

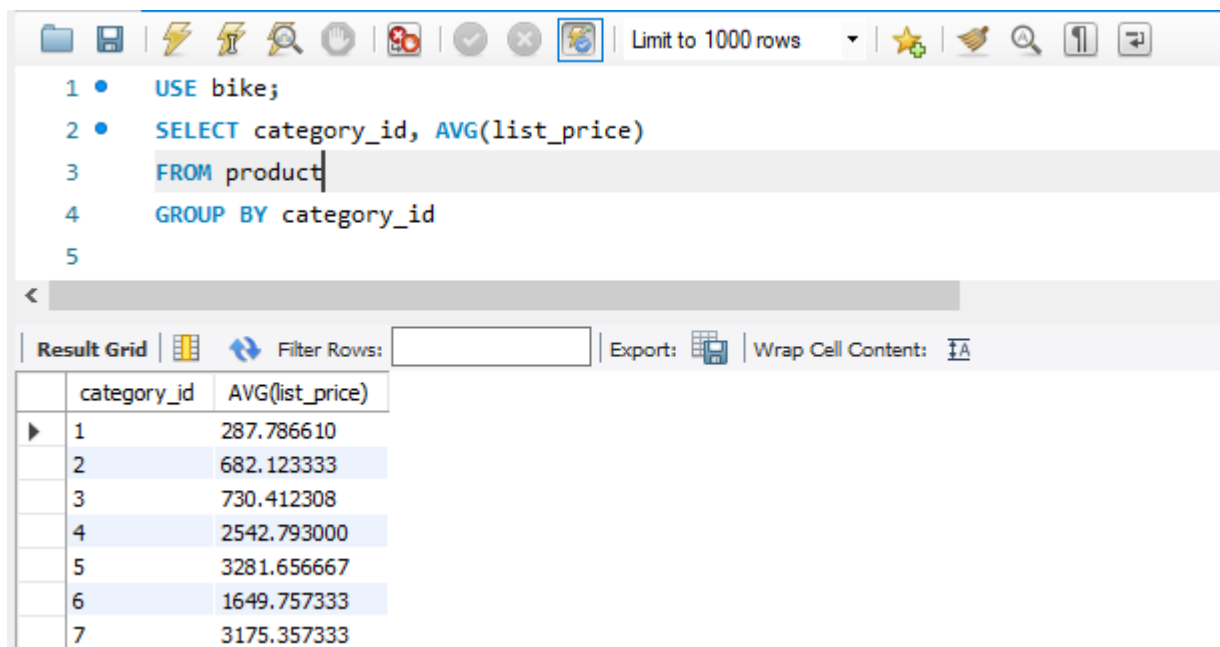
5.3

Simple GROUP BY Query

Code Example:

```
USE bike;
SELECT category_id, AVG(list_price)
FROM product
GROUP BY category_id
```

Results:



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and settings. The query editor contains the following SQL code:

```
1 • USE bike;
2 • SELECT category_id, AVG(list_price)
3   FROM product
4   GROUP BY category_id
5
```

Below the query editor is a 'Result Grid' tab. It shows a table with two columns: 'category_id' and 'AVG(list_price)'. The table contains 7 rows of data. The 'Filter Rows' field is empty, and the 'Export' button is visible. The 'Wrap Cell Content' option is also present.

category_id	AVG(list_price)
1	287.786610
2	682.123333
3	730.412308
4	2542.793000
5	3281.656667
6	1649.757333
7	3175.357333

USE bike:

- Set the bike database to be the default

SELECT category_id, AVG(list_price):

- Select the category_id from the base table
- Calculate the Average of the list price for all rows in the table

FROM product:

- Product is the base table from which data will be returned

GROUP BY category_id:

- Instead of returning a single value that is the average of all list_price items in the product table, return an average list_price for each category
- Without the **GROUP BY** clause, we see from our first example only a single row is returned with an average list_price of 1520.591402.
- With the **GROUP BY** clause, we return an average for each category_id.



This content is provided to you freely by BYU-I Books.

Access it online or download it at https://books.byui.edu/learning_mysql/simple_group_by_quer.