

Stage 3

CS 411 Team 012

GCP Terminal Connection

The screenshot displays the Google Cloud Platform console interface. At the top, the navigation bar shows 'Google Cloud' and 'CS411 Final Project'. A search bar is available on the right. The left sidebar contains a menu with options: Overview, System insights, Query insights, Connections, Users, Databases, Backups, Replicas, and Release notes. The main content area is titled 'Overview' and shows details for the SQL instance 'cs411-final-project', which is a MySQL 8.0 instance. Below the instance name, there is a chart titled 'CPU utilisation' showing usage over time. The chart has a dropdown menu set to 'CPU utilisation' and a time range selector set to '1 day'. The chart shows a blue area representing CPU usage, with peaks around 18:00 and 22:00 UTC-5. Below the chart, there is a link to 'Go to Query insights for more in-depth info on queries and performance'. At the bottom, a terminal window is open, showing the output of MySQL commands. The terminal has a title bar 'Terminal (cs411-final-project-392323)' and an 'Open editor' button. The terminal output shows the following commands and results:

```
mysql> show databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'databases' at line 1
mysql> show databases;
+-----+
| Database |
+-----+
| chronic_illness_data |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_chronic_illness_data |
+-----+
| CheckIns |
| Conditions |
| Symptoms |
| Treatments |
| Users |
+-----+
5 rows in set (0.00 sec)
```

DDL Commands

```
mysql> CREATE TABLE Users(user_id VARCHAR(255) NOT NULL, age INT NOT NULL, sex VARCHAR(255) NOT NULL, trackable_type VARCHAR(255) NOT NULL)ENGINE = InnoDB DEFAULT CHARSET=latin1;
```

```
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> CREATE TABLE CheckIns(user_id VARCHAR(255) NOT NULL, checkin_date VARCHAR(255) NOT NULL, symptom VARCHAR(255) DEFAULT NULL, severity VARCHAR(255) DEFAULT NULL, feedback VARCHAR(255) DEFAULT NULL)ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> CREATE TABLE Conditions(user_id VARCHAR(255) NOT NULL, trackable_type VARCHAR(255) NOT NULL, trackable_name VARCHAR(255) NOT NULL, trackable_value VARCHAR(255) NOT NULL)ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE Symptoms(user_id VARCHAR(255) NOT NULL, trackable_type VARCHAR(255) NOT NULL, trackable_name VARCHAR(255) NOT NULL, trackable_value VARCHAR(255) NOT NULL)ENGINE=InnoDB DEFAULT CHARSET=latin1;;
```

```
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> CREATE TABLE Treatments(user_id VARCHAR(255) NOT NULL, trackable_type VARCHAR(255) NOT NULL, trackable_name VARCHAR(255) NOT NULL, trackable_value VARCHAR(255) NOT NULL)ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_chronic_illness_data |
+-----+
| CheckIns                        |
| Conditions                     |
| Symptoms                       |
| Treatments                     |
| Users                          |
+-----+
```

```
5 rows in set (0.00 sec)
```

Number of Rows Per Table

```
mysql> SELECT COUNT(user_id) from Users;
+-----+
| COUNT(user_id) |
+-----+
|          754610 |
+-----+
1 row in set (0.08 sec)

mysql> SELECT COUNT(user_id) from CheckIns;
+-----+
| COUNT(user_id) |
+-----+
|          754610 |
+-----+
1 row in set (0.09 sec)

mysql> SELECT COUNT(user_id) from Conditions;
+-----+
| COUNT(user_id) |
+-----+
|          169921 |
+-----+
1 row in set (0.03 sec)

mysql> SELECT COUNT(user_id) from Symptoms;
+-----+
| COUNT(user_id) |
+-----+
|          486465 |
+-----+
1 row in set (0.05 sec)

mysql> SELECT COUNT(user_id) from Treatments;
+-----+
| COUNT(user_id) |
+-----+
|           98226 |
+-----+
1 row in set (0.01 sec)
```

