**Aggregations with your Data**

The below transformations were performed using the Microsoft Azure platform using Ambari Views, specifically Hive View 2.0. Before any transformations or aggregations could be completed an HDInsight’s Cluster was created and the Video games sales data was loaded into a table called “sales”. The below aggregations take into consideration that the previously mentioned steps have already been completed.

**Step 1 (Create two tables (one for sports games and one for shooter games) displaying the average critic score):**

Use the following code to create the table for sports games average critic score:

CREATE TABLE sports\_avg\_critic\_score  
(  
Name string,

Genre string,  
Critic\_Score int   
);

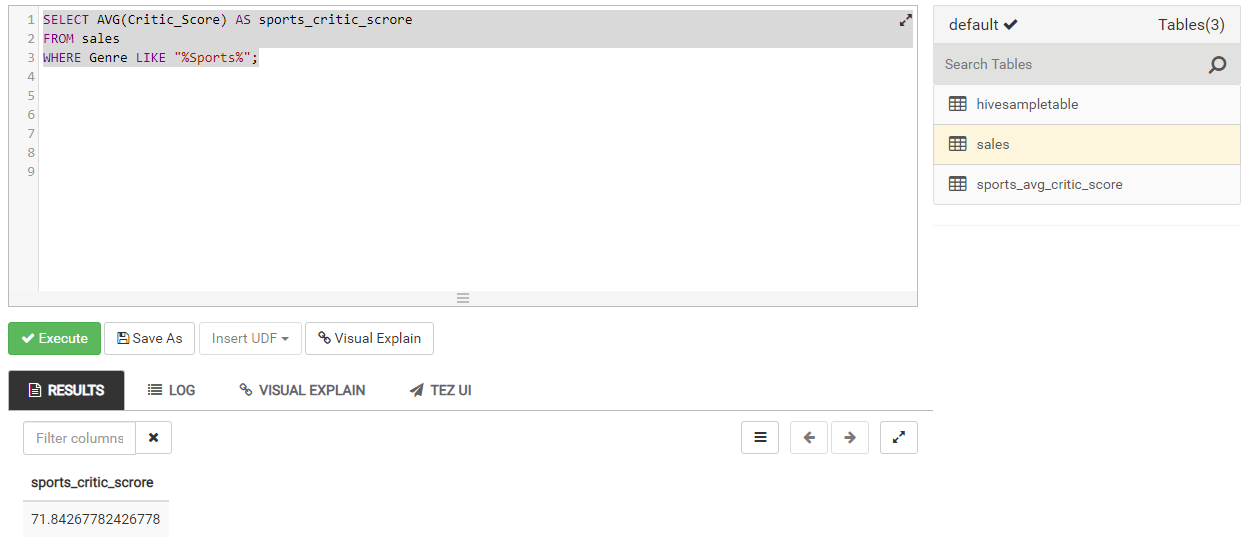
INSERT OVERWRITE TABLE sports\_avg\_critic\_score  
SELECT Name, Genre, Critic\_Score  
FROM sales;

Use the following code to ensure your aggregation has been performed correctly:

SELECT AVG(Critic\_Score) AS sports\_critic\_scrore

FROM sales

WHERE Genre LIKE "%Sports%";



You should see a table that looks like the above.

Use the following code to create the table for shooter games average critic score:

CREATE TABLE shooter\_avg\_critic\_score  
(  
Name string,

Genre string,  
Critic\_Score int   
);

