

Stage 3

GCP Screenshots of Connection and Table Descriptions

<https://colab.research.google.com/drive/1BDqivQL3J5XUwYTB5XRn6F96They7VRc?usp=sharing> - Mock Data script

```
mysql> SELECT COUNT(*) FROM Apartment;
+-----+
| COUNT(*) |
+-----+
|      1000 |
+-----+
1 row in set (0.12 sec)

mysql> SELECT COUNT(*) FROM Comment;
+-----+
| COUNT(*) |
+-----+
|      1000 |
+-----+
1 row in set (0.05 sec)

mysql> SELECT COUNT(*) FROM Management;
+-----+
| COUNT(*) |
+-----+
|        20 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT COUNT(*) FROM Review;
+-----+
| COUNT(*) |
+-----+
|      1000 |
+-----+
1 row in set (0.08 sec)

mysql> SELECT COUNT(*) FROM Resident;
+-----+
| COUNT(*) |
+-----+
|      1000 |
+-----+
1 row in set (0.06 sec)

mysql>
```

```
mysql> describe Review;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| ReviewId | int | NO | PRI | NULL | |
| ApartmentId | int | NO | MUL | NULL | |
| OverallAvgScore | double | YES | | 0 | |
| NoiseAvgScore | double | YES | | 0 | |
| LocationAvgScore | double | YES | | 0 | |
| WifiAvgScore | double | YES | | 0 | |
+-----+
6 rows in set (0.00 sec)

mysql>
```

```
mysql> describe Resident;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| ResidentId | int | NO | PRI | NULL | |
| ApartmentId | int | NO | MUL | NULL | |
| FirstName | varchar(30) | NO | | NULL | |
| LastName | varchar(30) | NO | | NULL | |
| Email | varchar(50) | NO | | NULL | |
| OAuthId | int | NO | | NULL | |
| ProfilePicture | varchar(150) | YES | | NULL | |
| SchoolGrade | enum('F','So','J','S') | NO | | NULL | |
+-----+
8 rows in set (0.00 sec)

mysql>
```

```
mysql> describe Management;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| Name | varchar(30) | NO | PRI | NULL | |
| Owner | varchar(100) | NO | | NULL | |
| HQ | enum('AL','AK','AZ','AR','CA','CO','CT','DE','FL','GA','HI','ID','IL','IN','IA','KS','KY','LA','ME','MD','MA','MI','MN','MS','MO','MT','NE','NV','NH','NJ','NM','NY','NC','ND','OH','OK','OR','PA','RI','SC','SD','TN','TX','UT','VT','VA','WA','WV','WI','WY') | NO | | NULL | |
+-----+
3 rows in set (0.01 sec)

mysql>
```

```
mysql> describe Comment;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| CommentId | int | NO | PRI | NULL | |
| ResidentId | int | NO | MUL | NULL | |
| ApartmentId | int | NO | MUL | NULL | |
| FeedbackText | varchar(255) | YES | | NULL | |
| OverallScore | int | YES | | NULL | |
| NoiseScore | int | YES | | NULL | |
| LocationScore | int | YES | | NULL | |
| WifiScore | int | YES | | NULL | |
+-----+
8 rows in set (0.01 sec)

mysql>
```

Google Cloud Platform CS411 1

SQL Overview EDIT IMPORT EXPORT RESTART STOP DELETE CLONE

Release Notes All instances > rate-my-apartment

CLOUD SHELL Terminal (cs411-1) +

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> show tables;

```
Tables in rate_my_apartment
+-----+
| Apartment |
| Comment   |
| Management |
| Resident  |
| Review     |
+-----+
```

