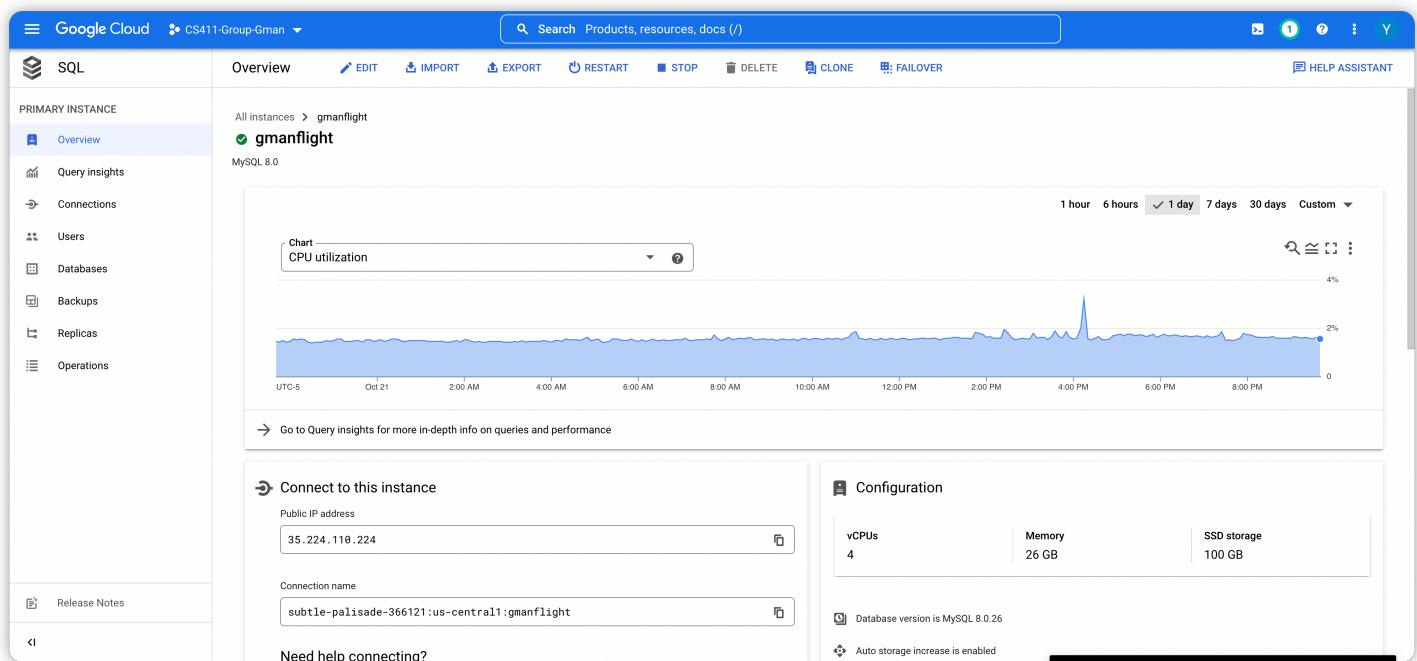


# Database Design

## Database Implementation

*Implementation on GCP*



Google Cloud CS411-Group-Gman

SQL Overview EDIT IMPORT EXPORT RESTART STOP DELETE CLONE FAILOVER HELP ASSISTANT

### PRIMARY INSTANCE

- Overview**
- Query insights
- Connections
- Users
- Databases
- Backups
- Replicas
- Operations
- Release Notes

**Connect to this instance**

Public IP address: 35.224.118.224

Connection name: subtle-palisade-366121:us-central1:gmanflight

**Need help connecting?**

Review the documentation to learn about the many ways to connect to your instance. [Learn more](#)

To connect using gcloud, [OPEN CLOUD SHELL](#)

To learn about connecting with a Compute Engine VM, [START TUTORIAL](#)

