Database Design Doc - teamOne

CREATE COMMANDS:

CREATE TABLE Retailer (retailerName VARCHAR(255), retailerAddress VARCHAR(255) DEFAULT NULL, PRIMARY KEY (retailerName));

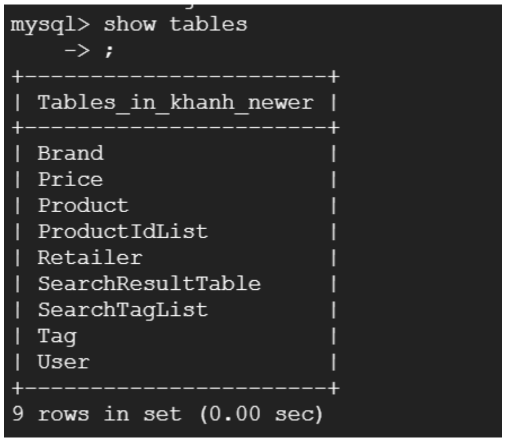
CREATE TABLE Brand (brandName VARCHAR(255), productCount INT DEFAULT 0, PRIMARY KEY (brandName));

CREATE TABLE Product (productId INT, productName VARCHAR(255), productUrl VARCHAR(255), brandName VARCHAR(255), PRIMARY KEY (productId), FOREIGN KEY (brandName) REFERENCES Brand(brandName) ON DELETE SET NULL ON UPDATE CASCADE);

CREATE TABLE Price (productId INT, retailerName VARCHAR(255), price DOUBLE, FOREIGN KEY (productId) REFERENCES Product(productId) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (retailerName) REFERENCES Retailer(retailerName) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE User (username VARCHAR(255), password VARCHAR(255), PRIMARY KEY (username));

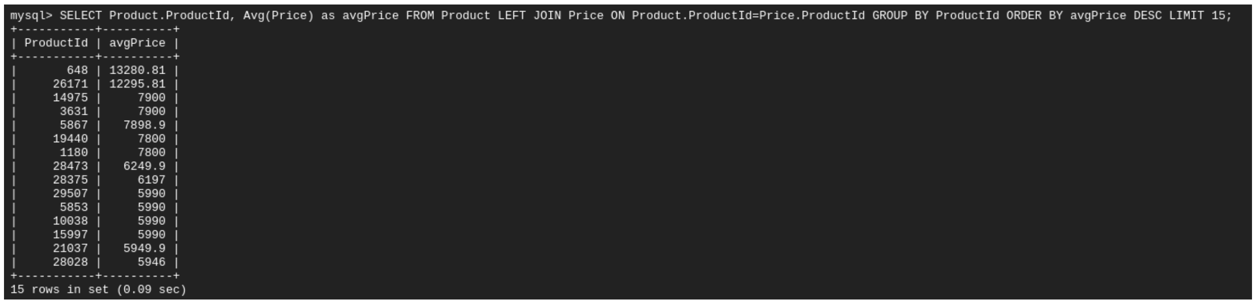
Screenshot of Database**:**



```
mysql> show tables
    -> ;
+-----------------------+
| Tables_in_khanh_newer |
+-----------------------+
| Brand                 |
| Price                 |
| Product               |
| ProductIdList         |
| Retailer              |
| SearchResultTable     |
| SearchTagList         |
| Tag                   |
| User                  |
+-----------------------+
9 rows in set (0.00 sec)
```

SQL Query 1:

SELECT Product.ProductId, Avg(Price) as avgPrice FROM Product LEFT JOIN Price ON Product.ProductId=Price.ProductId GROUP BY ProductId ORDER BY avgPrice DESC LIMIT 15

