**Experiment No. 08**

**Title: Execution of OLAP operations**

**Batch: B2** **Roll No.: 1914078** **Experiment No.:08**

**Aim: To execute OLAP operations.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Resources needed:** Postgres

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Theory**

In computing, online analytical processing, or OLAP is an approach to answering multi-dimensional analytical (MDA) queries. OLAP is part of the broader category of business intelligence, which also encompasses relational database report writing and data mining. Typical applications of OLAP include business reporting for sales, marketing, management reporting, business process management (BPM), budgeting and forecasting, financial reporting and similar areas, with new applications coming up, such as agriculture. The term OLAP was created as a slight modification of the traditional database term OLTP (Online Transaction Processing).

OLAP tools enable users to analyze multidimensional data interactively from multiple perspectives. OLAP consists of three basic analytical operations: consolidation (roll-up), drill-down, and slicing and dicing. Consolidation involves the aggregation of data that can be accumulated and computed in one or more dimensions. For example, all sales offices are rolled up to the sales department or sales division to anticipate sales trends. By contrast, the drill-down is a technique that allows users to navigate through the details. For instance, users can view the sales by individual products that make up a region’s sales. Slicing and dicing is a feature whereby users can take out (slicing) a specific set of data of the OLAP cube and view (dicing) the slices from different viewpoints.

OLAP queries can be implemented by using analytical SQL functions

Oracle has extensions to ANSI SQL to allow to quickly computing aggregations and rollups.

These new statements include:

* rollup
* cube
* grouping

These simple SQL operators allow creating easy aggregations directly inside the SQL.

**Creating tabular aggregates with ROLLUP:**

ROLLUP enables an SQL statement to calculate multiple levels of subtotals across a specified group of dimensions. It also calculates a grand total. ROLLUP is a simple extension to the GROUP BY clause, so its syntax is extremely easy to use. Create cross-tabular reports with CUBE:

In multidimensional jargon, a “cube” is a cross-tabulated summary of detail rows. CUBE enables a SELECT statement to calculate subtotals for all possible combinations of a group of dimensions. It also calculates a grand total.

This is the set of information typically needed for all cross-tabular reports, so CUBE can calculate a cross-tabular report with a single select statement.

Note in the example below that totals are calculated for each department, and also for each job category.

**Syntax:**

CREATE TABLE sales

(

year INT,

country VARCHAR(20),

product VARCHAR(32),

profit INT

);

mysql> SELECT year, SUM(profit) AS profit

FROM sales

GROUP BY year;

+------+--------+

| year | profit |

+------+--------+

| 2000 | 4525 |

| 2001 | 3010 |

+------+--------+

mysql> SELECT year, SUM(profit) AS profit

FROM sales

GROUP BY year WITH ROLLUP;

+------+--------+

| year | profit |

+------+--------+

| 2000 | 4525 |

| 2001 | 3010 |

| NULL | 7535 |

+------+--------+

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Activity:**

1. Create database in postgres
2. Execute OLAP queries on database table using rollup,cube and grouping sets.
3. Observe the result.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Results: (Program printout with output):**