5 rows in set (0.01 sec)

mysql> describe Apartment;

```
Field      Type      Null      Key      Default      Extra
+-----+
| ApartmentId | int      | NO       | PRI     | NULL        |
| ApartmentName | varchar(30) | NO       |         | NULL        |
| ManagementName | varchar(30) | NO       | MUL     | NULL        |
| State       | enum('AL','AK','AZ','AR','CA','CO','CT','DE','FL','GA','HI','ID','IL','IN','IA','KS','KY','LA','ME','MD','MA','MI','MN','MS','MO','MT','NE','NV','NH','NJ','NM','NY','NC','ND','OH','OK','OR','PA','RI','SC','SD','TN','TX','UT','VT','VA','WA','WV','WI','WY') | NO       |         | NULL        |
| City        | varchar(50) | NO       |         | NULL        |
| ZipCode     | int      | NO       |         | NULL        |
| ApartmentAddress | varchar(50) | NO       |         | NULL        |
| Has1Br      | tinyint(1) | YES      |         | NULL        |
| Has2Br      | tinyint(1) | YES      |         | NULL        |
| Has3Br      | tinyint(1) | YES      |         | NULL        |
| MinPrice    | int      | YES      |         | NULL        |
| MaxPrice    | int      | YES      |         | NULL        |
| BusStop     | tinyint(1) | YES      |         | NULL        |
| Gym         | tinyint(1) | YES      |         | NULL        |
| 6MonthLease | tinyint(1) | YES      |         | NULL        |
| 12MonthLease | tinyint(1) | YES      |         | NULL        |
+-----+
```

16 rows in set (0.00 sec)

mysql> \q

<https://console.cloud.google.com/sql/cloudshell=true&project=cs411-1&supportedpurview=project>

console.cloud.google.com/sql/instances/rate-my-apartment/overview?cloudshell=true&project=cs411-1&supportedpurview=project

Google Cloud Platform CS411 1

SQL Overview EDIT IMPORT EXPORT RESTART STOP DELETE CLONE

PRIMARY INSTANCE

Overview

Release Notes

All instances > rate-my-apartment

rate-my-apartment

MySQL 8.0

1 hour 6 hours 1 day 7 days 30 days Custom

Chart

CLOUD SHELL Terminal (cs411-1) +

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to cs411-1.
Use "gcloud config set project [PROJECT ID]" to change to a different project.
sobiarchna@cloudshell: (cs411-1) \$ gcloud sql connect rate-my-apartment --user=root --quit
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user: [root]. Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 19
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>

Uploads and CS411 1 operations

Started rate-my-apartment 11:24:53 PM GMT-5

DDL Script for Tables

```
CREATE TABLE Comment(  
    CommentId INTEGER PRIMARY KEY,  
    ResidentId INTEGER NOT NULL,  
    ApartmentId INTEGER NOT NULL,  
    FeedbackText VARCHAR(255),  
    OverallScore INTEGER,  
    NoiseScore INTEGER,  
    LocationScore INTEGER,  
    WifiScore INTEGER,  
    FOREIGN KEY(ResidentId) REFERENCES Resident(ResidentId) ON DELETE  
CASCADE,  
    FOREIGN KEY(ApartmentId) REFERENCES Apartment(ApartmentId) ON DELETE  
CASCADE  
);  
  
CREATE TABLE Resident(  
    ResidentId INTEGER PRIMARY KEY,  
    ApartmentId INTEGER NOT NULL,  
    FirstName VARCHAR(30) NOT NULL,  
    LastName VARCHAR(30) NOT NULL,  
    Email VARCHAR(50) NOT NULL,  
    OauthId INTEGER NOT NULL,  
    ProfilePicture VARCHAR(150),  
    SchoolGrade ENUM('F', 'So', 'J', 'S') NOT NULL,  
    FOREIGN KEY(ApartmentId) REFERENCES Apartment(ApartmentId) ON DELETE  
CASCADE  
);  
  
CREATE TABLE Apartment(  
    ApartmentId INTEGER PRIMARY KEY NOT NULL,  
    ApartmentName VARCHAR(30) NOT NULL,  
    ManagementName VARCHAR(30) NOT NULL,  
    State ENUM('AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'FL', 'GA',  
'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI',  
'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND',  
'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA',  
'WA', 'WV', 'WI', 'WY') NOT NULL,  
    City VARCHAR(50) NOT NULL,  
    ZipCode INTEGER NOT NULL,
```

```

    ApartmentAddress VARCHAR(50) NOT NULL,
    Has1Br BOOLEAN,
    Has2Br BOOLEAN,
    Has3Br BOOLEAN,
    MinPrice INTEGER,
    MaxPrice INTEGER,
    BusStop BOOLEAN,
    Gym: BOOLEAN,
    6MonthLease: BOOLEAN,
    12MonthLease: BOOLEAN,
    FOREIGN KEY(ManagementName) REFERENCES Management(Name) ON DELETE
CASCADE
);

CREATE TABLE Review(
    ReviewId INTEGER PRIMARY KEY,
    ApartmentId INTEGER NOT NULL,
    OverallAvgScore: REAL DEFAULT 0,
    NoiseAvgScore: REAL DEFAULT 0,
    LocationAvgScore: REAL DEFAULT 0,
    WifiAvgScore: REAL DEFAULT 0,
    FOREIGN KEY(ApartmentId) REFERENCES Apartment(ApartmentId) ON DELETE
CASCADE
);

CREATE Table Management(
    Name VARCHAR(30) PRIMARY KEY,
    Owner VARCHAR(100) NOT NULL,
    HQ: Enum('AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'FL', 'GA',
'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA', 'MI',
'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY', 'NC', 'ND',
'OH', 'OK', 'OR', 'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT', 'VA',
'WA', 'WV', 'WI', 'WY') NOT NULL
);