**Service account**

p1074096548208-98hr5e@gcp-sa-cloud-sql.iam.gserviceaccount.com

**Operations and logs**

Creation Time	Completion Time	Type	Status
Oct 21, 2022, 7:22:41 PM	Oct 21, 2022, 7:22:55 PM	Import	Import from gs://gmanflight/routes.csv succeeded.
Oct 21, 2022, 7:11:38 PM	Oct 21, 2022, 7:12:09 PM	Update	Update finished
Oct 21, 2022, 3:37:32 PM	Oct 21, 2022, 3:37:46 PM	Import	Import from gs://gmanflight/airlines.csv succeeded.
Oct 21, 2022, 3:33:11 PM	Oct 21, 2022, 3:33:26 PM	Import	Import from gs://gmanflight/airports.csv succeeded.
Oct 21, 2022, 3:26:40 PM	Oct 21, 2022, 3:26:54 PM	Import	Import from gs://gmanflight/airports.csv succeeded.

[View all operations](#)

[View MySQL error logs](#)

**Configuration**

vCPUs 4	Memory 26 GB	SSD storage 100 GB
------------	-----------------	-----------------------

**Instance details**

- Database version is MySQL 8.0.26
- Auto storage increase is enabled
- Automated backups are enabled
- Point-in-time recovery is enabled
- Instance deletion protection is enabled
- Located in us-central1-b
- Highly available (regional)
- No database flags set
- No labels set

[Edit configuration](#)

**Maintenance**

Maintenance window: Updates may occur any day of the week.

[Edit maintenance preferences](#)

Google Cloud CS411-Group-Gman

SQL Overview EDIT IMPORT EXPORT RESTART STOP DELETE CLONE FAILOVER HELP ASSISTANT

### PRIMARY INSTANCE

- Overview**
- Query insights
- Connections
- Users
- Databases
- Backups
- Replicas
- Operations
- Release Notes

To learn about connecting with a Compute Engine VM, [START TUTORIAL](#)

**Service account**

p1074096548208-98hr5e@gcp-sa-cloud-sql.iam.gserviceaccount.com

**Operations and logs**

Creation Time	Completion Time	Type	Status
Oct 21, 2022, 7:22:41 PM	Oct 21, 2022, 7:22:55 PM	Import	Import from gs://gmanflight/routes.csv succeeded.
Oct 21, 2022, 7:11:38 PM	Oct 21, 2022, 7:12:09 PM	Update	Update finished
Oct 21, 2022, 3:37:32 PM	Oct 21, 2022, 3:37:46 PM	Import	Import from gs://gmanflight/airlines.csv succeeded.
Oct 21, 2022, 3:33:11 PM	Oct 21, 2022, 3:33:26 PM	Import	Import from gs://gmanflight/airports.csv succeeded.
Oct 21, 2022, 3:26:40 PM	Oct 21, 2022, 3:26:54 PM	Import	Import from gs://gmanflight/airports.csv succeeded.

[View all operations](#)

[View MySQL error logs](#)

**Configuration**

vCPUs 4	Memory 26 GB	SSD storage 100 GB
------------	-----------------	-----------------------

**Instance details**

- Located in us-central1-b
- Highly available (regional)
- No database flags set
- No labels set

[Edit configuration](#)

**Maintenance**

Maintenance window: Updates may occur any day of the week.

Order of update: Later

Notifications: Off

Upcoming: No maintenance scheduled right now.

[Edit maintenance preferences](#)

[Edit notification preferences](#)

CLOUD SHELL Terminal (subtle-palisade-366121) +

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to subtle-palisade-366121.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
yumingxuanguo@cloudshell:~ (subtle-palisade-366121)$ gcloud sql connect gmanflight --user=root
Allowlisting your IP for incoming connection for 5 minutes...done.
Connected to MySQL server at 35.224.118.224, port 3306.
Welcome to the MySQL monitor. Commands end with ; or \q.
Your MySQL connection id is 23618
Server version: 8.0.26-google (Google)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help,' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

```
mysql> show databases;
+-----+
| Database      |
+-----+
| Gman_Open_Flight |
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
5 rows in set (0.00 sec)

mysql> use Gman_Open_Flight
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

```
mysql> show tables;
+-----+
| Tables_in_Gman_Open_Flight |
+-----+
| Airline          |
| Airport          |
| City             |
| Country          |
| Route            |
+-----+
5 rows in set (0.01 sec)
```

## *Data Definition Language (DDL) Commands*

### 1. Airline.