Advanced Query 1

```
mysql> SELECT trackable_name, COUNT(trackable_name) FROM Treatments t WHERE t.user_id IN(SELECT DISTINCT s.user_id FROM Symptoms s JOIN Conditions c ON s.user_id = c.user_id WHERE s.trackable_name = 'Headache' OR c.trackable_name = 'Headache') GROUP BY trackable_name ORDER BY COUNT(trackable_name) DESC LIMIT 15;
```

```
+-----+  
| trackable_name | COUNT(trackable_name) |  
+-----+  
| Ibuprofen      | 1232 |  
| Levothyroxine  | 1084 |  
| Tramadol       | 918  |  
| Vitamin d      | 825  |  
| Sleep          | 809  |  
| Exercise       | 705  |  
| Gabapentin     | 570  |  
| Zofran         | 537  |  
| Naproxen       | 493  |  
| Wellbutrin     | 483  |  
| Plaquenil      | 459  |  
| Omeprazole     | 398  |  
| Methylphenidate | 386  |  
| Paracetamol    | 375  |  
| Cymbalta       | 370  |  
+-----+
```

15 rows in set (17.46 sec)

```
mysql> █
```

Indexing Analysis - Original EXPLAIN ANALYZE

```
| -> Limit: 15 row(s) (actual time=210458.604..210458.606 rows=15 loops=1)
  -> Sort: Count(trackable_name) DESC, limit input to 15 row(s) per chunk (actual time=210458.603..210458.604 rows=15 loops=1)
    -> Table scan on <temporary> (actual time=210458.126..210458.448 rows=1237 loops=1)
      -> Aggregate using temporary table (actual time=210458.123..210458.123 rows=1237 loops=1)
        -> Nested loop inner join (cost=97767254217.03 rows=977671959110) (actual time=210060.627..210426.236 rows=49148 loops=1)
          -> Table scan on t (cost=9844.55 rows=96923) (actual time=70.473..377.352 rows=98226 loops=1)
            -> Single-row index lookup on <subquery2> using <auto_distinct_key> (user_id=t.user_id) (actual time=2.138..2.138 rows=1 loops=98226)
              -> Materialize with deduplication (cost=4556131.39..4556131.39 rows=10087100) (actual time=209989.085..209989.085 rows=2185 loops=1)
                -> Nested loop inner join (cost=3547421.43 rows=10087100) (actual time=89.583..206625.157 rows=4696834 loops=1)
                  -> Table scan on c (cost=16936.55 rows=166883) (actual time=16.699..759.779 rows=169921 loops=1)
                    -> Filter: ((s.trackable_name = 'Headache') or (c.trackable_name = 'Headache')) (cost=15.11 rows=60) (actual time=0.442..1.209 rows=28 loops=169921)
                      -> Index lookup on s using user_id_idx (user_id=c.user_id) (cost=15.11 rows=60) (actual time=0.008..1.089 rows=672 loops=169921)

```

Indexing Analysis after adding Index on Symptoms(user_id) and Conditions(user_id)

```
| -> Limit: 15 row(s) (actual time=204444.235..204444.237 rows=15 loops=1)
  -> Sort: Count(trackable_name) DESC, limit input to 15 row(s) per chunk (actual time=204444.234..204444.235 rows=15 loops=1)
    -> Table scan on <temporary> (actual time=204443.793..204444.080 rows=1237 loops=1)
      -> Aggregate using temporary table (actual time=204443.791..204443.791 rows=1237 loops=1)
        -> Nested loop inner join (cost=73231965958.13 rows=732319076521) (actual time=204174.838..204288.569 rows=49148 loops=1)
          -> Table scan on t (cost=9844.55 rows=96923) (actual time=0.031..58.710 rows=98226 loops=1)
            -> Single-row index lookup on <subquery2> using <auto_distinct_key> (user_id=t.user_id) (actual time=2.079..2.079 rows=1 loops=98226)
              -> Materialize with deduplication (cost=3416992.10..3416992.10 rows=7555679) (actual time=204174.668..204174.668 rows=2185 loops=1)
                -> Nested loop inner join (cost=2661424.20 rows=7555679) (actual time=0.354..201215.759 rows=4696834 loops=1)
                  -> Table scan on c (cost=16936.55 rows=166883) (actual time=0.008..371.472 rows=169921 loops=1)
                    -> Filter: ((s.trackable_name = 'Headache') or (c.trackable_name = 'Headache')) (cost=11.32 rows=45) (actual time=0.432..1.178 rows=28 loops=169921)
                )
              -> Index lookup on s using symptomuser (user_id=c.user_id) (cost=11.32 rows=45) (actual time=0.008..1.061 rows=672 loops=169921)
            )
          )
        )
      )
    )
  )
|
```

Added indexes on user_id for Symptoms and Conditions as both attributes are used for Join in subquery. This improved the cost of the nested loop inner join greatly which is where these attributes are used in the query.