```

Queries

[Query 1]: Selects all of the managements and scores from managements with an average rating above 1.25

SELECT Management.Name, AVG(OverallAvgScore) AS MgmtAvg

```

FROM Management JOIN Apartment ON Management.Name = Apartment.ManagementName
NATURAL JOIN Review
GROUP BY Management.Name
HAVING MgmtAvg > 1.25;

```

```

mysql> SELECT Management.Name, AVG(OverallAvgScore) AS MgmtAvg
-> FROM Management JOIN Apartment ON Management.Name = Apartment.ManagementName NATURAL JOIN Review
-> GROUP BY Management.Name
-> HAVING MgmtAvg > 1.25;
+-----+-----+
| Name                                     | MgmtAvg |
+-----+-----+
| Cole-Dougherty                         | 1.441580756013746 |
| Pollard-Mendoza                        | 1.606529209621993 |
| Ayala, Stevens and Chapman            | 1.5105263157894737 |
| Singh Inc                             | 1.6432748538011697 |
| Wilson, Gonzalez and Kim              | 1.3823024054982815 |
| Gutierrez LLC                         | 1.6253205128205128 |
| Young-Wagner                          | 1.3860544217687072 |
| Garza-Smith                           | 1.5344202898550723 |
| Thomas, Gutierrez and Alvarez         | 1.8061056105610562 |
| Martinez, Johnson and Campos          | 1.4714285714285713 |
| Fobby's Apartments                    | 2 |
| Zobby's Apartments                    | 2 |
| Cobby's Apartments                    | 1.3333333333333333 |
| Robby's Apartments                    | 1.8333333333333335 |
| Mobby's Apartments                    | 2 |
| Nobby's Apartments                    | 2 |
| Sobby's Apartments                    | 2.5 |
+-----+-----+
17 rows in set (0.01 sec)

```

[Query 1 but with a limit of 15]

```

SELECT Management.Name, AVG(OverallAvgScore) AS MgmtAvg
FROM Management JOIN Apartment ON Management.Name = Apartment.ManagementName
NATURAL JOIN Review
GROUP BY Management.Name
HAVING MgmtAvg > 1.25
LIMIT 15;

```

```
mysql> SELECT Management.Name, AVG(OverallAvgScore) AS MgmtAvg
-> FROM Management JOIN Apartment ON Management.Name = Apartment.ManagementName NATURAL JOIN Review
-> GROUP BY Management.Name
-> HAVING MgmtAvg > 1.25
-> LIMIT 15;
```

