Database Design Doc - teamOne

CREATE COMMANDS:

```
CREATE TABLE Retailer (
        retailerName VARCHAR(255),
        PRIMARY KEY (retailerName)
);
CREATE TABLE Brand (
        brandName VARCHAR(255),
        PRIMARY KEY (brandName)
);
CREATE Table Product (
        productId INT AUTO_INCREMENT,
        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,
        username VARCHAR(255) DEFAULT "BOT",
        PRIMARY KEY (productId, retailerName),
        FOREIGN KEY (productId)
        REFERENCES Product(productId)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (retailerName)
        REFERENCES Retailer(retailerName)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (username)
        REFERENCES User(username)
        ON DELETE SET "BOT"
        ON UPDATE CASCADE
```

```
);
CREATE TABLE Tag (
        productId INT,
        Tag VARCHAR(255),
        PRIMARY KEY (productId, Tag),
        FOREIGN KEY (productId)
        REFERENCES Product(productId)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
CREATE TABLE User (
        username VARCHAR(255),
        password VARCHAR(255) BINARY,
        PRIMARY KEY (username)
);
```

Screenshot of Database:

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 *
FROM Product NATURAL JOIN Price NATURAL JOIN
(SELECT productId, GROUP_CONCAT(Tag SEPARATOR ', ') AS tagList
FROM Tag GROUP BY productId) AS TEMP
WHERE productName LIKE "%apple%" limit 15;

```
+-----------+------------------------------------------------------------------------------------------------------------------------------+----------------------------------------------------
------------------------------------------------------------------------------------------------------+--------------------------------+-------------+-------+----------+----------------------
-----------------------------------------------------------+
|       155 | Dapple Baby Plant Based Calming Shampoo & Body Wash, Lavender & Jasmine, 16.9 oz.                                            | https://www.walmart.com/ip/Dapple-Baby-Plant-Based-Cal
ming-Shampoo-Body-Wash-Lavender-Jasmine-16-9-oz/241939104                                             | dapple                         | Walmart     |  7.48 | BOT      | Baby, Baby Care, Diaper
ing                                                        |
|       333 | Celestial Seasonings Tea Caffeine Free Herbal Tea, Cinnamon Apple Spice 20 ea                                                | https://www.walmart.com/ip/Celestial-Seasonings-Tea-Ca
ffeine-Free-Herbal-Tea-Cinnamon-Apple-Spice-20-ea/791767810                                           | Celestial Seasonings           | Walmart     |  2.37 | BOT      | Beverages, Celestial Se
asonings Tea, Food, Tea                                    |
|       427 | Big Apple Mints                                                                                                              | https://www.walmart.com/ip/Big-Apple-Mints/119044870
                                                                                                      | Zizo USA                       | Walmart     |  8.44 | BOT      | Chocolate, Candy & Gum,
 Food, Mints, Shop All Mints                               |
|       613 | Febreze Plug Air Freshener, Scented Oil Refill, Fresh-Pressed Apple, 2 count                                                 | https://www.walmart.com/ip/Febreze-Plug-Air-Freshener-
Scented-Oil-Refill-Fresh-Pressed-Apple-2-count/51446812                                               | Febreze                        | Walmart     |  9.47 | BOT      | Air Fresheners, Febreze
 Plug In Air Fresheners Scented Oils, Household Essentials |
|       785 | McCanns Apples & Cinnamon Instant Irish Oatmeal, 12.3 oz (Pack of 12)                                                        | https://www.walmart.com/ip/McCann-s-Apples-Cinnamon-In
stant-Irish-Oatmeal-12-3-oz-Pack-of-12/17196461                                                       |                                | Walmart     | 61.33 | BOT      | Breakfast & Cereal, Foo
d, Instant Oatmeal, Oatmeal & Grits                        |
|      1009 | Bobs Red Mill, Granola, Apple Blueberry, 12 oz (pack of 4)                                                                   | https://www.walmart.com/ip/Bob-s-Red-Mill-Granola-Appl
e-Blueberry-12-oz-pack-of-4/831861224                                                                 | Bobs Red Mill                  | Walmart     | 23.68 | BOT      | Breakfast & Cereal, Foo
d, Granola & Muesli                                        |
|      1011 | Jarrow Formulas, Organic Apple Cider Vinegar, 16 fl oz (pack of 6)                                                           | https://www.walmart.com/ip/Jarrow-Formulas-Organic-App
le-Cider-Vinegar-16-fl-oz-pack-of-6/917510963                                                         | Jarrow Formulas                | Walmart     | 59.17 | BOT      | Condiments, Condiments,