Indexing Analysis after adding Index on Symptoms(trackable_name) and Conditions(trackable_name)

```
| -> Limit: 15 row(s) (actual time=203910.580..203910.581 rows=15 loops=1)
  -> Sort: Count(trackable_name) DESC, limit input to 15 row(s) per chunk (actual time=203910.579..203910.579 rows=15 loops=1)
    -> Table scan on <temporary> (actual time=203910.109..203910.391 rows=1237 loops=1)
      -> Aggregate using temporary table (actual time=203910.107..203910.107 rows=1237 loops=1)
        -> Nested loop inner join (cost=97767254217.03 rows=977671959110) (actual time=203517.276..203881.611 rows=49148 loops=1)
          -> Table scan on t (cost=9844.55 rows=96923) (actual time=0.042..183.916 rows=98226 loops=1)
            -> Single-row index lookup on <subquery2> using <auto_distinct_key> (user_id=t.user_id) (actual time=2.074..2.074 rows=1 loops=98226)
              -> Materialize with deduplication (cost=4556131.39..4556131.39 rows=10087100) (actual time=203517.088..203517.088 rows=2185 loops=1)
                -> Nested loop inner join (cost=3547421.43 rows=10087100) (actual time=0.544..200937.587 rows=4696834 loops=1)
                  -> Table scan on c (cost=16936.55 rows=166883) (actual time=0.013..241.499 rows=169921 loops=1)
                    -> Filter: ((s.trackable_name = 'Headache') or (c.trackable_name = 'Headache')) (cost=15.11 rows=60) (actual time=0.447..1.179 rows=28 loops=169921)
                )
              -> Index lookup on s using user_id_idx (user_id=c.user_id) (cost=15.11 rows=60) (actual time=0.005..1.047 rows=672 loops=169921)
            )
          )
        )
      )
    )
  )
|
```

Added indexes on trackable_name for Symptoms and Conditions as both attributes are used for Where in subquery. This did not have a great effect on the performance of the query which is most likely due to the cost already being low for the filter and it happening near the end of the query.

Indexing Analysis after adding Index on Treatments(user_id) and Treatments(trackable_name)

```
| -> Limit: 15 row(s) (actual time=196821.268..196821.271 rows=15 loops=1)
  -> Sort: Count(trackable_name) DESC, limit input to 15 row(s) per chunk (actual time=196821.266..196821.268 rows=15 loops=1)
    -> Table scan on <temporary> (actual time=196820.533..196821.005 rows=1237 loops=1)
      -> Aggregate using temporary table (actual time=196820.531..196820.531 rows=1237 loops=1)
        -> Nested loop inner join (cost=97767254217.03 rows=977671959110) (actual time=196337.597..196774.827 rows=49148 loops=1)
          -> Table scan on t (cost=9844.55 rows=96923) (actual time=0.033..347.271 rows=98226 loops=1)
            -> Single-row index lookup on <subquery2> using <auto_distinct_key> (user_id=t.user_id) (actual time=1.999..2.000 rows=1 loops=98226)
              -> Materialize with deduplication (cost=4556131.39..4556131.39 rows=10087100) (actual time=196337.341..196337.341 rows=2185 loops=1)
                -> Nested loop inner join (cost=3547421.43 rows=10087100) (actual time=0.408..193827.619 rows=4696834 loops=1)
                  -> Table scan on c (cost=16936.55 rows=166883) (actual time=0.012..465.390 rows=169921 loops=1)
                    -> Filter: ((s.trackable_name = 'Headache') or (c.trackable_name = 'Headache')) (cost=15.11 rows=60) (actual time=0.414..1.134 rows=28 loops=169921)
                )
              -> Index lookup on s using user_id_idx (user_id=c.user_id) (cost=15.11 rows=60) (actual time=0.004..1.012 rows=672 loops=169921)
            )
          )
        )
      )
    )
  )
|
```

Added indexes on user_id and trackable_name for Treatments as the attributes are used for Order By and Where. This improved the cost of the overall query and in the sorting and table scan in the beginning.