Name	MgmtAvg
Ayala, Stevens and Chapman	1.5105263157894737
Cobby's Apartments	1.3333333333333333
Cole-Dougherty	1.4415807560137455
Fobby's Apartments	2
Garza-Smith	1.5344202898550723
Gutierrez LLC	1.6253205128205126
Martinez, Johnson and Campos	1.4714285714285713
Mobby's Apartments	2
Nobby's Apartments	2
Pollard-Mendoza	1.6065292096219932
Robby's Apartments	1.8333333333333335
Singh Inc	1.6432748538011694
Sobby's Apartments	2.5
Thomas, Gutierrez and Alvarez	1.8061056105610562
Wilson, Gonzalez and Kim	1.382302405498282

15 rows in set (0.01 sec)

[Query 2]: Select all apartment comments from AK

```
SELECT CommentId, OverallScore, NoiseScore, LocationScore, WifiScore, FeedbackText,
State
FROM Comment c JOIN Resident r ON r.ResidentId = c.ResidentId JOIN Apartment b ON
b.ApartmentId = r.ApartmentId
WHERE r.ApartmentId IN (SELECT a.ApartmentId
FROM Apartment a
WHERE a.State = 'AK');
```

```
mysql> SELECT CommentId, OverallScore, NoiseScore, LocationScore, WifiScore, FeedbackText, State
-> FROM Comment c JOIN Resident r ON r.ResidentId = c.ResidentId JOIN Apartment b ON b.ApartmentId = r.ApartmentId
-> WHERE r.ApartmentId IN (SELECT a.ApartmentId
-> FROM Apartment a
-> WHERE a.State = 'AK');
```

CommentId	OverallScore	NoiseScore	LocationScore	WifiScore	FeedbackText	State
3999	1	1	3	1	Foreign skin interview all reach education building activity. Interview somebody Mrs many strategy possible camera.	AK
7494	3	4	4	4	Enough page dog sea you. Outside health action total.	AK
8247	2	1	4	1	Good deal than mention argue leave box. Meet last fear plant soon size national board.	AK
9312	3	3	4	1	Stuff trade might yard rock. Big black window add PM teach.	AK
5709	3	2	3	3	Among measure wear maybe. Ever decade stage meeting drop.	AK
5441	4	4	2	1	End view approach lose ok. Move the peace former candidate buy run.	AK
8677	2	4	2	4	Understand personal employee page manager. Response radio she human low movement then college.	AK
9480	2	1	4	3	Participant do interest film. Space first fear itself area.	AK
1309	2	1	3	2	Player relationship responsibility sense. Receive training old among reality magazine.	AK
7913	1	2	3	1	Government former particularly had write detail particularly. Single especially security.	AK
9676	4	3	2	3	Threat health stop position spend white state. Best coach dinner begin your.	AK
9296	2	3	2	1	Alone himself through really car. Movie ability reflect attorney bit good.	AK
3988	4	4	3	1	Herself town manager close. He rock amount we force dream.	AK
388	3	4	4	2	Current environment rock director. Open effort pass from.	AK
9121	2	1	3	3	Almost us next why. Student hour program evidence major.	AK
8801	1	4	3	2	Military as make act. Poor world financial sport home.	AK
925	4	3	1	2	Scene add evidence evidence bit theory nothing natural. Tax manage rise television condition.	AK
4958	4	1	4	3	Instead later ability into indicate reduce. Center as condition second.	AK
7086	4	1	3	2	Success resource argue participant act. Wait down possible very physical let discussion former.	AK
9311	3	3	2	1	Game structure security though whom. Product fill across but next how well make.	AK
3358	1	3	4	2	Off anything even home almost activity join late. High these truth mention campaign.	AK
548	1	3	1	3	Station while technology start nearly. Bed magazine cause.	AK
7249	3	4	3	4	Allow imagine bill month arm. Public help then natural number detail.	AK
7859	2	4	1	1	Assume project war put set. Table store real imagine site dream any popular.	AK

24 rows in set (0.00 sec)