**Create database in Postgres:**

* *First, we create our ‘items’ table in the database:*

CREATE TABLE items

(

item\_id INT primary key,

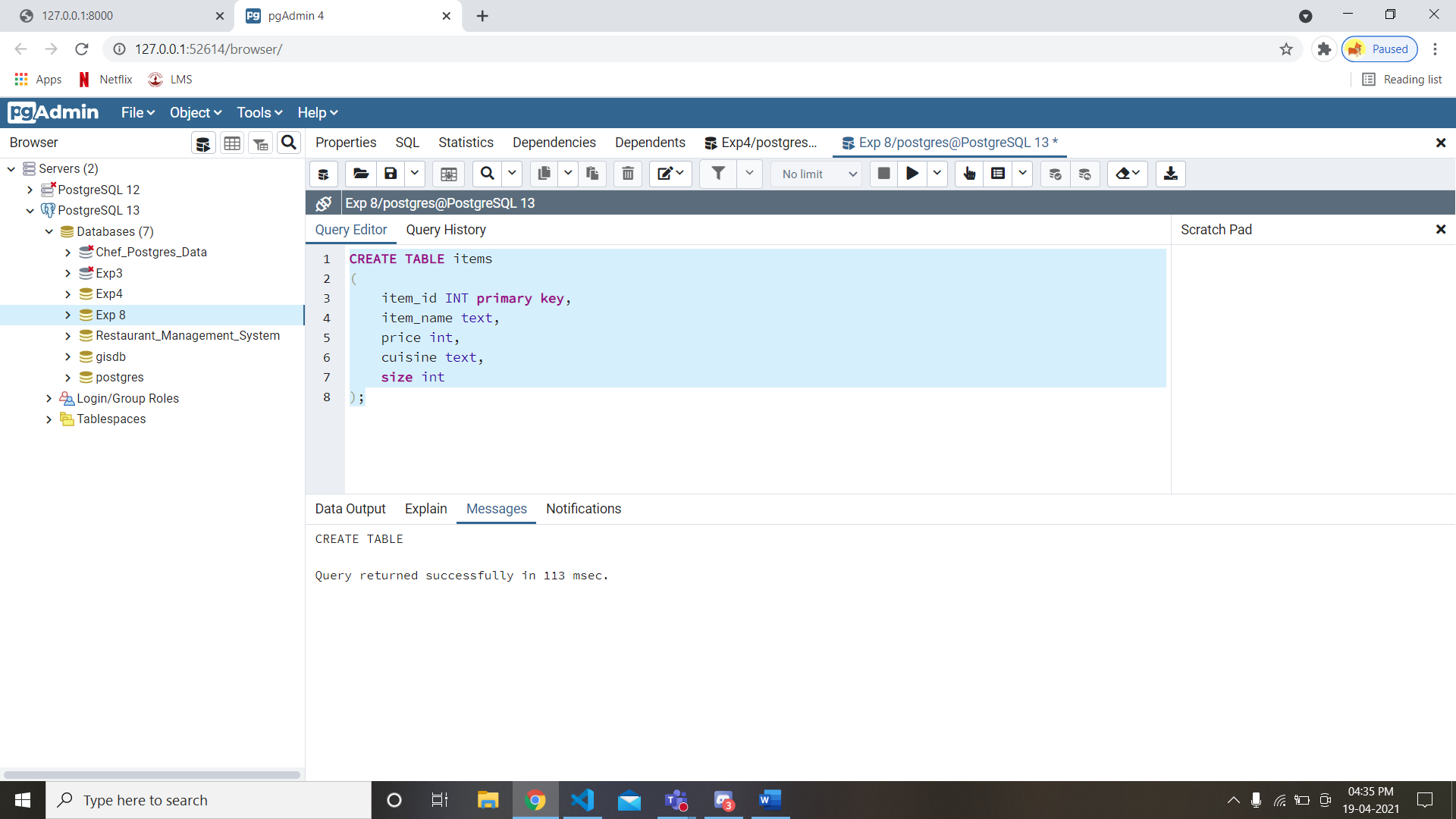
item\_name text,

price int,

cuisine text,

dish\_size text

);



* *Now, we insert values into our ‘items’ table in the database:*

insert into items values(1,'Hakka Noodles', 250, 'Chinese', 'Medium');

insert into items values(2,'Fried Rice', 300, 'Chinese', 'Large');

insert into items values(3,'Shezwan Noodles', 270, 'Chinese', 'Medium');

insert into items values(4,'Pizza', 350, 'Italian', 'Small');