 Sauces & Spices, Food                                     |
|      1305 | Dynamic Health - Organic Apple Cider Vinegar with the Mother and Natural Honey Glass - 32 oz.                                | https://www.walmart.com/ip/Dynamic-Health-Organic-Appl
e-Cider-Vinegar-with-the-Mother-and-Natural-Honey-Glass-32-oz/108027555                               | Dynamic Health                 | Walmart     | 11.64 | BOT      | Condiments, Condiments,
 Sauces & Spices, Food                                    |
|      1390 | Dynamic Health Laboratories Inc Organic Apple Cider Vinegar w/Mother 16 Ounce, Pack of 2                                     | https://www.walmart.com/ip/Dynamic-Health-Laboratories
-Inc-Organic-Apple-Cider-Vinegar-w-Mother-16-Ounce-Pack-of-2/623176027                                | Dynamic Health Laboratories Inc | Walmart    | 31.28 | BOT      | Condiments, Condiments,
 Sauces & Spices, Food                                    |
|      1616 | Gain Apple Mango Tango HE, Liquid Laundry Detergent, 150 Fl Oz 96 loads                                                      | https://www.walmart.com/ip/Gain-Apple-Mango-Tango-HE-L
iquid-Laundry-Detergent-150-Fl-Oz-96-loads/21899543                                                   | Gain                           | Walmart     | 13.44 | BOT      | Household Essentials, L
aundry, Laundry Detergents                                |
|      1676 | Apple Cider Vinegar and Honey Tonic - 16 fl. oz (473 ml) by Fire Cider                                                       | https://www.walmart.com/ip/Apple-Cider-Vinegar-and-Hon
ey-Tonic-16-fl-oz-473-ml-by-Fire-Cider/196683152                                                      | Fire Cider                     | Walmart     | 23.39 | BOT      | Health, Natural Alterna
tives, Superfoods, Superfoods & Cleanses                  |
|      1852 | Leather Magnetic Smart Case Cover Stand for Apple iPad Mini 5 5th (A2133, A2124, A2126) Eat Sleep Lacrosse Repeat (Green) | https://www.walmart.com/ip/Leather-Magnetic-Smart-Case
-Cover-Stand-for-Apple-iPad-Mini-5-5th-A2133-A2124-A2126-Eat-Sleep-Lacrosse-Repeat-Green/201219538 | MIP                            | Walmart     | 24.99 | BOT      | Lacrosse, Lacrosse Goal
```

```
|      1676 | Apple Cider Vinegar and Honey Tonic - 16 fl. oz (473 ml) by Fire Cider                                                       | https://www.walmart.com/ip/Apple-Cider-Vinegar-and-Hon
ey-Tonic-16-fl-oz-473-ml-by-Fire-Cider/196683152                                                      | Fire Cider                     | Walmart     | 23.39 | BOT      | Health, Natural Alterna
tives, Superfoods, Superfoods & Cleanses                  |
|      1852 | Leather Magnetic Smart Case Cover Stand for Apple iPad Mini 5 5th (A2133, A2124, A2126) Eat Sleep Lacrosse Repeat (Green) | https://www.walmart.com/ip/Leather-Magnetic-Smart-Case
-Cover-Stand-for-Apple-iPad-Mini-5-5th-A2133-A2124-A2126-Eat-Sleep-Lacrosse-Repeat-Green/201219538 | MIP                            | Walmart     | 24.99 | BOT      | Lacrosse, Lacrosse Goal
s, Sports, Sports & Outdoors                              |
|      2019 | Febreze Small Spaces Air Freshener, Fresh-Pressed Apple, 1 count                                                             | https://www.walmart.com/ip/Febreze-Small-Spaces-Air-Fr
eshener-Fresh-Pressed-Apple-1-count/761228842                                                         | Febreze                        | Walmart     |  2.97 | BOT      | Air Fresheners, Febreze
 Small Spaces Air Fresheners, Household Essentials        |
|      2063 | Leather Magnetic Smart Case Cover Stand for Apple iPad Eat Sleep Lacrosse Repeat (for iPad Mini 4, White)                    | https://www.walmart.com/ip/Leather-Magnetic-Smart-Case
-Cover-Stand-for-Apple-iPad-Eat-Sleep-Lacrosse-Repeat-for-iPad-Mini-4-White/659794548              | MIP                            | Walmart     | 19.98 | BOT      | Lacrosse, Lacrosse Goal
s, Sports, Sports & Outdoors                              |
|      2243 | 6 Pack - Air Wick Scented Oil - Refill Apple Cinnamon Medley 5 ct.                                                           | https://www.walmart.com/ip/6-Pack-Air-Wick-Scented-Oil
-Refill-Apple-Cinnamon-Medley-5-ct/615027764                                                          | Air Wick                       | Walmart     | 56.71 | BOT      | Air Freshener & Deodori
zer Refills, Bathroom, Household Essentials               |
+-----------+------------------------------------------------------------------------------------------------------------------------------+----------------------------------------------------
------------------------------------------------------------------------------------------------------+--------------------------------+-------------+-------+----------+----------------------
-----------------------------------------------------------+
15 rows in set (0.31 sec)
```