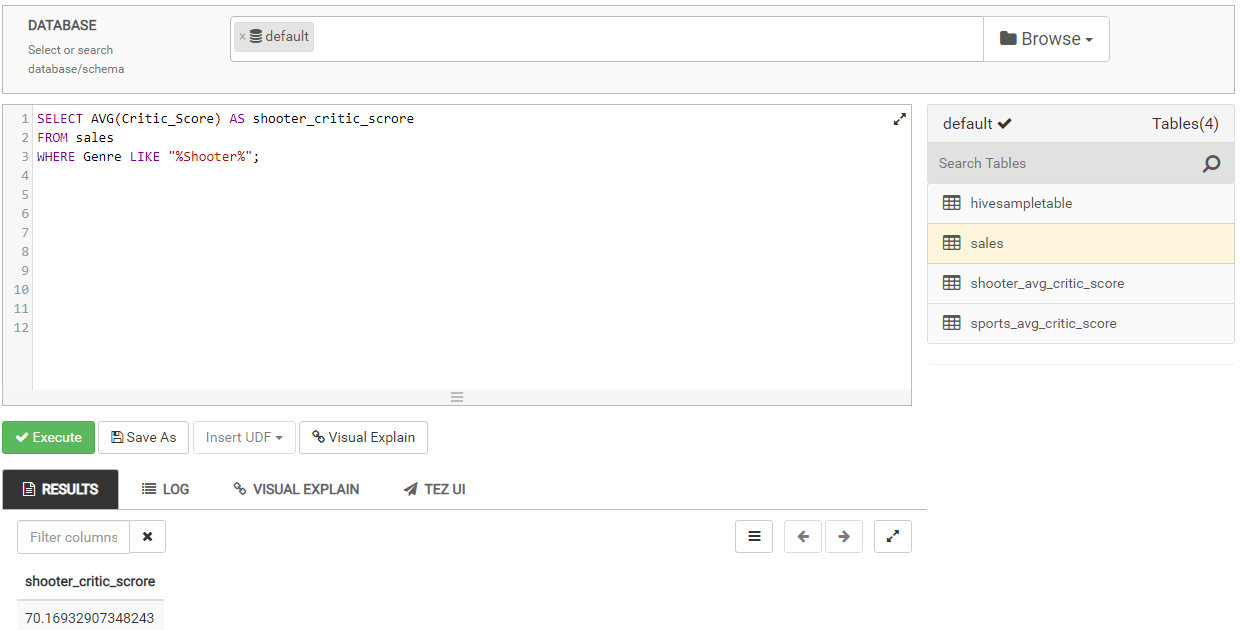
INSERT OVERWRITE TABLE shooter\_avg\_critic\_score  
SELECT Name, Genre, Critic\_Score  
FROM sales;

Use the following code to ensure your aggregation has been performed correctly:

SELECT AVG(Critic\_Score) AS shooter\_critic\_scrore

FROM sales

WHERE Genre LIKE "%Shooter%";



You should see a table that looks like the above.

**Step 2 (Create three statistics tables for the global sales; one for all games, one for sports games and one for shooter games):**

Use the following code to create the table for all games:

CREATE TABLE all\_games\_global\_sales  
(  
Name string,

Genre string,  
Global\_Sales float   
);

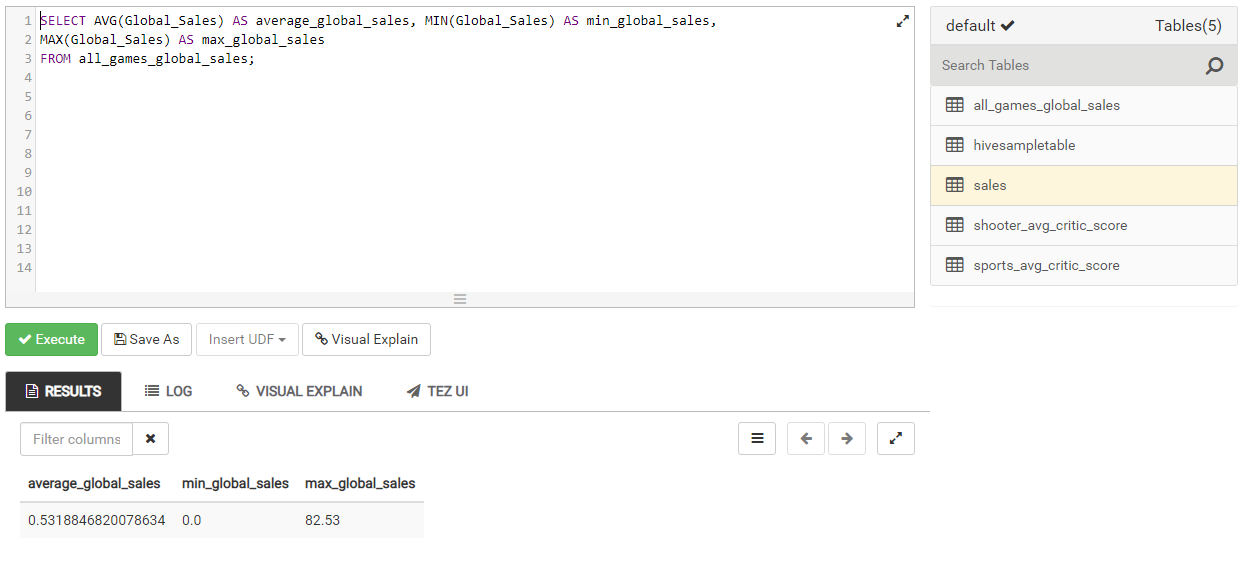
INSERT OVERWRITE TABLE all\_games\_global\_sales  
SELECT Name, Genre, Global\_Sales  
FROM sales;

Use the following code to ensure your aggregation has been performed correctly:

SELECT AVG(Global\_Sales) AS average\_global\_sales, MIN(Global\_Sales) AS min\_global\_sales,

MAX(Global\_Sales) AS max\_global\_sales

FROM all\_games\_global\_sales;



You should see a table that looks like the above.