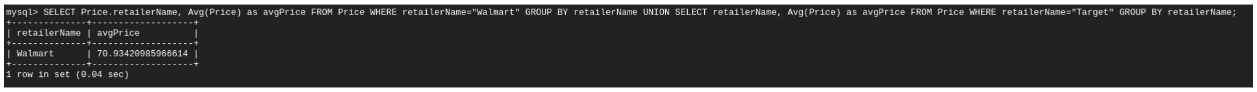


```
mysql> SELECT Product.ProductId, Avg(Price) as avgPrice FROM Product LEFT JOIN Price ON Product.ProductId=Price.ProductId GROUP BY ProductId ORDER BY avgPrice DESC LIMIT 15;
+-----------+----------+
| ProductId | avgPrice |
+-----------+----------+
|       648 | 13280.81 |
|     26171 | 12295.81 |
|     14975 |     7900 |
|      3631 |     7900 |
|      5867 |   7898.9 |
|     19440 |     7800 |
|      1180 |     7800 |
|     28473 |   6249.9 |
|     28375 |     6197 |
|     29507 |     5990 |
|      5853 |     5990 |
|     10038 |     5990 |
|     15997 |     5990 |
|     21037 |   5949.9 |
|     28028 |     5946 |
+-----------+----------+
15 rows in set (0.09 sec)
```

The above query returns the average price of each product in the database over all retailers.

<u>SQL Query 2:</u>

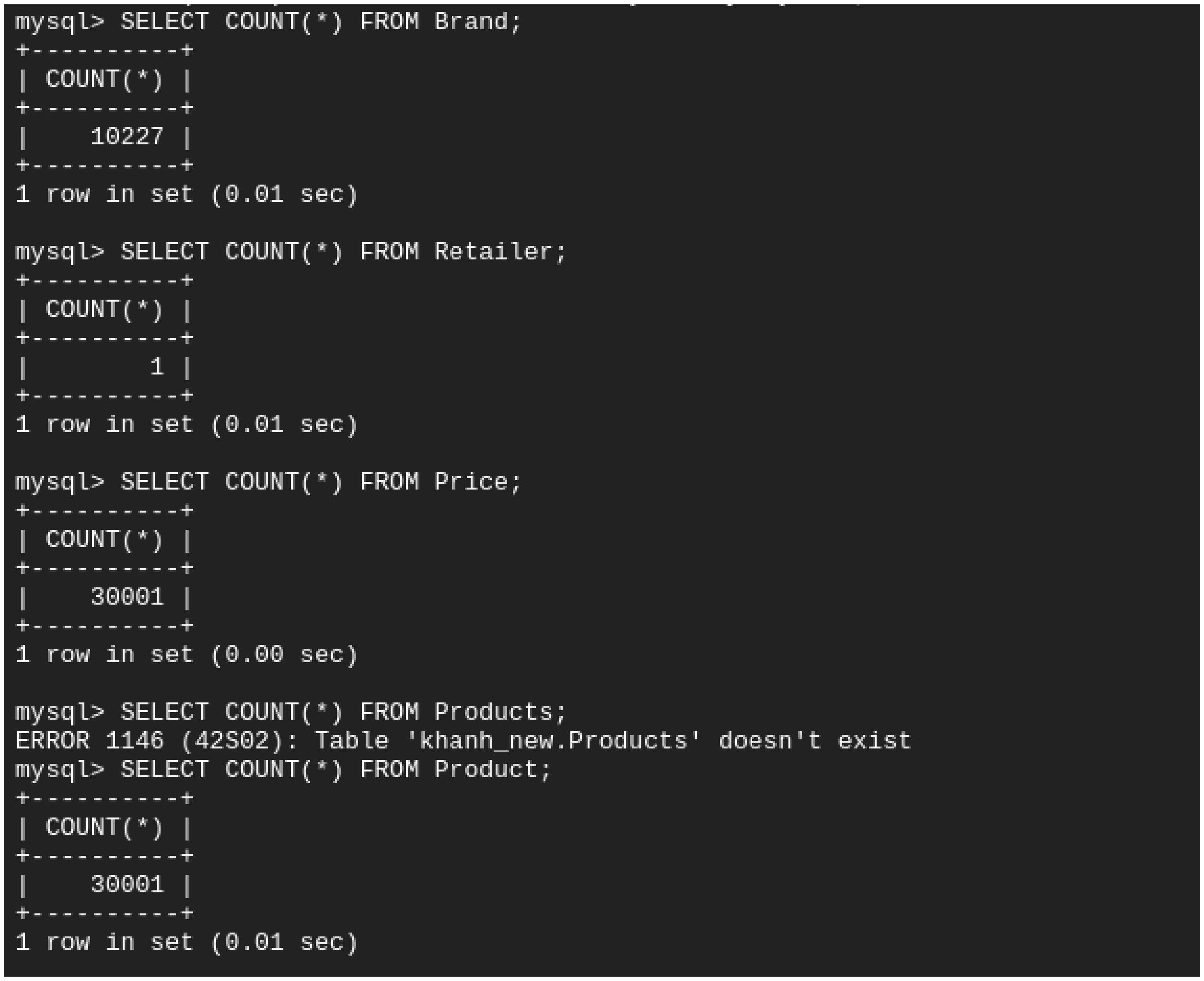
SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Walmart' UNION SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Target'



```
mysql> SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName="Walmart" GROUP BY retailerName UNION SELECT retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName="Target" GROUP BY retailerName;
+--------------+-------------------+
| retailerName | avgPrice          |
+--------------+-------------------+
| Walmart      | 70.93420985966614 |
+--------------+-------------------+
1 row in set (0.04 sec)
```