The above query returns the search results for products with “apple” in its name along with the tag associated with the product as tagList and other information that we would then display on our application.

<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:

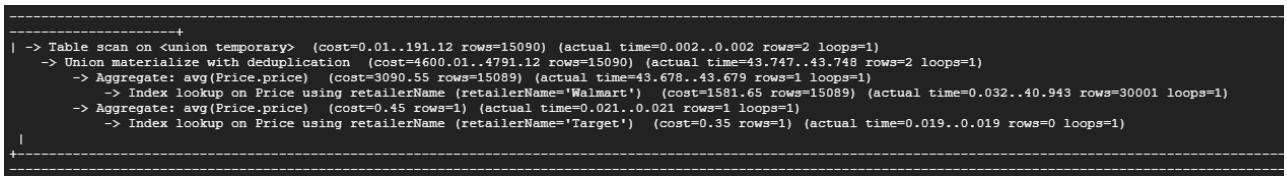
```
mysql> everettyang@cloudshell:~ (inventaggies)$ gcloud sql connect produce-database --user=root
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection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

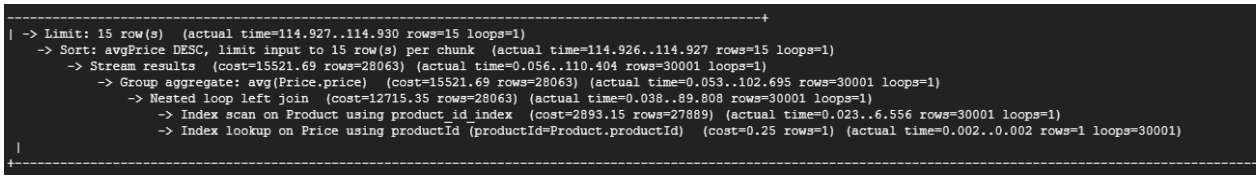
Indexing:

QUERY 1:

With no index query 1:

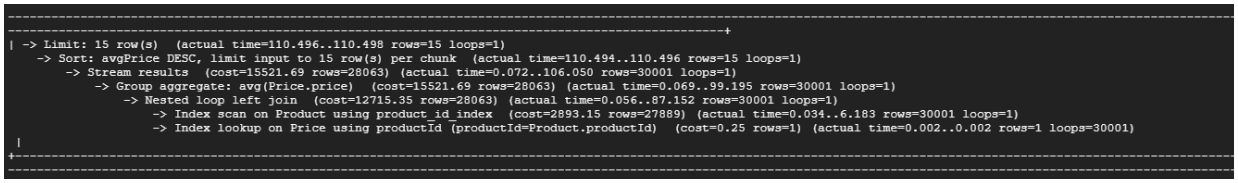


Adding index to Products.productId for query 1:



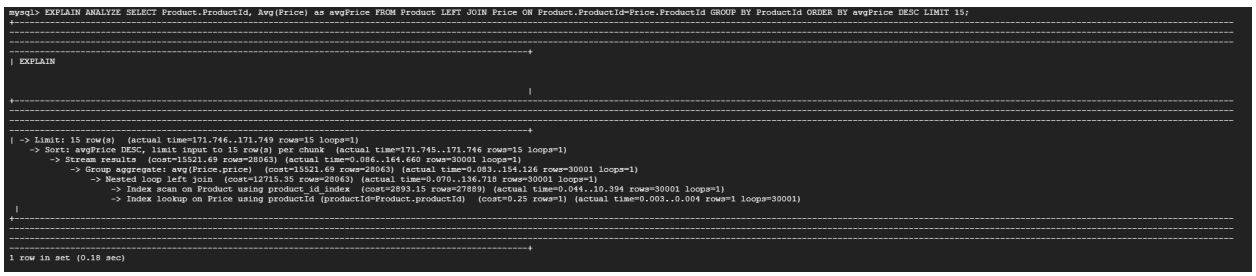
Putting an index on Products.productId makes the query faster than with no index. This is because we are grouping by productId, which means having an index for productId makes finding and matching productId faster, thus improving run time.

Adding index to Price.retailerName for query 1:



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.

Adding index to Price.price for query 1:



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.

QUERY 2

With no index query 2:

```
-> Limit: 10 row(s)  (cost=40679592.00 rows=10) (actual time=102.271..104.144 rows=10 loops=1)
    -> Nested loop inner join  (cost=40679592.00 rows=406647368) (actual time=102.268..104.141 rows=10 loops=1)
        -> Nested loop inner join  (cost=6608.16 rows=3299) (actual time=0.426..2.258 rows=10 loops=1)
            -> Filter: (Product.productName like '%apple%')  (cost=3073.45 rows=3299) (actual time=0.330..2.103 rows=10 loops=1)
                -> Table scan on Product  (cost=3073.45 rows=29692) (actual time=0.131..0.896 rows=1612 loops=1)
            -> Index lookup on Price using PRIMARY (productId=Product.productId)  (cost=0.97 rows=1) (actual time=0.014..0.015 rows=1 loops=10)
        -> Index lookup on TEMP using <auto_key0> (productId=Product.productId)  (actual time=0.005..0.005 rows=1 loops=10)
            -> Materialize  (cost=37069.85..37069.85 rows=123272) (actual time=101.861..101.864 rows=29979 loops=1)
                -> Group aggregate: group_concat(Tag.Tag separator ', ')  (cost=24742.65 rows=123272) (actual time=0.066..70.929 rows=29979 loops=1)
                    -> Index scan on Tag using PRIMARY  (cost=12415.45 rows=123272) (actual time=0.042..36.410 rows=113078 loops=1)
```

Adding index to Tag.productId for query 2:

```
-> Limit: 10 row(s)  (cost=40679592.00 rows=10) (actual time=225.054..226.968 rows=10 loops=1)
    -> Nested loop inner join  (cost=40679592.00 rows=406647368) (actual time=225.053..226.965 rows=10 loops=1)
        -> Nested loop inner join  (cost=6608.16 rows=3299) (actual time=0.267..2.139 rows=10 loops=1)
            -> Filter: (Product.productName like '%apple%')  (cost=3073.45 rows=3299) (actual time=0.249..2.061 rows=10 loops=1)
                -> Table scan on Product  (cost=3073.45 rows=29692) (actual time=0.068..0.855 rows=1612 loops=1)
            -> Index lookup on Price using PRIMARY (productId=Product.productId)  (cost=0.97 rows=1) (actual time=0.006..0.007 rows=1 loops=10)
        -> Index lookup on TEMP using <auto_key0> (productId=Product.productId)  (actual time=0.005..0.005 rows=1 loops=10)
            -> Materialize  (cost=37069.85..37069.85 rows=123272) (actual time=224.816..224.819 rows=29979 loops=1)
                -> Group aggregate: group_concat(Tag.Tag separator ', ')  (cost=24742.65 rows=123272) (actual time=2.372..191.069 rows=29979 loops=1)
                    -> Index scan on Tag using idx_productId_tag  (cost=12415.45 rows=123272) (actual time=2.341..153.875 rows=113078 loops=1)
```

We can see that after adding index for productId to Tag, the time of table scan on Product is improved. This is because the subquery in the query group by productId, so index on productId will help speed up the process.

Adding index to Price.price for query 2:

```
-> Limit: 10 row(s)  (cost=40679592.00 rows=10) (actual time=108.668..110.952 rows=10 loops=1)
    -> Nested loop inner join  (cost=40679592.00 rows=406647368) (actual time=108.667..110.948 rows=10 loops=1)
        -> Nested loop inner join  (cost=6608.16 rows=3299) (actual time=0.261..2.474 rows=10 loops=1)
            -> Filter: (Product.productName like '%apple%')  (cost=3073.45 rows=3299) (actual time=0.242..2.360 rows=10 loops=1)
                -> Table scan on Product  (cost=3073.45 rows=29692) (actual time=0.070..0.929 rows=1612 loops=1)
            -> Index lookup on Price using PRIMARY (productId=Product.productId)  (cost=0.97 rows=1) (actual time=0.009..0.011 rows=1 loops=10)
        -> Index lookup on TEMP using <auto_key0> (productId=Product.productId)  (actual time=0.007..0.007 rows=1 loops=10)
            -> Materialize  (cost=37069.85..37069.85 rows=123272) (actual time=108.459..108.462 rows=29979 loops=1)
                -> Group aggregate: group_concat(Tag.Tag separator ', ')  (cost=24742.65 rows=123272) (actual time=0.049..75.171 rows=29979 loops=1)
                    -> Index scan on Tag using PRIMARY  (cost=12415.45 rows=123272) (actual time=0.036..37.946 rows=113078 loops=1)
```

We can see that the speed increased slightly compared to having no index. However it is slower than having an index on productId. This is because the subquery group by productId and the join also join on productId. Thus having an index on productId will make a bigger difference compared to price.

Adding index to Product.productName for query 2:

```
-> Limit: 10 row(s)  (cost=40679592.00 rows=10) (actual time=109.579..111.596 rows=10 loops=1)
    -> Nested loop inner join  (cost=40679592.00 rows=406647368) (actual time=109.577..111.593 rows=10 loops=1)
        -> Nested loop inner join  (cost=6608.16 rows=3299) (actual time=0.867..2.831 rows=10 loops=1)
            -> Filter: (Product.productName like '%apple%')  (cost=3073.45 rows=3299) (actual time=0.847..2.740 rows=10 loops=1)
                -> Table scan on Product  (cost=3073.45 rows=29692) (actual time=0.665..1.514 rows=1612 loops=1)
            -> Index lookup on Price using PRIMARY (productId=Product.productId)  (cost=0.97 rows=1) (actual time=0.007..0.008 rows=1 loops=10)
        -> Index lookup on TEMP using <auto_key0> (productId=Product.productId)  (actual time=0.006..0.006 rows=1 loops=10)
            -> Materialize  (cost=37069.85..37069.85 rows=123272) (actual time=108.748..108.751 rows=29979 loops=1)
                -> Group aggregate: group_concat(Tag.Tag separator ', ')  (cost=24742.65 rows=123272) (actual time=0.057..74.660 rows=29979 loops=1)
                    -> Index scan on Tag using PRIMARY  (cost=12415.45 rows=123272) (actual time=0.042..38.461 rows=113078 loops=1)
```

Having index on productName proved to be unbeneficial even though we have a condition to match productName LIKE “%apple%”. This might be because the LIKE function doesn’t benefit from an index and the index might even take more time to read than actually reading from the table itself.

In conclusion, 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 variable. However, index on productId proved to be the fastest design. This is because our database is built around the unique productId to join all the tables as well as performing group by.