```
1 | CREATE TABLE Airline(
2 |   Airline_ID      INT NOT NULL,
3 |   Name            VARCHAR(50),
4 |   IATA            CHAR(2),
5 |   ICAO            CHAR(3),
6 |   Callsign:       VARCHAR(50),
7 |   Country:        VARCHAR(50),
8 |   Active:         CHAR(1),
9 |
10| PRIMARY KEY    (Airline_ID)
11| );
```

```
mysql> select count(*) from Airline;
+-----+
| count(*) |
+-----+
|      5887 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> describe Airline;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Airline_ID | int | NO | PRI | NULL | |
| Name | varchar(50) | YES | | NULL | |
| IATA | char(2) | YES | | NULL | |
| ICAO | char(3) | YES | | NULL | |
| Callsign | varchar(50) | YES | | NULL | |
| Country | varchar(50) | YES | MUL | NULL | |
| Active | char(1) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)
```

## 2. Airport.

```
1 CREATE TABLE Airport(
2     Airport_ID    INT NOT NULL,
3     Name          VARCHAR(50),
4     IATA          CHAR(3),
5     ICAO          CHAR(4),
6     Latitude      DOUBLE,
7     Longitude     DOUBLE,
8     Altitude      DOUBLE,
9     City          INT,
10
11    PRIMARY KEY  (Airport_ID),
12    FOREIGN KEY   (City) REFERENCES City(City_ID)
13                      ON DELETE CASCADE
14 );
```

```
mysql> select count(*) from Airport;
+-----+
| count(*) |
+-----+
|      2149 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> describe Airport;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Airport_ID | int        | NO   | PRI | NULL    |       |
| Name        | varchar(50) | YES  |     | NULL    |       |
| IATA         | char(3)    | YES  |     | NULL    |       |
| ICAO         | char(4)    | YES  |     | NULL    |       |
| Latitude     | double     | YES  |     | NULL    |       |
| Longitude    | double     | YES  |     | NULL    |       |
| Altitude     | double     | YES  |     | NULL    |       |
| City         | int        | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

3. City.

```
1 CREATE TABLE City(
2     City_ID      INT NOT NULL AUTO_INCREMENT,
3     Name         VARCHAR(50),
4     Country      CHAR(2),
5
6     PRIMARY KEY  (Name),
7     FOREIGN KEY  (Country) REFERENCES Country(iso_code)
8             ON DELETE CASCADE
9 );
```

```
mysql> select count(*) from City;
+-----+
| count(*) |
+-----+
|      21413 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> describe City;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key  | Default | Extra       |
+-----+-----+-----+-----+-----+-----+
| City_ID | int       | NO   | PRI  | NULL    | auto_increment |
| Name    | varchar(50) | YES  |      | NULL    |              |
| Country | char(2)    | YES  | MUL  | NULL    |              |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

4. Country.

```
1 CREATE TABLE Country(
2     iso_code      CHAR(2) NOT NULL,
3     Name          VARCHAR(50),
4
5     PRIMARY KEY   (Name)
6 );
```

```
mysql> select count(*) from Country;
+-----+
| count(*) |
+-----+
|      240 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> describe Country;
+-----+-----+-----+-----+-----+-----+
| Field    | Type     | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| iso_code | char(2) | NO   | PRI  | NULL    |       |
| Name     | varchar(50)| YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

5. Route.

```
1 | CREATE TABLE Route(
2 |   Route_ID          INT,
3 |   Airline_ID        INT,
4 |   Source_Airport_ID INT,
5 |   Destination_Airport_ID INT,
6 |   Stops             INT,
7 |
8 |   PRIMARY KEY       (Route_ID, Source_Airport_ID, Destination_Airport_ID)
9 |   FOREIGN KEY       (Airline_ID) REFERENCES Airline(Airline_ID),
10 |  FOREIGN KEY       (Source_Airport_ID) REFERENCES Airport(Airport_ID),
11 |  FOREIGN KEY       (Destination_Airport_ID) REFERENCES Airport(Airport_ID)
12 |  ON DELETE CASCADE
13 | )
```

```
mysql> select count(*) from Route;
+-----+
| count(*) |
+-----+
|      18454 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> describe Route;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key  | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| Route_ID       | int    | NO   | PRI  | NULL    | auto_increment |
| Airline_ID     | int    | NO   | MUL  | NULL    |                 |
| Source_Airport_ID | int    | NO   | PRI  | NULL    |                 |
| Destination_Airport_ID | int    | NO   | PRI  | NULL    |                 |
| Stops          | int    | YES  |      | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

# Advanced SQL Queries

1. Select all the routes' destination airport names whose Source\_ID is an element of the set of routes' Source\_IDs whose Airline\_ID is 4611.