[Query 2 but with a limit of 15]

```
SELECT CommentId, OverallScore, NoiseScore, LocationScore, WifiScore, FeedbackText,
State
FROM Comment c JOIN Resident r ON r.ResidentId = c.ResidentId JOIN Apartment b ON
b.ApartmentId = r.ApartmentId
```

```

WHERE r.ApartmentId IN (SELECT a.ApartmentId
                        FROM Apartment a
                        WHERE a.State = 'AK')

LIMIT 15;

```

```

mysql> SELECT CommentId, OverallScore, NoiseScore, LocationScore, WifiScore, FeedbackText, State
-> FROM Comment c JOIN Resident r ON r.ResidentId = c.ResidentId JOIN Apartment b ON b.ApartmentId = r.ApartmentId
-> WHERE r.ApartmentId IN (SELECT a.ApartmentId
->                        FROM Apartment a
->                        WHERE a.State = 'AK') LIMIT 15;

```

CommentId	OverallScore	NoiseScore	LocationScore	WifiScore	FeedbackText	State
3999	1	1	3	1	Foreign skin interview all reach education building activity. Interview somebody Mrs many strategy possible camera.	AK
7494	1	4	4	4	Enough page dog sea you. Outside health action total.	AK
8247	2	1	4	1	Good deal than mention argue leave box. Meet last fear plant soon size national board.	AK
9312	3	3	4	1	Stuff trade might yard rock. Big black window add PM teach.	AK
5709	3	2	3	3	Among measure wear maybe. Ever decade stage meeting drop.	AK
5441	4	4	2	1	End view approach lose ok. Move the peace former candidate buy run.	AK
8677	2	4	2	4	Understand personal employee page manager. Response radio she human low movement then college.	AK
9480	2	1	4	3	Participant do interest film. Space first fear itself area.	AK
1309	2	1	3	2	Player relationship responsibility sense. Receive training old among reality magazine.	AK
7913	1	2	3	1	Government former particularly bad write detail particularly. Single especially security.	AK
9676	4	3	2	3	Threat health stop position spend white state. Best coach dinner begin your.	AK
9296	2	3	2	1	Alone himself through really car. Movie ability reflect attorney bit good.	AK
3988	4	4	3	1	Herself town manager close. He rock amount we force dream.	AK
388	3	4	4	2	Current environment rock director. Open effort pass from.	AK
9121	2	1	3	3	Almost us next why. Student hour program evidence major.	AK

15 rows in set (0.01 sec)

Indexing

To demonstrate performance before and after, here are the results of running EXPLAIN ANALYZE on **Query 1** (as denoted above)

BEFORE indexing:

```

mysql> EXPLAIN ANALYZE SELECT Management.Name, AVG(OverallAvgScore) AS MgmtAvg FROM Management JOIN Apartment ON Management.Name = Apartment.ManagementName NATURAL JOIN Review GROUP BY Management.Name
HAVING MgmtAvg > 1.25;

```

```

+-----+
| EXPLAIN |
+-----+

```

```

+-----+
| -> Filter: (MgmtAvg > 1.25) (actual time=124.031..124.038 rows=17 loops=1)
-> Table scan on <temporary> (actual time=0.002..0.004 rows=20 loops=1)
-> Aggregate using temporary table (actual time=124.021..124.025 rows=20 loops=1)
-> Nested loop inner join (cost=2306.00 rows=1000) (actual time=67.660..122.947 rows=1000 loops=1)
-> Nested loop inner join (cost=1206.00 rows=1000) (actual time=65.412..118.833 rows=1000 loops=1)
-> Table scan on Review (cost=106.00 rows=1000) (actual time=50.483..78.066 rows=1000 loops=1)
-> Single-row index lookup on Apartment using PRIMARY (ApartmentId=Review.ApartmentId) (cost=1.00 rows=1) (actual time=0.040..0.040 rows=1 loops=1000)
-> Single-row index lookup on Management using PRIMARY (Name=Apartment.ManagementName) (cost=1.00 rows=1) (actual time=0.004..0.004 rows=1 loops=1000)
+-----+