Use the following code to create the table for sports games:

CREATE TABLE sports\_games\_global\_sales  
(  
Name string,

Genre string,  
Global\_Sales float   
);

INSERT OVERWRITE TABLE sports\_games\_global\_sales  
SELECT Name, Genre, Global\_Sales  
FROM sales;

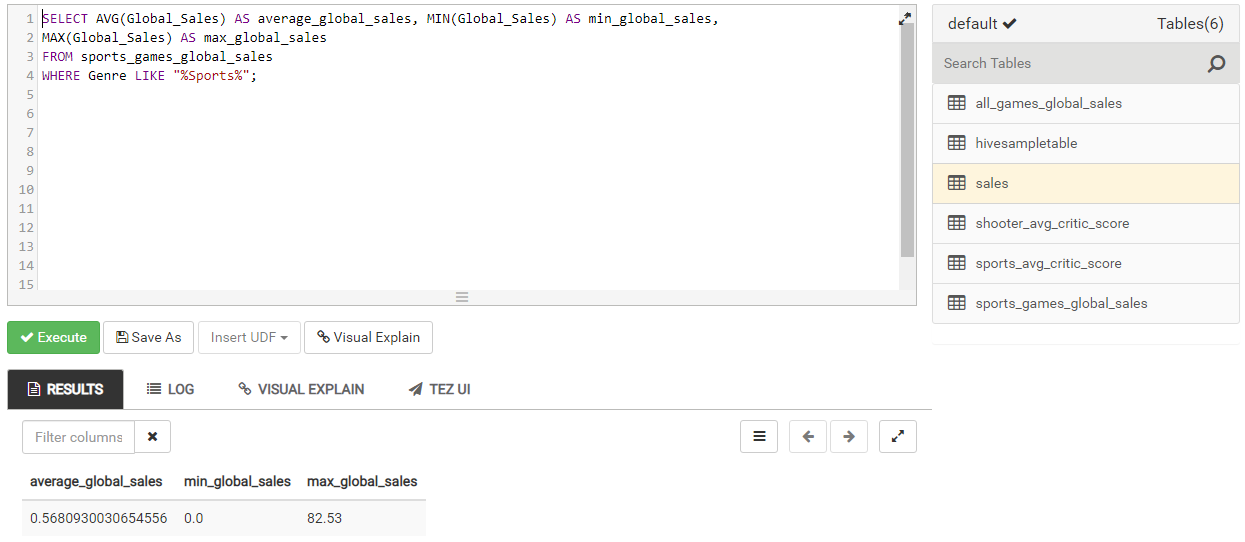
Use the following code to ensure your aggregation has been performed correctly:

SELECT AVG(Global\_Sales) AS average\_global\_sales, MIN(Global\_Sales) AS min\_global\_sales,

MAX(Global\_Sales) AS max\_global\_sales

FROM sports\_games\_global\_sales

WHERE Genre LIKE "%Sports%";



You should see a table that looks like the above.

Use the following code to create the table for sports games:

CREATE TABLE shooter\_games\_global\_sales  
(  
Name string,

Genre string,  
Global\_Sales float   
);

INSERT OVERWRITE TABLE shooter\_games\_global\_sales  
SELECT Name, Genre, Global\_Sales  
FROM sales;