```
1 | select distinct a.Name
2 | from Airport a join Route r on a.Airport_ID = r.Destination_Airport_ID
3 | where r.Source_Airport_ID in (select Source_Airport_ID from Route where Airline_ID =
4611);
```

```
mysql> select distinct a.Name
    -> from Airport a join Route r on a.Airport_ID = r.Destination_Airport_ID
    -> where r.Source_Airport_ID in (select Source_Airport_ID from Route where Airline_ID = 4611);
+-----+
| Name
+-----+
| Hangzhou Xiaoshan International Airport
| Kunming Changshui International Airport
| Ürümqi Diwopu International Airport
| Taichung Ching Chuang Kang Airport
| Kaohsiung International Airport
| Haikou Meilan International Airport
| Longdongbao Airport
| Lanzhou Zhongchuan Airport
| Macau International Airport
| Mianyang Airport
| Nanning Wuxu Airport
| Hong Kong International Airport
| Ordos Ejin Horo Airport
| Fuzhou Changle International Airport
| Yulin Yuyang Airport
| Wenzhou Longwan International Airport
| Yiwu Airport
| Yangzhou Taizhou Airport
| Nanchang Changbei International Airport
| Jieyang Chaoshan International Airport
| Nanjing Lukou Airport
| Quanzhou Jinjiang International Airport
| Taiyuan Wusu Airport
| Jingdezhen Airport
| Sunan Shuofang International Airport
| Zhengzhou Xinzheng International Airport
| Changzhou Benniu Airport
| Kuala Lumpur International Airport
| Kansai International Airport
| Anqing Tianzhushan Airport
| Zhanjiang Airport
| Zunyi Xinzhou Airport
```

2. Select for each City whose name starts with an "L" the count of Routes that use them as the Source\_ID.

```
1 | select c.Name, count(r.Source_Airport_ID) as num
2 | from City c join Route r on c.City_ID = r.Source_Airport_ID
3 | where c.Name like "L%"
4 | group by c.Name
5 | order by num desc;
```

```
mysql> select c.Name, count(r.Source_Airport_ID) as num
-> from City c join Route r on c.City_ID = r.Source_Airport_ID
-> where c.Name like "L%"
-> group by c.Name
-> order by num desc;
+-----+-----+
| Name          | num |
+-----+-----+
| Licheng       | 324 |
| Linz          | 306 |
| Liestal       | 143 |
| Le Grand-Quevilly | 140 |
| Loures        | 118 |
| Liévin        | 92  |
| León          | 62  |
| Les Pavillons-sous-Bois | 51  |
| Leping         | 49  |
| Luena          | 45  |
| Leeds          | 39  |
| Lin'an         | 37  |
| Loudi          | 32  |
| Lviv           | 32  |
| Loznica        | 28  |
| Laaoouama     | 27  |
| La Celle-Saint-Cloud | 16  |
| Lhokseumawe   | 14  |
| Las Cruces    | 13  |
| Louisville    | 11  |
| Liberia        | 10  |
```

## Indexing Analysis

# 1.