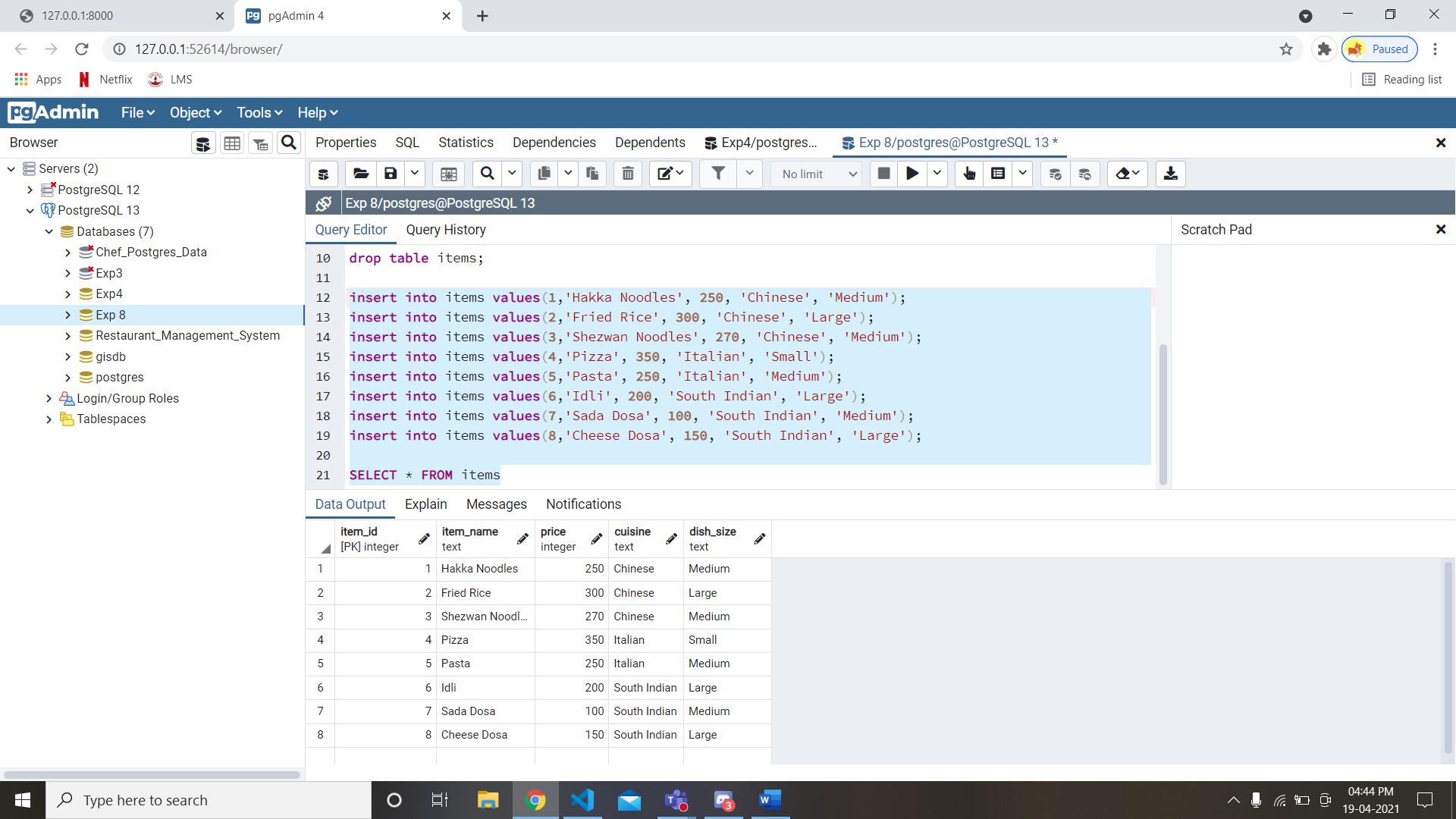
insert into items values(5,'Pasta', 250, 'Italian', 'Medium');

insert into items values(6,'Idli', 200, 'South Indian', 'Large');

insert into items values(7,'Sada Dosa', 100, 'South Indian', 'Medium');

insert into items values(8,'Cheese Dosa', 150, 'South Indian', 'Large');