```

```

1 row in set (0.13 sec)

mysql>

```

AFTER indexing on Management Name in Management:

```

mysql> CREATE INDEX management_name ON Management(Name);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> EXPLAIN ANALYZE SELECT Management.Name, AVG(OverallAvgScore) AS MgmtAvg
-> FROM Management JOIN Apartment ON Management.Name = Apartment.ManagementName NATURAL JOIN Review
-> GROUP BY Management.Name
-> HAVING MgmtAvg > 1.25;

```

```

+-----+
| EXPLAIN |
+-----+

```

```

+-----+
| -> Filter: (MgmtAvg > 1.25) (actual time=13.733..13.743 rows=17 loops=1)
-> Table scan on <temporary> (actual time=0.001..0.003 rows=20 loops=1)
-> Aggregate using temporary table (actual time=13.709..13.714 rows=20 loops=1)
-> Nested loop inner join (cost=801.50 rows=1000) (actual time=0.143..12.168 rows=1000 loops=1)
-> Nested loop inner join (cost=431.50 rows=1000) (actual time=0.123..9.633 rows=1000 loops=1)
-> Table scan on Review (cost=101.50 rows=1000) (actual time=0.088..3.882 rows=1000 loops=1)
-> Single-row index lookup on Apartment using PRIMARY (ApartmentId=Review.ApartmentId) (cost=0.25 rows=1) (actual time=0.005..0.005 rows=1 loops=1000)
-> Single-row index lookup on Management using PRIMARY (Name=Apartment.ManagementName) (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=1000)
+-----+

```

```

1 row in set (0.02 sec)

mysql>

```

- AFTER indexing on Management Name in Apartment:**

```
mysql> 
```


Here are the results of running EXPLAIN ANALYZE on **Query 2** (as denoted above)

BEFORE indexing:

[illegible]

AFTER indexing on ApartmentId and State from the Apartment table:

```
mysql> CREATE INDEX ap_id_and_state ON Apartment(ApartmentId, State);

Query OK, 0 rows affected (0.24 sec)
Records: 0 Duplicates: 0 Warnings: 0


mysql> EXPLAIN ANALYZE SELECT CommentId, OverallScore, NoiseScore, LocationScore, WifiScore, FeedbackText, State
-> FROM Comment c JOIN Resident r ON r.ResidentId = c.ResidentId JOIN Apartment b ON b.ApartmentId = r.ApartmentId
-> WHERE r.ApartmentId IN (SELECT a.ApartmentId
-> FROM Apartment a
-> WHERE a.State = 'AK');
+-----+
|      |
+-----+
| EXPLAIN
+-----+
|
+-----+
|
+-----+
| -> Nested loop inner join (cost=524.59 rows=246) (actual time=53.927..67.394 rows=24 loops=1)
|   -> Nested loop inner join (cost=253.46 rows=159) (actual time=33.491..34.228 rows=23 loops=1)
|     -> Nested loop inner join (cost=137.50 rows=100) (actual time=0.072..0.766 rows=21 loops=1)
|       -> Filter: (a.State = 'AK') (cost=102.50 rows=100) (actual time=0.059..0.388 rows=21 loops=1)
|         -> Index scan on a using ap_id_and_state (cost=102.50 rows=100) (actual time=0.039..0.379 rows=1000 loops=1)
|           -> Single-row index lookup on b using PRIMARY (ApartmentId=a.ApartmentId) (cost=0.25 rows=1) (actual time=0.008..0.008 rows=1 loops=21)
|             -> Index lookup on r using ApartmentId (ApartmentId=a.ApartmentId) (cost=1.00 rows=2) (actual time=1.592..1.593 rows=1 loops=21)
|               -> Index lookup on c using ResidentId (ResidentId=r.ResidentId) (cost=1.55 rows=2) (actual time=1.440..1.441 rows=1 loops=23)
|
+-----+
|
+-----+
|
+-----+
| 1 row in set (0.07 sec)
|
mysql>
```