```
mysql> explain analyze
-- select distinct a.Name
-- from Airport a join Route r on a.Airport_ID = r.Destination_Airport_ID
-- where r.Source_Airport_ID in (select Source_Airport_ID from Route where Airline_ID = 4611);
+-----+
| EXPLAIN
+-----+
| > Table scan on <temporary> (cost=0.01..18.02 rows=1243) (actual time=0.000..0.007 rows=105 loops=1)
|   -> Temporary table with deduplication (cost=717.45..735.46 rows=1243) (actual time=1.935..1.939 rows=105 loops=1)
|     -> Nested loop inner join (cost=593.15 rows=1243) (actual time=0.079..1.414 rows=920 loops=1)
|       -> Nested loop inner join (cost=158.16 rows=1243) (actual time=0.070..0.465 rows=920 loops=1)
|         -> Table scan on <subquery> (cost=0.05..3.33 rows=66) (actual time=0.001..0.003 rows=22 loops=1)
|           -> Materialize with deduplication (cost=13.53..16.81 rows=66) (actual time=0.051..0.054 rows=22 loops=1)
|             -> Index lookup on Route using Airline_ID (Airline_ID=4611) (cost=6.88 rows=66) (actual time=0.027..0.035 rows=66 loops=1)
|             -> Index lookup on r using Source_Airport_ID (Source_Airport_ID=<subquery> .Source_Airport_ID) (cost=18.96 rows=19) (actual time=0.005..0.016 rows=42 loops=22)
|           -> Single-row index lookup on a using PRIMARY (Airport_ID=r.Destination_Airport_ID) (cost=16.51 rows=1) (actual time=0.001..0.001 rows=1 loops=920)
|
+-----+
1 row in set (0.01 sec)
```

```
mysql> create index A on Airport(Airport_ID);
Query OK, 0 rows affected, 1 warning (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql> explain analyze
-- select distinct a.Name
-- from Airport a join Route r on a.Airport_ID = r.Destination_Airport_ID
-- where r.Source_Airport_ID in (select Source_Airport_ID from Route where Airline_ID = 4611);
+-----+
| EXPLAIN
+-----+
| > Table scan on <temporary> (cost=0.01..18.02 rows=1243) (actual time=0.000..0.007 rows=105 loops=1)
|   -> Temporary table with deduplication (cost=717.45..735.46 rows=1243) (actual time=1.961..1.979 rows=105 loops=1)
|     -> Nested loop inner join (cost=593.15 rows=1243) (actual time=0.076..1.460 rows=920 loops=1)
|       -> Nested loop inner join (cost=158.16 rows=1243) (actual time=0.058..0.505 rows=920 loops=1)
|         -> Table scan on <subquery> (cost=0.05..3.33 rows=66) (actual time=0.001..0.002 rows=22 loops=1)
|           -> Materialize with deduplication (cost=13.53..16.81 rows=66) (actual time=0.043..0.046 rows=22 loops=1)
|             -> Index lookup on Route using Airline_ID (Airline_ID=4611) (cost=6.88 rows=66) (actual time=0.019..0.027 rows=66 loops=1)
|             -> Index lookup on r using Source_Airport_ID (Source_Airport_ID=<subquery> .Source_Airport_ID) (cost=18.96 rows=19) (actual time=0.008..0.018 rows=42 loops=22)
|           -> Single-row index lookup on a using PRIMARY (Airport_ID=r.Destination_Airport_ID) (cost=16.51 rows=1) (actual time=0.001..0.001 rows=1 loops=920)
|
+-----+
1 row in set (0.01 sec)
```

```
mysql> create index B on Route(Destination_Airport_ID);
Query OK, 0 rows affected, 1 warning (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 1

mysql> explain analyze
-- select distinct a.Name
-- from Airport a join Route r on a.Airport_ID = r.Destination_Airport_ID
-- where r.Source_Airport_ID in (select Source_Airport_ID from Route where Airline_ID = 4611);
+-----+
| EXPLAIN
+-----+
| > Table scan on <temporary> (cost=0.01..18.02 rows=1243) (actual time=0.000..0.007 rows=105 loops=1)
|   -> Temporary table with deduplication (cost=717.45..735.46 rows=1243) (actual time=1.922..1.935 rows=105 loops=1)
|     -> Nested loop inner join (cost=593.15 rows=1243) (actual time=0.074..1.419 rows=920 loops=1)
|       -> Nested loop inner join (cost=158.16 rows=1243) (actual time=0.066..0.460 rows=920 loops=1)
|         -> Table scan on <subquery> (cost=0.05..3.33 rows=66) (actual time=0.001..0.003 rows=22 loops=1)
|           -> Materialize with deduplication (cost=13.53..16.81 rows=66) (actual time=0.052..0.056 rows=22 loops=1)
|             -> Index lookup on Route using Airline_ID (Airline_ID=4611) (cost=6.88 rows=66) (actual time=0.028..0.037 rows=66 loops=1)
|             -> Index lookup on r using Source_Airport_ID (Source_Airport_ID=<subquery> .Source_Airport_ID) (cost=18.96 rows=19) (actual time=0.005..0.016 rows=42 loops=22)
|           -> Single-row index lookup on a using PRIMARY (Airport_ID=r.Destination_Airport_ID) (cost=16.51 rows=1) (actual time=0.001..0.001 rows=1 loops=920)
|
+-----+
1 row in set (0.00 sec)
```