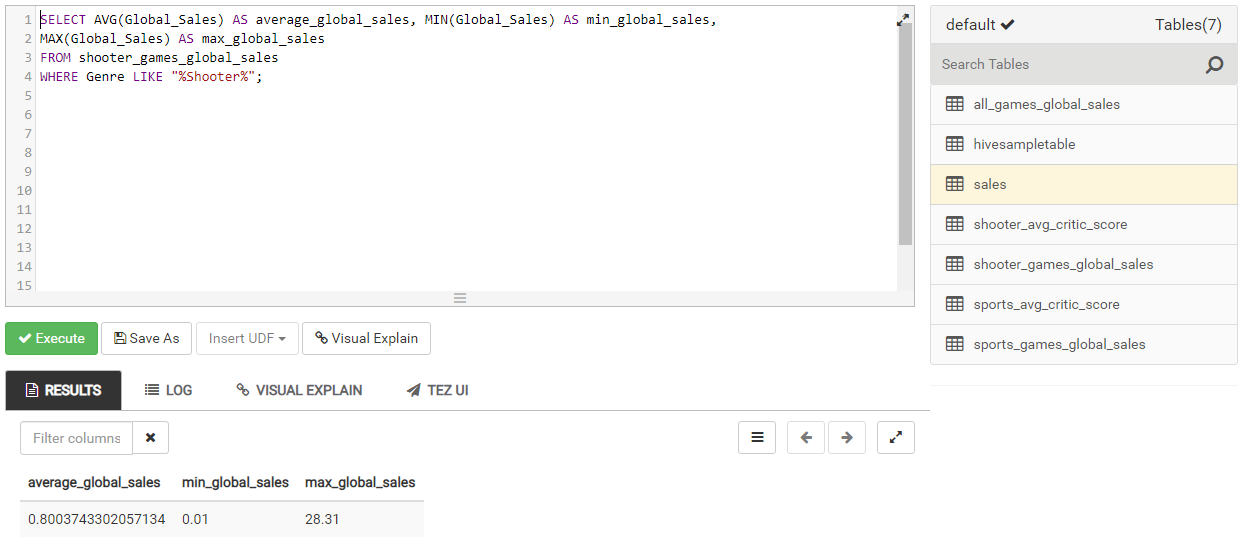
Use the following code to ensure your aggregation has been performed correctly:

SELECT AVG(Global\_Sales) AS average\_global\_sales, MIN(Global\_Sales) AS min\_global\_sales,

MAX(Global\_Sales) AS max\_global\_sales

FROM shooter\_games\_global\_sales

WHERE Genre LIKE "%Shooter%";



You should see a table that looks like the above.

**Step 3 (Create two tables (one for sports games and one for shooter games) for global sales and the number of games matching that global sales amount):**

Use the following code to create the table for sports games global sales:

CREATE TABLE sports\_games\_countglobal\_sales  
(  
Name string,

Genre string,  
Global\_Sales float   
);

INSERT OVERWRITE TABLE sports\_games\_countglobal\_sales  
SELECT Name, Genre, Global\_Sales  
FROM sales;

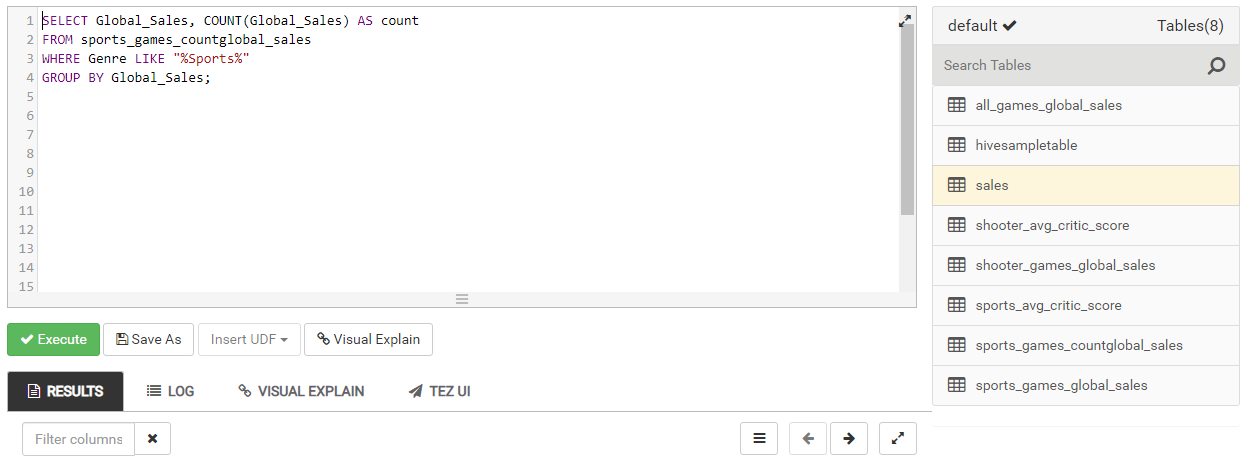
Use the following code to ensure your aggregation has been performed correctly:

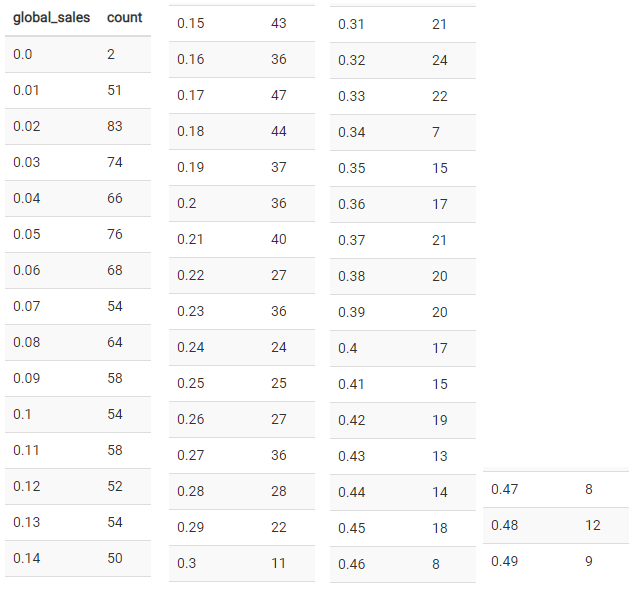
SELECT Global\_Sales, COUNT(Global\_Sales) AS count

FROM sports\_games\_countglobal\_sales

WHERE Genre LIKE "%Sports%"

GROUP BY Global\_Sales;





You should see a table that looks like the above.

Use the following code to create the table for sports games global sales:

CREATE TABLE shooter\_games\_countglobal\_sales  
(  
Name string,

Genre string,  
Global\_Sales float   
);

INSERT OVERWRITE TABLE shooter\_games\_countglobal\_sales  
SELECT Name, Genre, Global\_Sales  
FROM sales;

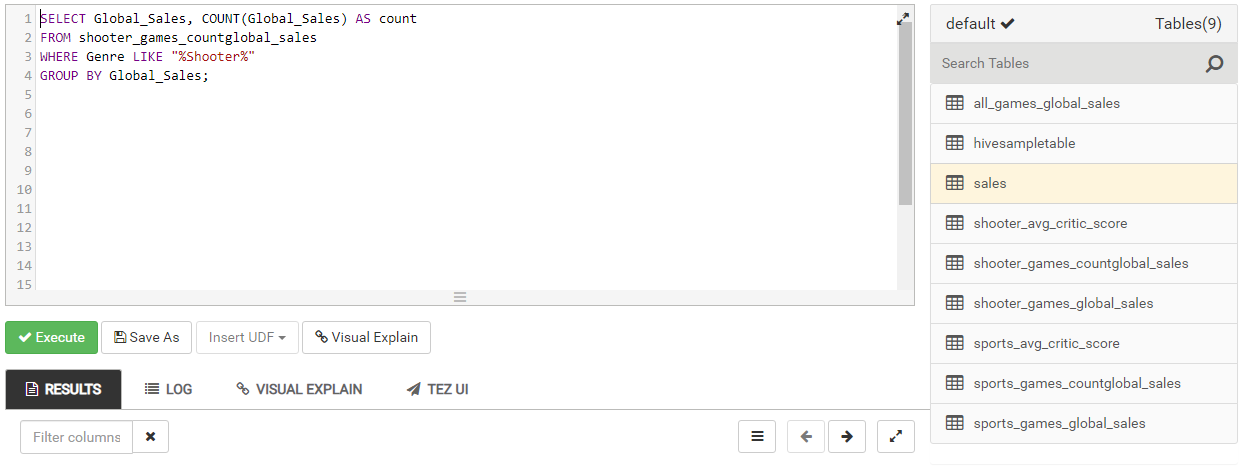
Use the following code to ensure your aggregation has been performed correctly:

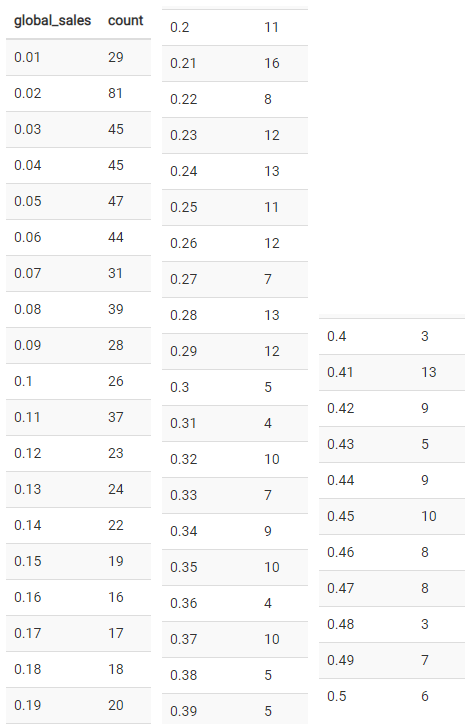
SELECT Global\_Sales, COUNT(Global\_Sales) AS count

FROM shooter\_games\_countglobal\_sales

WHERE Genre LIKE "%Shooter%"

GROUP BY Global\_Sales;





You should see a table that looks like the above.

Congratulations you have now completed all of your data aggregations!