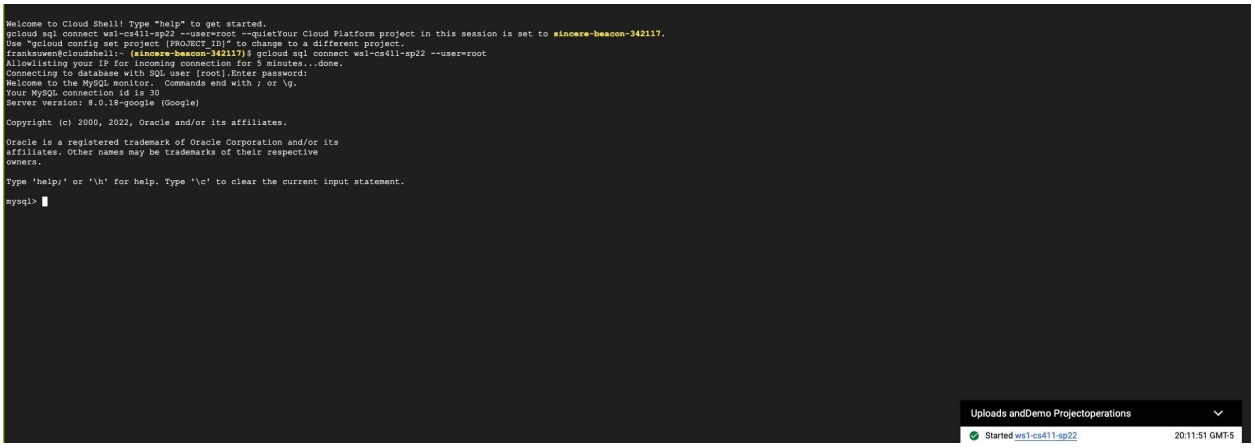
Screenshot of the connection:



DDL commands for out tables:

```sql
DROP TABLE IF EXISTS `news`;

CREATE TABLE `news` (
   `newsID` VARCHAR(255) NOT NULL PRIMARY KEY,
   `date` VARCHAR(255) NOT NULL,
   `title` VARCHAR(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `comments`;

CREATE TABLE `comments`(
   `commentID` INT NOT NULL PRIMARY KEY,
   `context` VARCHAR(255) NOT NULL,
   `date` VARCHAR(255) NOT NULL,
   `vote` INT NOT NULL,
   `newsID` VARCHAR(255) NOT NULL,
   `userID` INT NOT NULL,
   FOREIGN KEY (`newsID`) REFERENCES `news` (`newsID`),
   FOREIGN KEY (`userID`) REFERENCES `users` (`userID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `users`;
```

```sql
CREATE TABLE `users`(
   `userID` INT NOT NULL PRIMARY KEY,
   `name` VARCHAR(255) NOT NULL,
   `introduction` VARCHAR(255) NOT NULL,
   `password` VARCHAR(15) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `categories`;

CREATE TABLE `categories`(
   `categoryName` VARCHAR(255) NOT NULL PRIMARY KEY,
   `count` INT NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `statistics`;

CREATE TABLE `statistics`(
   `date` VARCHAR(255) NOT NULL,
   `count` INT NOT NULL,
   `length` INT NOT NULL,
   `userID` INT NOT NULL,
   FOREIGN KEY (`userID`) REFERENCES `users` (`userID`),
   PRIMARY KEY(`date`, `userID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


DROP TABLE IF EXISTS `have`;

CREATE TABLE `have`(
   `categoryName` VARCHAR(255) NOT NULL,
   `newsID` VARCHAR (255) NOT NULL,
   FOREIGN KEY  (`categoryName`) REFERENCES `categories` (`categoryName`),
   FOREIGN KEY  (`newsID`) REFERENCES `news`(`newsID`),
   PRIMARY KEY(`categoryName`, `newsID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Result of count query to show that there are at least 1000 rows:

```
+-----------------------+
| Tables_in_forummodels |
+-----------------------+
| categories            |
| comments              |
| have                  |
| news                  |
| statistics            |
| users                 |
+-----------------------+
6 rows in set (0.01 sec)

mysql> SELECT count(*) FROM news;
+----------+
| count(*) |
+----------+
|     4482 |
+----------+
1 row in set (0.13 sec)

mysql> SELECT count(*) FROM comments;
+----------+
| count(*) |
+----------+
|     2086 |
+----------+
1 row in set (0.15 sec)

mysql> SELECT count(*) FROM users;
+----------+
| count(*) |
+----------+
|     1869 |
+----------+
1 row in set (0.06 sec)
```

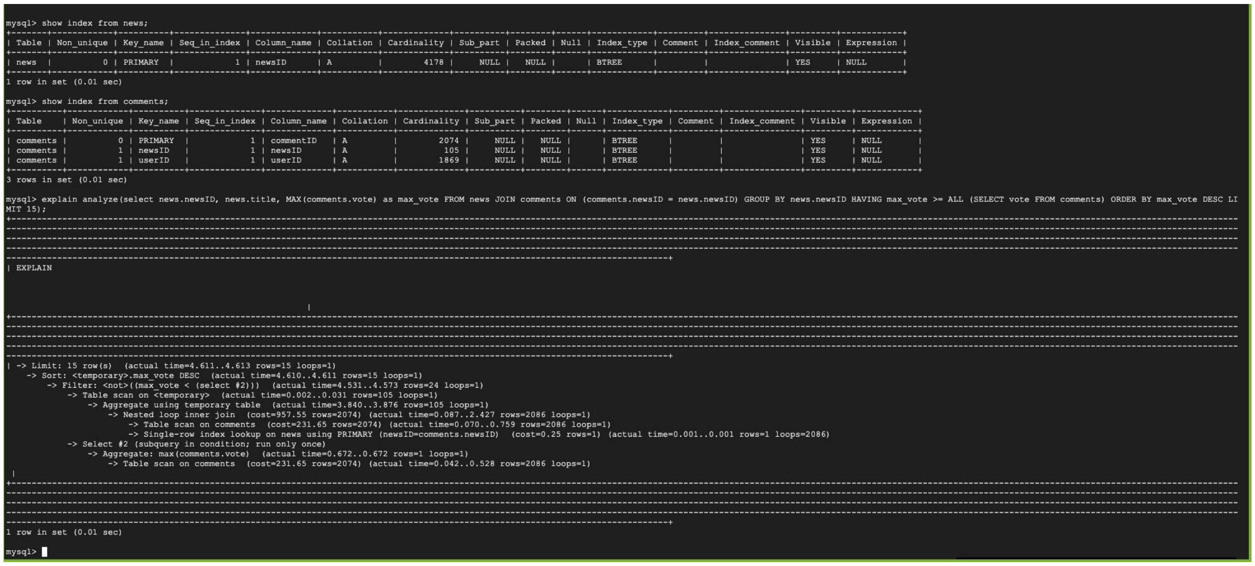
Advanced query1:

```
mysql> select news.newsID, news.title, MAX(comments.vote) as max_vote FROM news JOIN comments ON (comments.newsID = news.newsID) GROUP BY news.newsID HAVING max_vote >= ALL (SELECT vote FROM comments) ORDER BY max_vote DESC LIMIT 15;
+--------------------------------------------------------+----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------+----------+
| newsID                                                 | title
         | max_vote |
+--------------------------------------------------------+----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------+----------+
| nyt://article/76ac575d-7772-537c-b901-cc726e127fbe     | During the contentious meeting, the president made his case for a border wall and rejected Democrats’ proposals for reopening the government while the two sides ironed out their diffe
rences.  |       99 |
| nyt://article/1e9b2c2a-35a9-5939-9af8-d10f448a9ede     | Both sides seemed to be standing firm, suggesting no easy end to a funding impasse that shuttered a large portion of the federal government.
         |       99 |
| nyt://article/4b40e607-f777-52f6-92bf-1b611968513c     | A December gain of 312,000 jobs sent stocks up more than 3 percent, and the Federal Reserve chairman offered soothing words on the central bank’s plans.
         |       99 |
| nyt://article/95e4d0f7-0081-5e92-a820-1cfc91b48346     | The Israeli author will be spared the littleness of Israeli and Palestinian leaders.
         |       99 |
| nyt://article/69ba3051-d7b1-5c46-b225-89a01e66a364     | In the On Politics newsletter, the government shutdown threatens workers living paycheck to paycheck. Plus, a dispatch from Elizabeth Warren’s Iowa trip.
         |       99 |
| nyt://article/87bd742b-41f3-5006-9be7-39acd9725261     | True freshman quarterback Trevor Lawrence led the Tigers to their second championship in three years.
         |       99 |
| nyt://article/fbc8b40c-8319-544c-b58b-1d4be2739191     | Much about the case has been cloaked in secrecy, but it involves a sealed grand jury subpoena to a foreign corporation issued by a federal prosecutor who may be Robert S. Mueller III.
         |       99 |
| nyt://interactive/37c640d0-9565-5885-b30c-f40a3becb5f3 | A starter kit for escaping into the world.
         |       99 |
| nyt://article/1148d15b-d515-5d13-bc8a-bf4bd415ae1f     | What we can learn from “Canada in a thong.”
         |       99 |
| nyt://article/7a5029a5-6806-5df0-baec-079b7454dc4c     | Ari Folman and David Polonsky’s graphic novel adaptation of the famous diary brings out new and vibrant aspects of Anne’s voice.
         |       99 |
| nyt://article/9794f233-61ca-565c-8c6b-85df03b6c6b9     | A Polish citizen who works for the French telecommunications company Orange was also charged, officials said.
         |       99 |
| nyt://article/63565f2c-17bf-5381-8f05-4bc685d303c4     | Touring eight countries in one week, Secretary of State Mike Pompeo sought to build a regional coalition against Iran. The obstacles are formidable.
         |       99 |
| nyt://article/2b377853-5eef-57dd-b198-94703adaf0f3     | American governance shuts down.
         |       99 |
| nyt://article/5f016c1f-4f30-59dd-aacb-e7947ba615e2     | The president’s personal lawyer had left open the possibility in a television interview that campaign aides took part in Russia’s election interference.
         |       99 |
| nyt://article/e53b77dd-ce15-52c3-a940-75816b4654f0     | In his enjoyable, messy sequel to “Unbreakable” and “Split,” Shyamalan joins together three adversaries in a world of ordinary terrors.
         |       99 |
+--------------------------------------------------------+----------------------------------------------------------------------------------------------------------------------------------------------------------------
----------+----------+
15 rows in set (0.00 sec)

mysql>
```

Advanced query2:

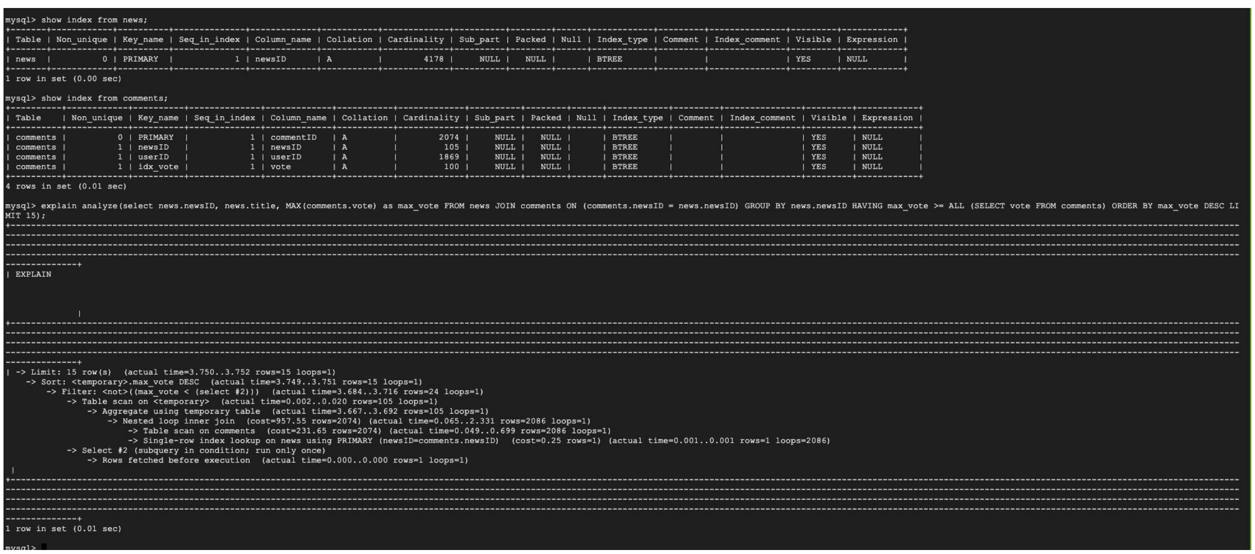
```
mysql> (SELECT users.name FROM users WHERE users.name = 'J%') UNION (SELECT users.name FROM users JOIN comments ON users.userID = comments.userID WHERE comments.vote = 90) ORDER BY name LIMIT 15;
+-------------------+
| name              |
+-------------------+
| Ben Lieberman     |
| Brazilianheat     |
| cb                |
| David             |
| Douglas Jay       |
| Edward            |
| fish out of Water |
| Julia             |
| Larry L           |
| Prunella Arnold   |
| tfair             |
| Tino              |
| Zareen            |
+-------------------+
13 rows in set (0.00 sec)
```

Query 1 - Before adding any indexing:



```
mysql> show index from news;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| news  |          0 | PRIMARY  |            1 | newsID      | A         |        4178 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
1 row in set (0.01 sec)

mysql> show index from comments;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| comments |          0 | PRIMARY  |            1 | commentID   | A         |        2074 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | newsID   |            1 | newsID      | A         |         105 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | userID   |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
3 rows in set (0.01 sec)

mysql> explain analyze(select news.newsID, news.title, MAX(comments.vote) as max_vote FROM news JOIN comments ON (comments.newsID = news.newsID) GROUP BY news.newsID HAVING max_vote >= ALL (SELECT vote FROM comments) ORDER BY max_vote DESC LI
MIT 15);
+----------------------------------------------------------------------------------------------------+
| EXPLAIN
                                                  |
+----------------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=4.611..4.613 rows=15 loops=1)
    -> Sort: <temporary>.max_vote DESC  (actual time=4.610..4.611 rows=15 loops=1)
        -> Filter: <not>((max_vote < (select #2)))  (actual time=4.531..4.573 rows=24 loops=1)
            -> Table scan on <temporary>  (actual time=0.002..0.031 rows=105 loops=1)
                -> Aggregate using temporary table  (actual time=3.840..3.876 rows=105 loops=1)
                    -> Nested loop inner join  (cost=957.55 rows=2074) (actual time=0.087..2.427 rows=2086 loops=1)
                        -> Table scan on comments  (cost=231.65 rows=2074) (actual time=0.070..0.759 rows=2086 loops=1)
                        -> Single-row index lookup on news using PRIMARY (newsID=comments.newsID)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=2086)
            -> Select #2 (subquery in condition; run only once)
                -> Aggregate: max(comments.vote)  (actual time=0.672..0.672 rows=1 loops=1)
                    -> Table scan on comments  (cost=231.65 rows=2074) (actual time=0.042..0.528 rows=2086 loops=1)
 |
+----------------------------------------------------------------------------------------------------+
1 row in set (0.01 sec)

mysql>
```

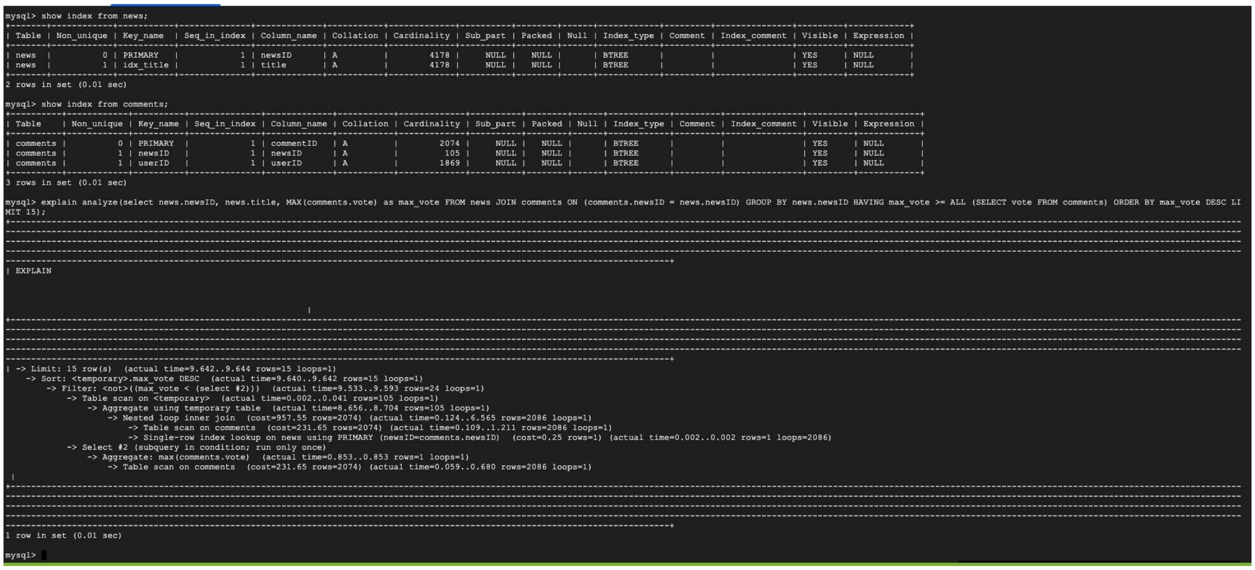
Query 1 - Indexing on vote:



```
mysql> show index from news;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| news  |          0 | PRIMARY  |            1 | newsID      | A         |        4178 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
1 row in set (0.00 sec)

mysql> show index from comments;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| comments |          0 | PRIMARY  |            1 | commentID   | A         |        2074 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | newsID   |            1 | newsID      | A         |         105 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | userID   |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | idx_vote |            1 | vote        | A         |         100 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
4 rows in set (0.01 sec)

mysql> explain analyze(select news.newsID, news.title, MAX(comments.vote) as max_vote FROM news JOIN comments ON (comments.newsID = news.newsID) GROUP BY news.newsID HAVING max_vote >= ALL (SELECT vote FROM comments) ORDER BY max_vote DESC LI
MIT 15);
+----------------------------------------------------------------------------------------------------+
| EXPLAIN
              |
+----------------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=3.750..3.752 rows=15 loops=1)
    -> Sort: <temporary>.max_vote DESC  (actual time=3.749..3.751 rows=15 loops=1)
        -> Filter: <not>((max_vote < (select #2)))  (actual time=3.684..3.716 rows=24 loops=1)
            -> Table scan on <temporary>  (actual time=0.002..0.020 rows=105 loops=1)
                -> Aggregate using temporary table  (actual time=3.667..3.692 rows=105 loops=1)
                    -> Nested loop inner join  (cost=957.55 rows=2074) (actual time=0.065..2.331 rows=2086 loops=1)
                        -> Table scan on comments  (cost=231.65 rows=2074) (actual time=0.049..0.699 rows=2086 loops=1)
                        -> Single-row index lookup on news using PRIMARY (newsID=comments.newsID)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=2086)
            -> Select #2 (subquery in condition; run only once)
                -> Rows fetched before execution  (actual time=0.000..0.000 rows=1 loops=1)
 |
+----------------------------------------------------------------------------------------------------+
1 row in set (0.01 sec)

mysql>
```

Explain: We decide to index on vote because we have a subquery that finds the max vote of comments, so we expect to add an index on vote will sort the vote attribute that will help us to find the max vote in O(1). As shown in the Select #2 of the image, rows have been fetched before execution, which improves the performance.

Query 1 - Indexing on title:



```
mysql> show index from news;
+-------+------------+-----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name  | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-------+------------+-----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| news  |          0 | PRIMARY   |            1 | newsID      | A         |        4178 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| news  |          1 | idx_title |            1 | title       | A         |        4178 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+-------+------------+-----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (0.01 sec)

mysql> show index from comments;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| comments |          0 | PRIMARY  |            1 | commentID   | A         |        2074 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | newsID   |            1 | newsID      | A         |         105 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | userID   |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
3 rows in set (0.01 sec)