SELECT \* FROM items



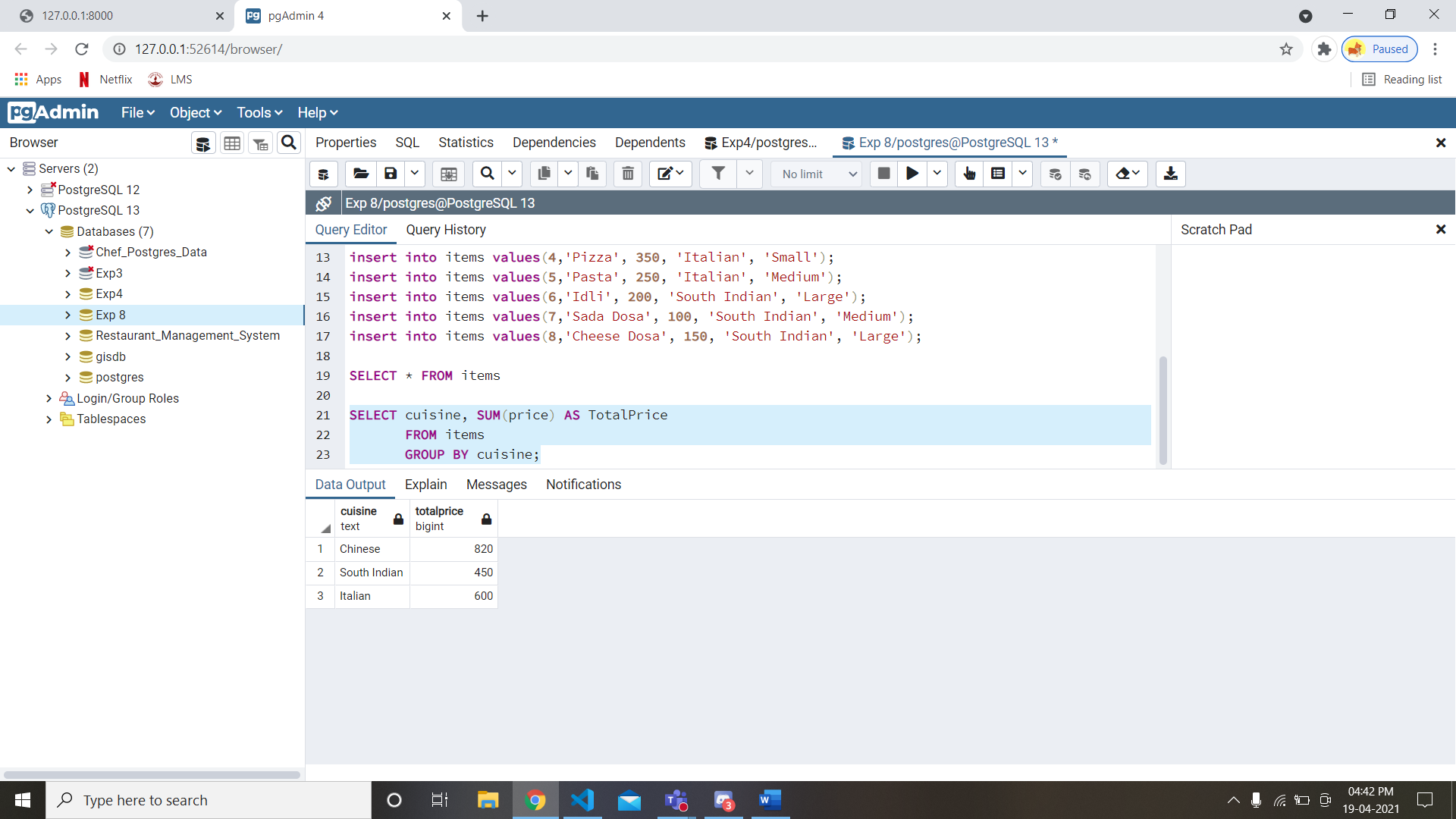
**Execute OLAP queries on database table using rollup, cube and grouping sets and observe the results:**

* *Now, we use ‘GROUP BY’ on our ‘Cuisine’ column and display each value in the cuisine column along with the sum of the prices of each item of that particular value(i.e. cuisine):*

SELECT cuisine, SUM(price) AS TotalPrice

FROM items

GROUP BY cuisine;