```

mysql> create index C on Airport(Name);
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain analyze
--> select distinct a.Name
--> from Airport a join Route r on a.Airport_ID = r.Destination_Airport_ID
--> where r.Source_Airport_ID in (select Source_Airport_ID from Route where Airline_ID = 4611);
+-----+
| EXPLAIN
+-----+
|-----+
| > Table scan on <temporary> (cost=0.01..18.02 rows=1243) (actual time=0.001..0.007 rows=105 loops=1)
|   -> Temporary table with deduplication (cost=717.45..735.46 rows=1243) (actual time=1.932..1.944 rows=105 loops=1)
|     -> Nested loop inner join (cost=593.15 rows=1243) (actual time=0.059..1.426 rows=920 loops=1)
|       -> Nested loop inner join (cost=158.16 rows=1243) (actual time=0.052..0.436 rows=920 loops=1)
|         -> Table scan on <subquery> (cost=0.05..3.33 rows=66) (actual time=0.001..0.002 rows=22 loops=1)
|           -> Materialize with deDuplication (cost=0.05..1.01 rows=66) (actual time=0.042..0.045 rows=22 loops=1)
|             -> Index lookup on Route for Airline_ID = 4611 (Source_Airport_ID=<subquery>.Source_Airport_ID) (cost=18.96 rows=19) (actual time=0.018..0.026 rows=66 loops=1)
|             -> Index lookup on r using Source_Airport_ID (Source_Airport_ID=r.Destination_Airport_ID) (cost=16.51 rows=1) (actual time=0.001..0.001 rows=1 loops=920)
|       -> Single-row index lookup on a using PRIMARY (Airport_ID=a.Destination_Airport_ID) (cost=16.51 rows=1) (actual time=0.001..0.001 rows=1 loops=920)
|-----+
+-----+
1 row in set (0.01 sec)

```

In our first advanced query, we created three different indices on the table. However, none of them gives us significant improvement. Our guessing is that, for the first two indices, which are placed on the primary keys, SQL already made some default and optimal indices for them, so our indices might not work as expected; for the third index, there probably is not much work to be done when selecting distinct names, so our index might not improve much.

## 2.

```

mysql> explain analyze
--> select c.Name, count(r.Source_Airport_ID) as num
--> from City c join Route r on c.City_ID = r.Source_Airport_ID
--> where c.Name like "L%"
--> group by c.Name
--> order by num desc;
+-----+
| EXPLAIN
+-----+
|-----+
| > Sort: num DESC (actual time=11.951..11.954 rows=55 loops=1)
|   -> Table scan on <temporary> (actual time=0.002..0.010 rows=55 loops=1)
|     -> Aggregate using temporary table (actual time=11.908..11.919 rows=55 loops=1)
|       -> Nested loop inner join (cost=7167.13 rows=44218) (actual time=0.088..11.025 rows=1677 loops=1)
|         -> Filter: (c.Name like 'L%') (cost=2137.85 rows=2348) (actual time=0.044..8.060 rows=1149 loops=1)
|           -> Table scan on c (cost=2137.85 rows=21136) (actual time=0.036..5.950 rows=21413 loops=1)
|             -> Index lookup on r using Source_Airport_ID (Source_Airport_ID=r.City_ID) (cost=0.26 rows=1) (actual time=0.002..0.002 rows=1 loops=1149)
|-----+
+-----+
1 row in set (0.01 sec)

```