mysql> explain analyze(select news.newsID, news.title, MAX(comments.vote) as max_vote FROM news JOIN comments ON (comments.newsID = news.newsID) GROUP BY news.newsID HAVING max_vote >= ALL (SELECT vote FROM comments) ORDER BY max_vote DESC LI
MIT 15);
+------------------------------------------------------------------------------------------------+
| EXPLAIN
                                                  |
+------------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=9.642..9.644 rows=15 loops=1)
    -> Sort: <temporary>.max_vote DESC  (actual time=9.640..9.642 rows=15 loops=1)
        -> Filter: <not>((max_vote < (select #2)))  (actual time=9.533..9.593 rows=24 loops=1)
            -> Table scan on <temporary>  (actual time=0.002..0.041 rows=105 loops=1)
                -> Aggregate using temporary table  (actual time=8.656..8.704 rows=105 loops=1)
                    -> Nested loop inner join  (cost=957.55 rows=2074) (actual time=0.124..6.565 rows=2086 loops=1)
                        -> Table scan on comments  (cost=231.65 rows=2074) (actual time=0.109..1.211 rows=2086 loops=1)
                        -> Single-row index lookup on news using PRIMARY (newsID=comments.newsID)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=2086)
            -> Select #2 (subquery in condition; run only once)
                -> Aggregate: max(comments.vote)  (actual time=0.853..0.853 rows=1 loops=1)
                    -> Table scan on comments  (cost=231.65 rows=2074) (actual time=0.059..0.680 rows=2086 loops=1)
 |
+------------------------------------------------------------------------------------------------+
1 row in set (0.01 sec)

mysql>
```

We decide to add an index on title since we want to output the title of selected news as the result of our query, and we hope the performance of the query will be optimized. However, since there isn't any further optimization involved shown by the SQL report, this particular indexing does not work. After further consideration, we think that this indexing does not improve the performance because we only want to output the title of news, while the order of the title of those news does not matter in the query, and we do not apply any constraint on titles on the query, which causes indexing on titles has no effect.

Query 1 - Indexing on both vote and title:

```
mysql> show index from news;
+-------+------------+-----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name  | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-------+------------+-----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| news  |          0 | PRIMARY   |            1 | newsID      | A         |        4178 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| news  |          1 | idx_title |            1 | title       | A         |        4178 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+-------+------------+-----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (0.01 sec)

mysql> show index from comments;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| comments |          0 | PRIMARY  |            1 | commentID   | A         |        2074 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | newsID   |            1 | newsID      | A         |         105 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | userID   |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | idx_vote |            1 | vote        | A         |         100 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
4 rows in set (0.01 sec)

mysql> explain analyze(select news.newsID, news.title, MAX(comments.vote) as max_vote FROM news JOIN comments ON (comments.newsID = news.newsID) GROUP BY news.newsID HAVING max_vote >= ALL (SELECT vote FROM comments) ORDER BY max_vote DESC LI
MIT 15);
+--------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------+
| EXPLAIN

              |
+--------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------+
| -> Limit: 15 row(s)  (actual time=3.853..3.855 rows=15 loops=1)
    -> Sort: <temporary>.max_vote DESC  (actual time=3.852..3.853 rows=15 loops=1)
        -> Filter: <not>((max_vote < (select #2)))  (actual time=3.780..3.817 rows=24 loops=1)
            -> Table scan on <temporary>  (actual time=0.002..0.025 rows=105 loops=1)
                -> Aggregate using temporary table  (actual time=3.763..3.793 rows=105 loops=1)
                    -> Nested loop inner join  (cost=957.55 rows=2074) (actual time=0.066..2.329 rows=2086 loops=1)
                        -> Table scan on comments  (cost=231.65 rows=2074) (actual time=0.049..0.710 rows=2086 loops=1)
                        -> Single-row index lookup on news using PRIMARY (newsID=comments.newsID)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=2086)
            -> Select #2 (subquery in condition; run only once)
                -> Rows fetched before execution  (actual time=0.000..0.000 rows=1 loops=1)
 |
+--------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------+
1 row in set (0.00 sec)

mysql> 
```

Since adding the index on vote works while title does not, we design our third implementation as adding indexes on both vote and titles, for the purpose of figuring out whether a bad indexing will affect the validity of a good indexing. As the performance report shows, this design has no difference with the design of having vote as the only index, where the subquery is still optimized, suggesting that different indexings are independent from each other (at least in our implementation, while we do think some poor designs may have negative impact on the performance of a optimized query).

Query 2 - Before adding any indexing:

```
mysql> show index from users;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| users |          0 | PRIMARY  |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
1 row in set (0.01 sec)

mysql> show index from comments;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| comments |          0 | PRIMARY  |            1 | commentID   | A         |        2074 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | newsID   |            1 | newsID      | A         |         105 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | userID   |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
3 rows in set (0.01 sec)

mysql> explain analyze((SELECT users.name FROM users WHERE users.name = 'J%') UNION (SELECT users.name FROM users JOIN comments ON users.userID = comments.userID WHERE comments.vote = 90) ORDER BY name LIMIT 15);
+----------------------------------------------------------------------------------------------------+
| EXPLAIN
                                                  |
+----------------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=0.019..0.021 rows=13 loops=1)
    -> Sort: <union temporary>.name, limit input to 15 row(s) per chunk  (cost=2.50 rows=0) (actual time=0.018..0.019 rows=13 loops=1)
        -> Table scan on <union temporary>  (actual time=0.001..0.002 rows=13 loops=1)
            -> Union materialize with deduplication  (actual time=1.381..1.384 rows=13 loops=1)
                -> Filter: (users.`name` = 'J%')  (cost=188.65 rows=187) (actual time=0.540..0.540 rows=0 loops=1)
                    -> Table scan on users  (cost=188.65 rows=1869) (actual time=0.038..0.429 rows=1869 loops=1)
                -> Nested loop inner join  (cost=304.24 rows=207) (actual time=0.055..0.803 rows=13 loops=1)
                    -> Filter: (comments.vote = 90)  (cost=231.65 rows=207) (actual time=0.045..0.762 rows=13 loops=1)
                        -> Table scan on comments  (cost=231.65 rows=2074) (actual time=0.043..0.637 rows=2086 loops=1)
                    -> Single-row index lookup on users using PRIMARY (userID=comments.userID)  (cost=0.25 rows=1) (actual time=0.003..0.003 rows=1 loops=13)
 |
+----------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

Query 2 - Indexing on vote:

```
mysql> show index from users;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| users |          0 | PRIMARY  |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
1 row in set (0.00 sec)

mysql> show index from comments;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| comments |          0 | PRIMARY  |            1 | commentID   | A         |        2074 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | newsID   |            1 | newsID      | A         |         105 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | userID   |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | idx_vote |            1 | vote        | A         |         100 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
4 rows in set (0.00 sec)

mysql> explain analyze((SELECT users.name FROM users WHERE users.name = 'J%') UNION (SELECT users.name FROM users JOIN comments ON users.userID = comments.userID WHERE comments.vote = 90) ORDER BY name LIMIT 15);
+----------------------------------------------------------------------------------------------------+
| EXPLAIN
                                          |
+----------------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=0.018..0.020 rows=13 loops=1)
    -> Sort: <union temporary>.name, limit input to 15 row(s) per chunk  (cost=2.50 rows=0) (actual time=0.018..0.018 rows=13 loops=1)
        -> Table scan on <union temporary>  (actual time=0.001..0.002 rows=13 loops=1)
            -> Union materialize with deduplication  (actual time=0.811..0.814 rows=13 loops=1)
                -> Filter: (users.`name` = 'J%')  (cost=188.65 rows=187) (actual time=0.552..0.552 rows=0 loops=1)
                    -> Table scan on users  (cost=188.65 rows=1869) (actual time=0.057..0.436 rows=1869 loops=1)
                -> Nested loop inner join  (cost=9.10 rows=13) (actual time=0.051..0.220 rows=13 loops=1)
                    -> Index lookup on comments using idx_vote (vote=90)  (cost=4.55 rows=13) (actual time=0.044..0.052 rows=13 loops=1)
                    -> Single-row index lookup on users using PRIMARY (userID=comments.userID)  (cost=0.26 rows=1) (actual time=0.013..0.013 rows=1 loops=13)
 |
+----------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

Explain: We decide to index on vote because we need to check where the vote on the particular comment is equal to 90 or not. We expect adding an index on vote will improve the performance of our query, since it should add some order to vote. After adding this index and analyzing the performance of this design, the query does show a better performance, in the way that it changes from table scan to index lookup on the attribute vote.

Query 2 - Indexing on name:

```
mysql> show index from users;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| users |          0 | PRIMARY  |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| users |          1 | idx_name |            1 | name        | A         |        1672 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (0.01 sec)

mysql> show index from comments;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| comments |          0 | PRIMARY  |            1 | commentID   | A         |        2074 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | newsID   |            1 | newsID      | A         |         105 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | userID   |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
3 rows in set (0.00 sec)

mysql> explain analyze((SELECT users.name FROM users WHERE users.name = 'J%') UNION (SELECT users.name FROM users JOIN comments ON users.userID = comments.userID WHERE comments.vote = 90) ORDER BY name LIMIT 15);
+------------------------------------------------------------
| EXPLAIN
                                                    |
+------------------------------------------------------------
| -> Limit: 15 row(s)  (actual time=0.019..0.021 rows=13 loops=1)
    -> Sort: <union temporary>.name, limit input to 15 row(s) per chunk  (cost=2.50 rows=0) (actual time=0.018..0.019 rows=13 loops=1)
        -> Table scan on <union temporary>  (actual time=0.001..0.002 rows=13 loops=1)
            -> Union materialize with deduplication  (actual time=0.904..0.906 rows=13 loops=1)
                -> Index lookup on users using idx_name (name='J%')  (cost=0.85 rows=1) (actual time=0.008..0.008 rows=0 loops=1)
                -> Nested loop inner join  (cost=304.24 rows=207) (actual time=0.059..0.862 rows=13 loops=1)
                    -> Filter: (comments.vote = 90)  (cost=231.65 rows=207) (actual time=0.050..0.767 rows=13 loops=1)
                        -> Table scan on comments  (cost=231.65 rows=2074) (actual time=0.046..0.642 rows=2086 loops=1)
                    -> Single-row index lookup on users using PRIMARY (userID=comments.userID)  (cost=0.25 rows=1) (actual time=0.007..0.007 rows=1 loops=13)
 |
+------------------------------------------------------------
1 row in set (0.00 sec)
```

Explain: We decide to index on name because we need to check where the name on the particular user starts with “J” or not. We expect adding an index on name will improve the performance of our query, since it should add some order to name. After adding this index and analyzing the performance of this design, the query does show a better performance, in the way that it changes from table scan to index lookup on the attribute name.

Query 2: Indexing on both name and vote:

```
mysql> create index idx_vote on comments(vote);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from users;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| users |          0 | PRIMARY  |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| users |          1 | idx_name |            1 | name        | A         |        1672 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (0.01 sec)

mysql> show index from comments;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| comments |          0 | PRIMARY  |            1 | commentID   | A         |        2074 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | newsID   |            1 | newsID      | A         |         105 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | userID   |            1 | userID      | A         |        1869 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| comments |          1 | idx_vote |            1 | vote        | A         |         100 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
4 rows in set (0.01 sec)

mysql> explain analyze((SELECT users.name FROM users WHERE users.name = 'J%') UNION (SELECT users.name FROM users JOIN comments ON users.userID = comments.userID WHERE comments.vote = 90) ORDER BY name LIMIT 15);
+------------------------------------------------------------------------------------------------+
| EXPLAIN                                                                                        |
+------------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=0.014..0.016 rows=13 loops=1)
    -> Sort: <union temporary>.name, limit input to 15 row(s) per chunk  (cost=2.50 rows=0) (actual time=0.014..0.015 rows=13 loops=1)
        -> Table scan on <union temporary>  (actual time=0.001..0.002 rows=13 loops=1)
            -> Union materialize with deduplication  (actual time=0.125..0.128 rows=13 loops=1)
                -> Index lookup on users using idx_name (name='J%')  (cost=0.85 rows=1) (actual time=0.011..0.011 rows=0 loops=1)
                -> Nested loop inner join  (cost=9.10 rows=13) (actual time=0.054..0.088 rows=13 loops=1)
                    -> Index lookup on comments using idx_vote (vote=90)  (cost=4.55 rows=13) (actual time=0.046..0.049 rows=13 loops=1)
                    -> Single-row index lookup on users using PRIMARY (userID=comments.userID)  (cost=0.26 rows=1) (actual time=0.003..0.003 rows=1 loops=13)
 |
+------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

Explain: Since adding indexes on both vote and name can improve the performance of both queries, we decide to make our third design as adding both indexes together. As the result of “explain analyze” shows in the terminal, this design does help to improve the performance of our query, especially that there are two index lookups in our optimization steps shown.