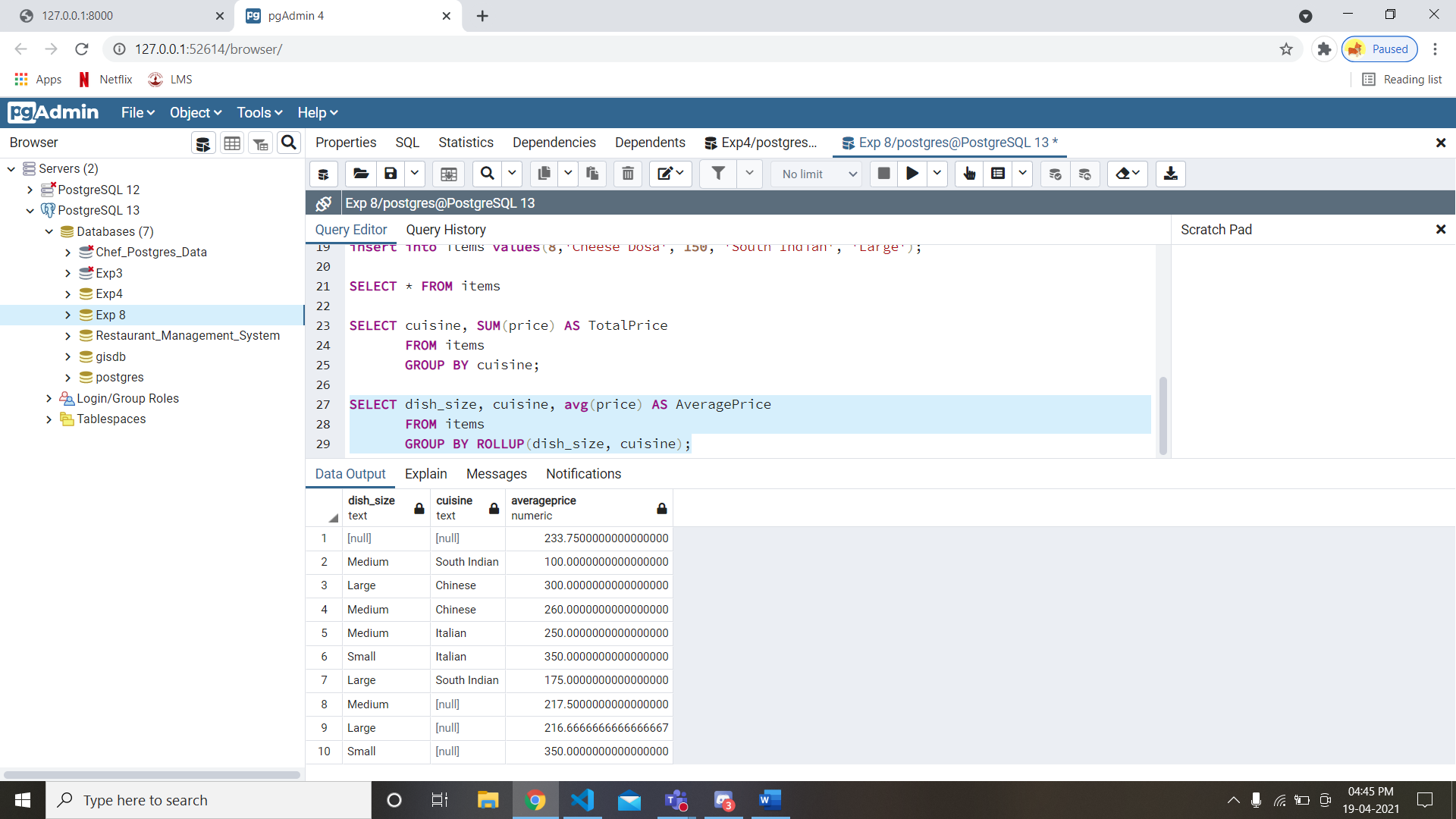


* *Now, we use ‘GROUP BY’ along with ‘ROLLUP’ .*

SELECT dish\_size, cuisine, avg(price) AS AveragePrice

FROM items

GROUP BY ROLLUP(dish\_size, cuisine);