```

mysql> create index A on City(Name);
Query OK, 0 rows affected (0.21 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain analyze
--> select c.Name, count(r.Source_Airport_ID) as num
--> from City c join Route r on c.City_ID = r.Source_Airport_ID
--> where c.Name like "L%"
--> group by c.Name
--> order by num desc;
+-----+
| EXPLAIN
+-----+
|-----+
| > Sort: num DESC (actual time=3.536..3.540 rows=55 loops=1)
|   -> Stream results (cost=4876.37 rows=21636) (actual time=0.160..3.501 rows=55 loops=1)
|     -> Group aggregate: count(r.Source_Airport_ID) (cost=4876.37 rows=21636) (actual time=0.156..3.480 rows=55 loops=1)
|       -> Nested loop inner join (cost=2712.73 rows=21636) (actual time=0.134..3.241 rows=1677 loops=1)
|         -> Filter: (c.Name like 'L%') (cost=2511.66 rows=1149) (actual time=0.1084..3.1642 rows=1149 loops=1)
|           -> Index range scan on c for Name like 'L%' (cost=2511.66 rows=1149) (actual time=0.088..0.456 rows=1149 loops=1)
|             -> Index lookup on r using Source_Airport_ID (Source_Airport_ID=c.City_ID) (cost=0.26 rows=19) (actual time=0.002..0.002 rows=1 loops=1149)
|-----+
+-----+
1 row in set (0.00 sec)

```

Notice that in our first index, which is placed on `City(Name)`, greatly improved the performance in joining the two tables. This improvement stems from the reduction of rows checking due to the index range scan.

```
mysql> create index B on City(City_ID);
Query OK, 0 rows affected (0.12 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain analyze
--> select c.Name, count(r.Source_Airport_ID) as num
--> from City c join Route r on c.City_ID = r.Source_Airport_ID
--> where c.Name like "L%"
--> group by c.Name
--> order by num desc;
+
+-----+
| EXPLAIN
+-----+
|
+-----+
| -> Sort: num DESC (actual time=11.812..11.815 rows=55 loops=1)
|   -> Table scan on <temporary> (actual time=0.002..0.011 rows=55 loops=1)
|     -> Aggregate using temporary table (actual time=11.765..11.778 rows=55 loops=1)
|       -> Nested loop inner join (cost=167.13 rows=55 loops=1) (actual time=0.001..11.852 rows=16777 loops=1)
|         -> Filter: (c.Name like "L%")
|           -> Index scan on c (cost=2137.85 rows=21136 loops=1) (actual time=0.046..8.007 rows=1149 loops=1)
|             -> Index lookup on r using Source_Airport_ID (Source_Airport_ID=c.City_ID) (cost=0.26 rows=19) (actual time=0.002..0.002 rows=1 loops=1149)
|
+-----+
1 row in set (0.01 sec)
```

Our second index does not give us significant improvement. Reasons are similar to our first query.

```
mysql> create index C on Route(Source_Airport_ID);
Query OK, 0 rows affected (0.18 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain analyze
--> select c.Name, count(r.Source_Airport_ID) as num
--> from City c join Route r on c.City_ID = r.Source_Airport_ID
--> where c.Name like "L%"
--> group by c.Name
--> order by num desc;
+
+-----+
| EXPLAIN
+-----+
|
+-----+
| -> Sort: num DESC (actual time=18.975..18.979 rows=55 loops=1)
|   -> Table scan on <temporary> (actual time=0.001..0.007 rows=55 loops=1)
|     -> Aggregate using temporary table (actual time=18.929..18.938 rows=55 loops=1)
|       -> Nested loop inner join (cost=8328.55 rows=2050) (actual time=0.096..17.971 rows=1677 loops=1)
|         -> Index scan on r using Airline_ID (cost=1869.65 rows=18454) (actual time=0.073..4.795 rows=18454 loops=1)
|           -> Filter: (c.Name like "L%") (cost=0.25 rows=0) (actual time=0.001..0.001 rows=0 loops=18454)
|             -> Single-row index lookup on c using PRIMARY (City_ID=r.Source_Airport_ID) (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=18454)
|
+-----+
1 row in set (0.02 sec)
```

Our third index exchanged the operation priority when joining the two tables, which seems to decrease the number of checked rows. However, while we have this decreament, the actual time increased. We think this is because checking the `Airline_ID`, which is of type `INT`, might be more time consuming.