

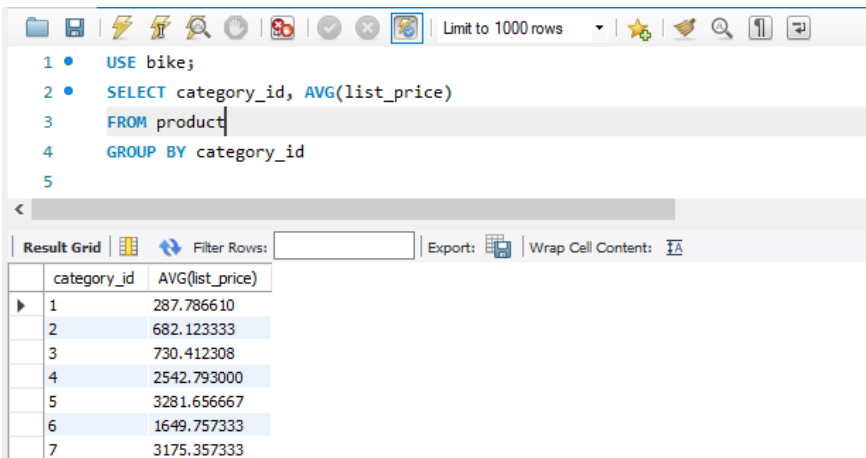
5.3

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, search, and execution. The main area displays the following SQL query:

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

Below the query editor, there is a "Result Grid" section with a "Filter Rows" input field and an "Export" button. The results are displayed in a table with 7 rows and 2 columns: "category_id" and "AVG(list_price)".

	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):

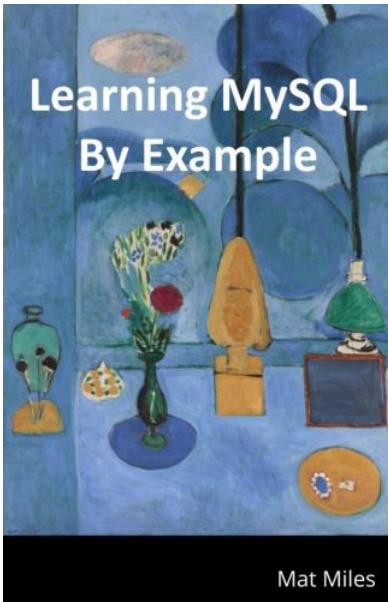
- Select the category_id from the base table
- Calculate the Average of the list price for all rows in the table

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.



Miles, M. (2021). *Learning MySQL By Example*. EdTech Books. https://edtechbooks.org/learning_mysql