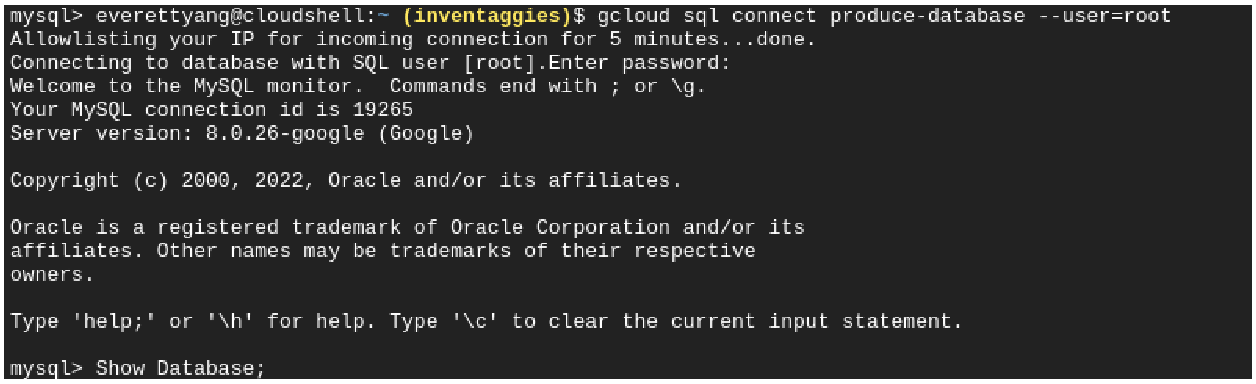
The above query returns the average price of all products in several retailers unioned together. There is only one row since the only data we have is from Walmart at the moment.

<u>Data Count:</u>


```
mysql> SELECT COUNT(*) FROM Brand;
+----------+
| COUNT(*) |
+----------+
|    10227 |
+----------+
1 row in set (0.01 sec)

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

mysql> SELECT COUNT(*) FROM Price;
+----------+
| COUNT(*) |
+----------+
|    30001 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) FROM Products;
ERROR 1146 (42S02): Table 'khanh_new.Products' doesn't exist
mysql> SELECT COUNT(*) FROM Product;
+----------+
| COUNT(*) |
+----------+
|    30001 |
+----------+
1 row in set (0.01 sec)
```

The above shows the row count for each table in our database.

