelled) (cost=296 rows=14) (actual time=0.331..0.969 rows=14 loops=1)\n

-> Nested loop inner join (cost=186 rows=1104) (actual time=0.217..0.91 rows=1104 loops=1)\n

-> Covering index scan on A using PRIMARY (cost=1.65 rows=14) (actual time=0.0151..0.0172 rows=14 loops=1)\n

-> Filter: (f.`year` = 2015) (cost=5.81 rows=78.9) (actual time=0.0431..0.0605 rows=78.9 loops=14)\n

-> Index lookup on F using airline (airline=a.IATAcodeAirline) (cost=5.81 rows=78.9) (actual time=0.0429..0.0568 rows=78.9 loops=14)\n'

Explanation:
For this advanced query, indexing by **idx_departureDelay** is the most efficient indexing option, as by going by overall time taken, it reduces the overall time taken by the largest margin. The other two attempted index options reduce an overall run time of 0.0037 to 0.0028 or 0.0029, but **idx_departureDelay** reduces the time taken to 0.0017, far and away the greatest performance improvement of the three index attempts.