AFTER indexing on ApartmentId and ResidentId from the Resident table:

```
mysql> CREATE INDEX ap_res_id ON Resident(ApartmentId, ResidentId);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> EXPLAIN ANALYZE SELECT CommentId, OverallScore, NoiseScore, LocationScore, WifiScore, FeedbackText, State
-> FROM Comment c JOIN Resident r ON r.ResidentId = c.ResidentId JOIN Apartment b ON b.ApartmentId = r.ApartmentId
-> WHERE r.ApartmentId IN (SELECT a.ApartmentId
-> FROM Apartment a
-> WHERE a.State = 'AK');
+-----+-----+
| EXPLAIN |
+-----+-----+
+-----+-----+
-> Nested loop inner join (cost=458.72 rows=246) (actual time=21.541..61.036 rows=24 loops=1)
-> Nested loop inner join (cost=187.59 rows=159) (actual time=0.090..0.965 rows=23 loops=1)
-> Nested loop inner join (cost=146.67 rows=100) (actual time=0.070..0.813 rows=21 loops=1)
-> Filter: (a.State = 'AK') (cost=103.33 rows=100) (actual time=0.086..0.605 rows=21 loops=1)
-> Index scan on a using ap_id_and_state (cost=103.33 rows=1000) (actual time=0.036..0.368 rows=1000 loops=1)
-> Single-row index lookup on b using PRIMARY (ApartmentId=a.ApartmentId) (cost=0.33 rows=1) (actual time=0.009..0.009 rows=1 loops=21)
-> Index lookup on r using ap_res_id (ApartmentId=a.ApartmentId) (cost=0.25 rows=2) (actual time=0.051..0.007 rows=1 loops=21)
-> Index lookup on c using ResidentId (ResidentId=r.ResidentId) (cost=1.55 rows=2) (actual time=2.610..2.611 rows=1 loops=23)
+-----+-----+
1 row in set (0.06 sec)

mysql>
```

AFTER indexing on CommentId and ResidentId from the Comment table:

```
Database changed
mysql> CREATE INDEX com_res_id ON Comment(CommentId, ResidentId);
Query OK, 0 rows affected (0.15 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> EXPLAIN ANALYZE SELECT CommentId, OverallScore, NoiseScore, LocationScore, WifiScore, FeedbackText, State
-> FROM Comment c JOIN Resident r ON r.ResidentId = c.ResidentId JOIN Apartment b ON b.ApartmentId = r.ApartmentId
-> WHERE r.ApartmentId IN (SELECT a.ApartmentId
-> FROM Apartment a
-> WHERE a.State = 'AK');
+-----+-----+
| EXPLAIN |
+-----+-----+
+-----+-----+
-> Nested loop inner join (cost=347.19 rows=246) (actual time=48.790..76.179 rows=24 loops=1)
-> Nested loop inner join (cost=260.92 rows=159) (actual time=34.696..61.975 rows=23 loops=1)
-> Nested loop inner join (cost=220.00 rows=100) (actual time=34.637..61.785 rows=21 loops=1)
-> Filter: (a.State = 'AK') (cost=110.00 rows=100) (actual time=21.993..22.516 rows=21 loops=1)
-> Index scan on a using ap_id_and_state (cost=110.00 rows=1000) (actual time=21.964..22.290 rows=1000 loops=1)
-> Single-row index lookup on b using PRIMARY (ApartmentId=a.ApartmentId) (cost=1.00 rows=1) (actual time=1.869..1.869 rows=1 loops=21)
-> Index lookup on r using ap_res_id (ApartmentId=a.ApartmentId) (cost=0.25 rows=2) (actual time=0.097..0.008 rows=1 loops=21)
-> Index lookup on c using ResidentId (ResidentId=r.ResidentId) (cost=0.39 rows=2) (actual time=0.616..0.617 rows=1 loops=23)
+-----+-----+
1 row in set (0.08 sec)

mysql>
```

- Clearly, the second index is going to be the most efficient index to use for query 2 because within the WHERE clause of condition 2 we are finding when apartment IDs and resident IDs are equal. By creating an index on both of these attributes, we determined that the query will be most optimized.