Database Terminal Info on Connection:

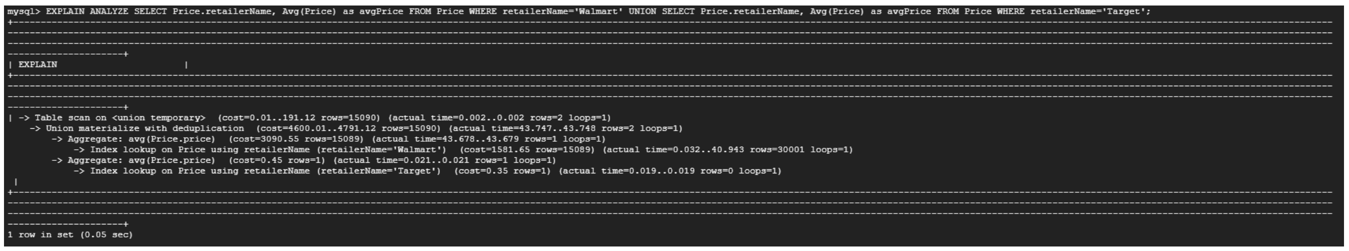

```
mysql> everettyang@cloudshell:~ (inventaggies)$ gcloud sql connect produce-database --user=root
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19265
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> Show Database;
```

The above is the connection to our database server.

Indexing:

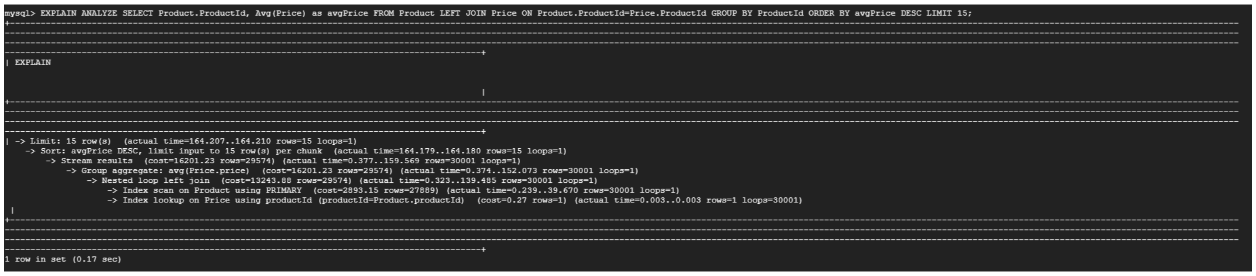
With no index:

SQL Query 1:



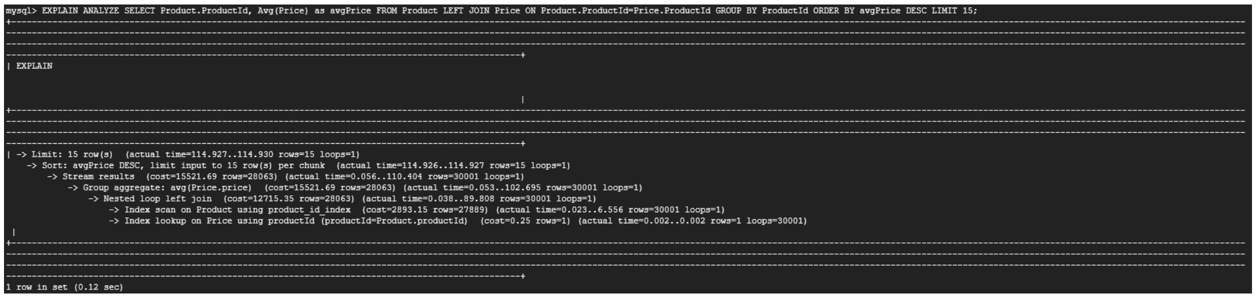
```
mysql> EXPLAIN ANALYZE SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Walmart' UNION SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Target';
+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
---------------------+
| EXPLAIN                      |
+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
---------------------+
| -> Table scan on <union temporary>  (cost=0.01..191.12 rows=15090) (actual time=0.002..0.002 rows=2 loops=1)
    -> Union materialize with deduplication  (cost=4600.01..4791.12 rows=15090) (actual time=43.747..43.748 rows=2 loops=1)
        -> Aggregate: avg(Price.price)  (cost=3090.55 rows=15089) (actual time=43.678..43.679 rows=1 loops=1)
            -> Index lookup on Price using retailerName (retailerName='Walmart')  (cost=1581.65 rows=15089) (actual time=0.032..40.943 rows=30001 loops=1)
        -> Aggregate: avg(Price.price)  (cost=0.45 rows=1) (actual time=0.021..0.021 rows=1 loops=1)
            -> Index lookup on Price using retailerName (retailerName='Target')  (cost=0.35 rows=1) (actual time=0.019..0.019 rows=0 loops=1)
 |
+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
---------------------+
1 row in set (0.05 sec)
```