**Observation:**

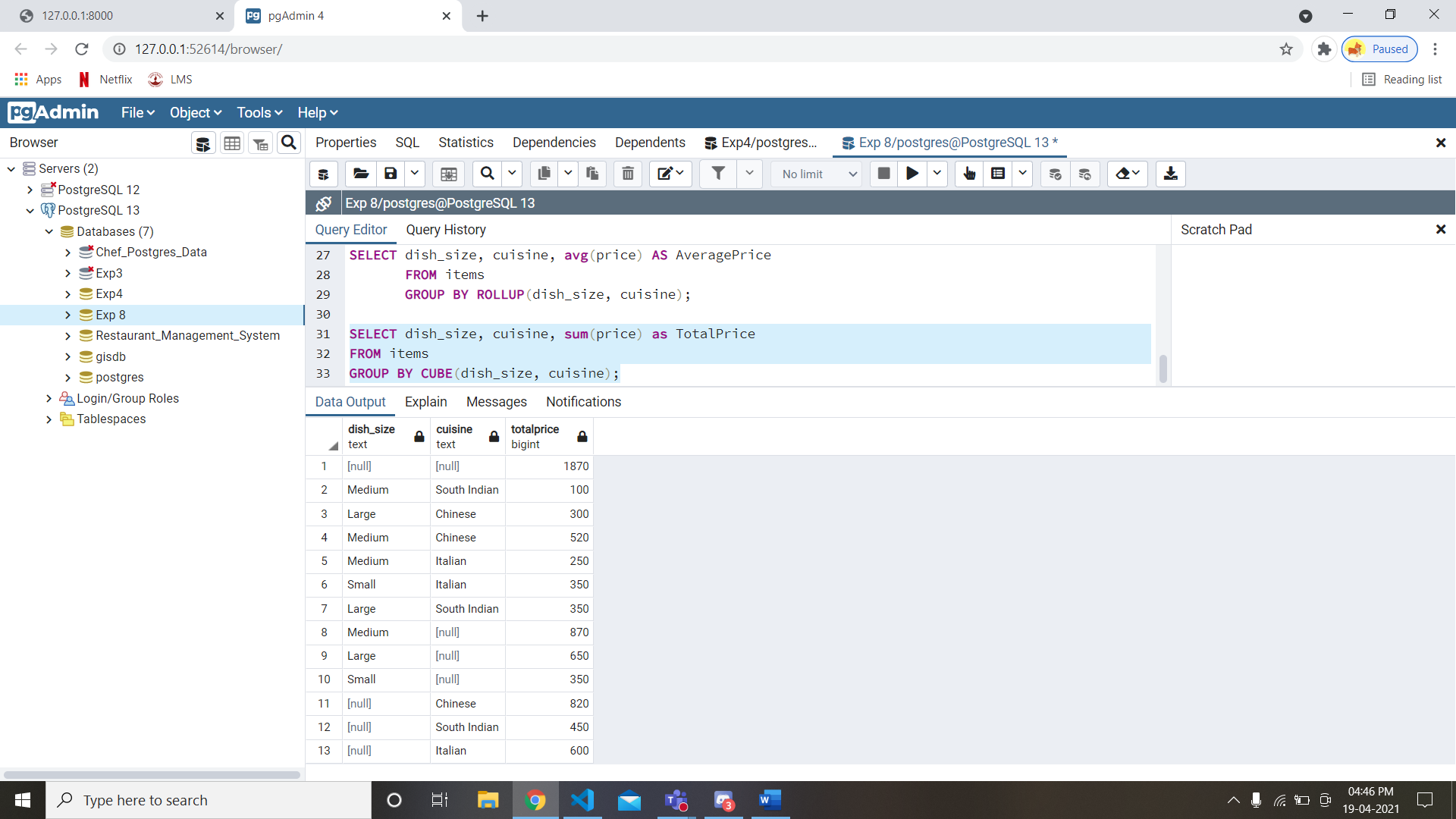
The logic of rollup is that it takes multiple arguments and applies group by while taking both into consideration. As shown above it group by of all the possible groups between dish\_size and cuisine and then it appends a group by clause applied to the first argument given, in this case dish\_size.

* *Now, we use ‘GROUP BY’ along with ‘ROLLUP’ on our ‘Cuisine’ column and display each value in the cuisine column along with the sum of the prices of each item of that particular value(i.e. cuisine):*

SELECT dish\_size, cuisine, sum(price) as TotalPrice

FROM items

GROUP BY CUBE(dish\_size, cuisine);