Advanced Query 2

```
mysql> SELECT c.trackable_name as conditions,COUNT(c.trackable_name) as conditionCount FROM Conditions c WHERE c.user_id IN(SELECT s.user_id FROM Symptoms s WHERE (s.trackable_name = 'fatigue' OR s.trackable_name = 'Nausea') AND s.trackable_value >= 3) GROUP BY conditions ORDER BY conditionCount DESC LIMIT 15;
```

conditions	conditionCount
Fibromyalgia	5916
Depression	5613
Anxiety	4828
Chronic fatigue syndrome	3135
Migraine	2576
Ehlers-Danlos syndrome	1888
Asthma	1681
IBS	1502
Irritable bowel syndrome	1338
Endometriosis	1088
Headaches	1062
POTS	1060
Chronic Migraines	986
Postural Orthostatic Tachycardia Syndrome	904
Fatigue	894

15 rows in set (0.50 sec)

```
-> Limit: 15 row(s) (actual time=599.911..599.913 rows=15 loops=1)
-> Sort: conditionCount DESC, limit input to 15 row(s) per chunk (actual time=599.911..599.912 rows=15 loops=1)
-> Table scan on <temporary> (actual time=599.357..599.726 rows=1354 loops=1)
-> Aggregate using temporary table (actual time=599.355..599.355 rows=1354 loops=1)
-> Nested loop inner join (cost=512914695.52 rows=5128810707) (actual time=362.964..553.205 rows=82998 loops=1)
-> Table scan on c (cost=16936.55 rows=166883) (actual time=0.031..97.423 rows=169921 loops=1)
-> Single-row index lookup on <subquery2> using <auto_distinct key> (user_id=c.user_id) (actual time=0.003..0.003 rows=0 loops=169921)
-> Materialize with deduplication (cost=52268.90..52268.90 rows=30733) (actual time=362.879..362.879 rows=2506 loops=1)
-> Filter: (((s.trackable_name = 'fatigue') or (s.trackable_name = 'Nausea')) and (s.trackable_value >= 3)) (cost=49195.60 rows=30733) (actual time=0.055..357.459 rows=9789 loops=1)
-> Table scan on s (cost=49195.60 rows=485306) (actual time=0.013..283.179 rows=486465 loops=1)

+-----+
|
+-----+
1 row in set (0.61 sec)
```

Indexing Analysis after adding Index on Symptoms(trackable_name)

Since we are specifically finding entries where `s.trackable_name` is either `fatigue` or `nausea`, we added an index on `Symptoms(trackable_name)`. By creating this index, we see that time is significantly reduced and the cost is improved of the overall query.

Adding a index on Symptoms(user_id) essentially does not have a great effect on the query as you still have to select all the user Id from Symptoms that match the conditions first before you select the condition name and count

Indexing Analysis after adding Index on Symptoms(trackable_value) and Conditions(user_id)

```

-> Limit: 15 row(s) (actual time=421.502..421.504 rows=15 loops=1)
  -> Sort: conditionCount DESC, limit input to 15 row(s) per chunk (actual time=421.502..421.503 rows=15 loops=1)
    -> Table scan on <temporary> (actual time=421.016..421.377 rows=1354 loops=1)
      -> Aggregate using temporary table (actual time=421.014..421.014 rows=1354 loops=1)
        -> Nested loop inner join (cost=512914695.52 rows=5128810707) (actual time=232.924..385.048 rows=82998 loops=1)
          -> Table scan on c (cost=16936.55 rows=166883) (actual time=0.024..68.352 rows=169921 loops=1)
          -> Single-row index lookup on <subquery2> using <auto distinct key> (user_id=c.user_id) (actual time=0.002..0.002 rows=0 loops=169921)
            -> Materialize with deduplication (cost=52268.90..52268.90 rows=30733) (actual time=232.873..232.873 rows=2506 loops=1)
              -> Filter: (((s.trackable_name = 'fatigue') or (s.trackable_name = 'Nausea')) and (s.trackable_value >= 3)) (cost=49195.60 rows=30733) (actual time=0.024..228.455 rows=9789 loops=1)
                -> Table scan on s (cost=49195.60 rows=485306) (actual time=0.007..180.440 rows=486465 loops=1)
1
+-----+
1 row in set, 1 warning (0.42 sec)

mysql>

```

Adding indexes on both Symptoms(trackable_value) and Conditions(user_id) does have an effect on the query as it improves the cost. We believe that this is due to the index on Symptoms(trackable_value) as it is specifically in the WHERE clause. However, this does not have as great an effect as creating an index on Symptoms(trackable_name).