SQL Query 2:



```
mysql> EXPLAIN ANALYZE SELECT Product.ProductId, Avg(Price) as avgPrice FROM Product LEFT JOIN Price ON Product.ProductId=Price.ProductId GROUP BY ProductId ORDER BY avgPrice DESC LIMIT 15;
+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------+
| EXPLAIN


                                                                                              |
+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=164.207..164.210 rows=15 loops=1)
    -> Sort: avgPrice DESC, limit input to 15 row(s) per chunk  (actual time=164.179..164.180 rows=15 loops=1)
        -> Stream results  (cost=16201.23 rows=29574) (actual time=0.377..159.569 rows=30001 loops=1)
            -> Group aggregate: avg(Price.price)  (cost=16201.23 rows=29574) (actual time=0.374..152.073 rows=30001 loops=1)
                -> Nested loop left join  (cost=13243.88 rows=29574) (actual time=0.323..139.485 rows=30001 loops=1)
                    -> Index scan on Product using PRIMARY  (cost=2893.15 rows=27889) (actual time=0.239..39.670 rows=30001 loops=1)
                    -> Index lookup on Price using productId (productId=Product.productId)  (cost=0.27 rows=1) (actual time=0.003..0.003 rows=1 loops=30001)
 |
+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------+
1 row in set (0.17 sec)
```

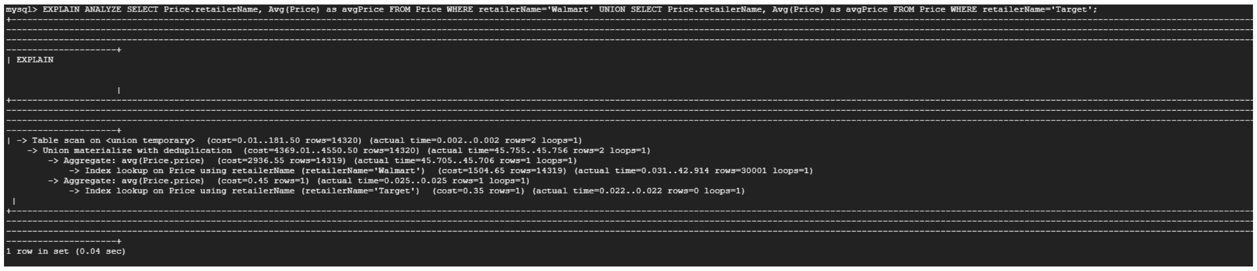
Adding index to Products.productId:
Q1:



```
mysql> EXPLAIN ANALYZE SELECT Product.ProductId, Avg(Price) as avgPrice FROM Product LEFT JOIN Price ON Product.ProductId=Price.ProductId GROUP BY ProductId ORDER BY avgPrice DESC LIMIT 15;
+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------+
| EXPLAIN


                                                                                              |
+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=114.927..114.930 rows=15 loops=1)
    -> Sort: avgPrice DESC, limit input to 15 row(s) per chunk  (actual time=114.926..114.927 rows=15 loops=1)
        -> Stream results  (cost=15521.69 rows=28063) (actual time=0.056..110.404 rows=30001 loops=1)
            -> Group aggregate: avg(Price.price)  (cost=15521.69 rows=28063) (actual time=0.053..102.695 rows=30001 loops=1)
                -> Nested loop left join  (cost=12715.35 rows=28063) (actual time=0.038..89.808 rows=30001 loops=1)
                    -> Index scan on Product using product_id_index  (cost=2893.15 rows=27889) (actual time=0.023..6.556 rows=30001 loops=1)
                    -> Index lookup on Price using productId (productId=Product.productId)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=30001)
 |
+------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------+
1 row in set (0.12 sec)
```

Putting an index on Products.productId makes the query faster than with no index

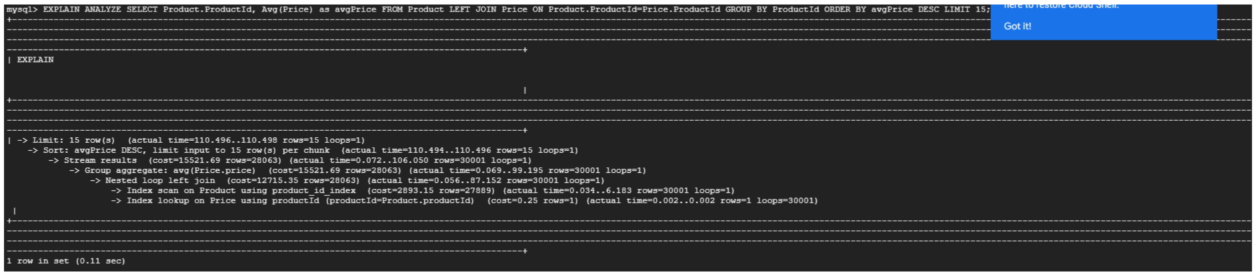
Q2:



```
mysql> EXPLAIN ANALYZE SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Walmart' UNION SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Target';
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------+
| EXPLAIN


                    |
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------+
| -> Table scan on <union temporary>  (cost=0.01..181.50 rows=14320) (actual time=0.002..0.002 rows=2 loops=1)
    -> Union materialize with deduplication  (cost=4369.01..4550.50 rows=14320) (actual time=45.755..45.756 rows=2 loops=1)
        -> Aggregate: avg(Price.price)  (cost=2936.55 rows=14319) (actual time=45.705..45.706 rows=1 loops=1)
            -> Index lookup on Price using retailerName (retailerName='Walmart')  (cost=1504.65 rows=14319) (actual time=0.031..42.914 rows=30001 loops=1)
        -> Aggregate: avg(Price.price)  (cost=0.45 rows=1) (actual time=0.025..0.025 rows=1 loops=1)
            -> Index lookup on Price using retailerName (retailerName='Target')  (cost=0.35 rows=1) (actual time=0.022..0.022 rows=0 loops=1)
 |
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------+
1 row in set (0.04 sec)
```

Adding index to Price.retailerName:

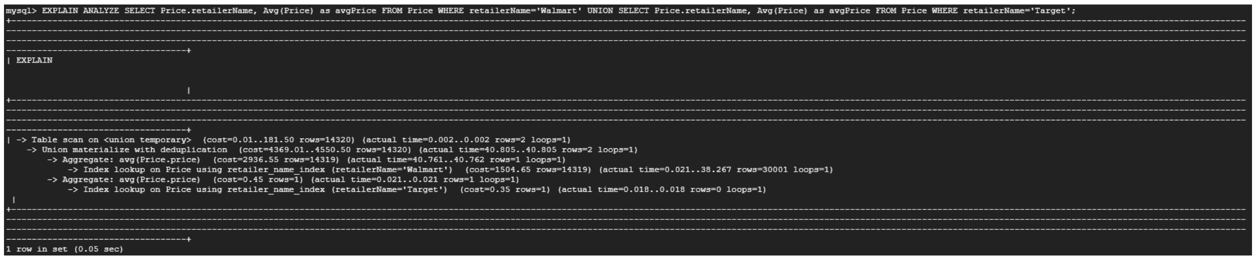
Q1:



```
mysql> EXPLAIN ANALYZE SELECT Product.ProductId, Avg(Price) as avgPrice FROM Product LEFT JOIN Price ON Product.ProductId=Price.ProductId GROUP BY ProductId ORDER BY avgPrice DESC LIMIT 15;
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------+
| EXPLAIN


                                                                                           |
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=110.496..110.498 rows=15 loops=1)
    -> Sort: avgPrice DESC, limit input to 15 row(s) per chunk  (actual time=110.494..110.496 rows=15 loops=1)
        -> Stream results  (cost=15521.69 rows=28063) (actual time=0.072..106.050 rows=30001 loops=1)
            -> Group aggregate: avg(Price.price)  (cost=15521.69 rows=28063) (actual time=0.069..99.195 rows=30001 loops=1)
                -> Nested loop left join  (cost=12715.35 rows=28063) (actual time=0.056..87.152 rows=30001 loops=1)
                    -> Index scan on Product using product_id_index  (cost=2893.15 rows=27889) (actual time=0.034..6.183 rows=30001 loops=1)
                    -> Index lookup on Price using productId (productId=Product.productId)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=30001)
 |
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------+
1 row in set (0.11 sec)
```

Putting an index on Price.retailerName makes the query faster than with no index but not as fast as an index on Products.productId. This is because we are aggregating over productId, so having an index on productId has a much greater impact on overall runtime.

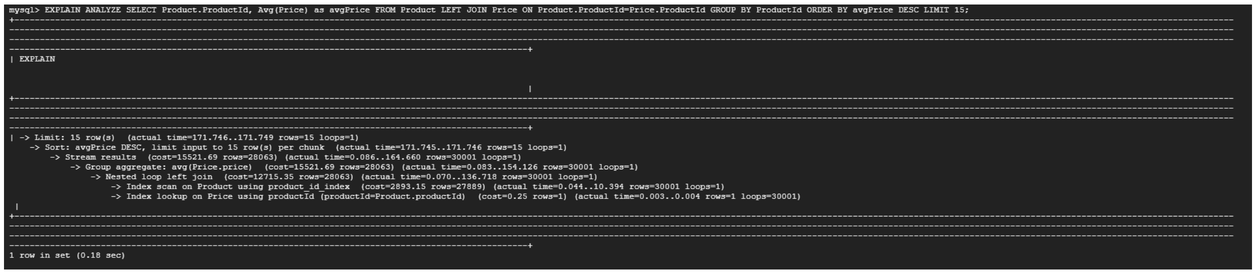
Q2:



```
mysql> EXPLAIN ANALYZE SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Walmart' UNION SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Target';
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-----------------------------------+
| EXPLAIN


                                   |
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-----------------------------------+
| -> Table scan on <union temporary>  (cost=0.01..181.50 rows=14320) (actual time=0.002..0.002 rows=2 loops=1)
    -> Union materialize with deduplication  (cost=4369.01..4550.50 rows=14320) (actual time=40.805..40.805 rows=2 loops=1)
        -> Aggregate: avg(Price.price)  (cost=2936.55 rows=14319) (actual time=40.761..40.762 rows=1 loops=1)
            -> Index lookup on Price using retailer_name_index (retailerName='Walmart')  (cost=1504.65 rows=14319) (actual time=0.021..38.267 rows=30001 loops=1)
        -> Aggregate: avg(Price.price)  (cost=0.45 rows=1) (actual time=0.021..0.021 rows=1 loops=1)
            -> Index lookup on Price using retailer_name_index (retailerName='Target')  (cost=0.35 rows=1) (actual time=0.018..0.018 rows=0 loops=1)
 |
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-----------------------------------+
1 row in set (0.05 sec)
```

Adding index to Price.price:

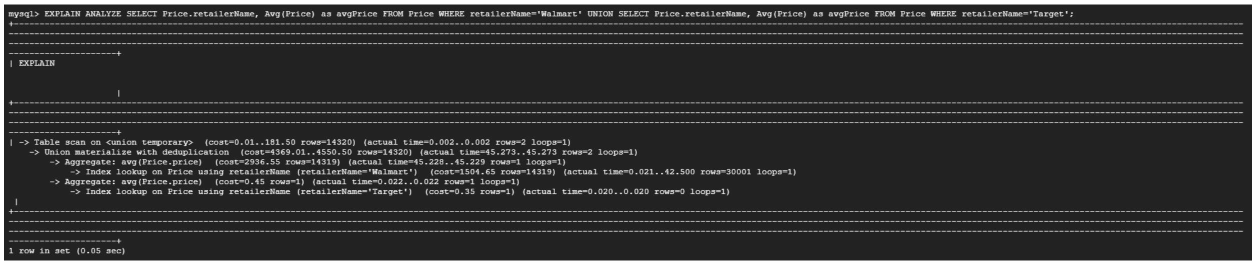
Q1:



```
mysql> EXPLAIN ANALYZE SELECT Product.ProductId, Avg(Price) as avgPrice FROM Product LEFT JOIN Price ON Product.ProductId=Price.ProductId GROUP BY ProductId ORDER BY avgPrice DESC LIMIT 15;
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------+
| EXPLAIN


                                                                                           |
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------+
| -> Limit: 15 row(s)  (actual time=171.746..171.749 rows=15 loops=1)
    -> Sort: avgPrice DESC, limit input to 15 row(s) per chunk  (actual time=171.745..171.746 rows=15 loops=1)
        -> Stream results  (cost=15521.69 rows=28063) (actual time=0.086..164.660 rows=30001 loops=1)
            -> Group aggregate: avg(Price.price)  (cost=15521.69 rows=28063) (actual time=0.083..154.126 rows=30001 loops=1)
                -> Nested loop left join  (cost=12715.35 rows=28063) (actual time=0.070..136.718 rows=30001 loops=1)
                    -> Index scan on Product using product_id_index  (cost=2893.15 rows=27889) (actual time=0.044..10.394 rows=30001 loops=1)
                    -> Index lookup on Price using productId (productId=Product.productId)  (cost=0.25 rows=1) (actual time=0.003..0.004 rows=1 loops=30001)
 |
+--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------+
1 row in set (0.18 sec)
```

Adding an index on Price.price is much better than having no index but performs the worst out of the three indexing schemes. This is because we are not aggregating over price.

Q2:



```
mysql> EXPLAIN ANALYZE SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Walmart' UNION SELECT Price.retailerName, Avg(Price) as avgPrice FROM Price WHERE retailerName='Target';
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------+
| EXPLAIN


                      |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------+
| -> Table scan on <union temporary>  (cost=0.01..181.50 rows=14320) (actual time=0.002..0.002 rows=2 loops=1)
    -> Union materialize with deduplication  (cost=4369.01..4550.50 rows=14320) (actual time=45.273..45.273 rows=2 loops=1)
        -> Aggregate: avg(Price.price)  (cost=2936.55 rows=14319) (actual time=45.228..45.229 rows=1 loops=1)
            -> Index lookup on Price using retailerName (retailerName='Walmart')  (cost=1504.65 rows=14319) (actual time=0.021..42.500 rows=30001 loops=1)
        -> Aggregate: avg(Price.price)  (cost=0.45 rows=1) (actual time=0.022..0.022 rows=1 loops=1)
            -> Index lookup on Price using retailerName (retailerName='Target')  (cost=0.35 rows=1) (actual time=0.020..0.020 rows=0 loops=1)
 |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------+
1 row in set (0.05 sec)
```

As seen above, each of the three index designs yields slight improvements in SELECT speed and aggregation speed due to the index providing faster retrieval time for the relevant productId and price columns.