**Observation:**

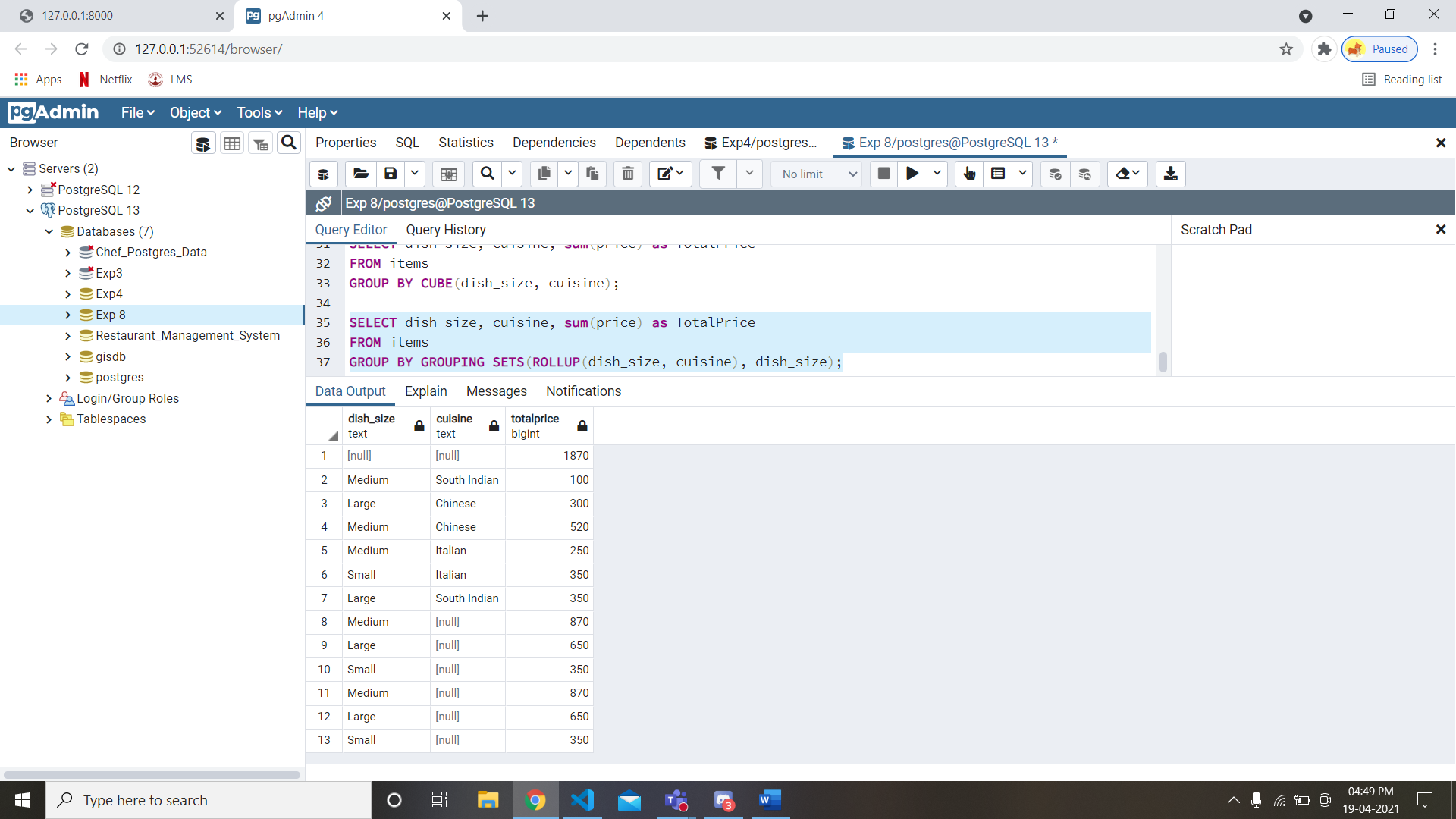
The logic of cube is that it takes multiple arguments and applies group by while taking both into consideration. As shown above it group by of all the possible groups between dish\_size and cuisine and then it appends a group by clause applied to each argument given like, in this case dish\_size and cuisine individually.

* *Now, we use ‘GROUP BY’ along with ‘ROLLUP’ on our ‘Cuisine’ column and display each value in the cuisine column along with the sum of the prices of each item of that particular value(i.e. cuisine):*

SELECT dish\_size, cuisine, sum(price) as TotalPrice

FROM items

GROUP BY GROUPING SETS(ROLLUP(dish\_size, cuisine), dish\_size);



**Observations:**

The logic of grouping sets is that it takes multiple arguments and it just takes the union of those two arguments applied with group by individually. Like here group by rollup union group by dish\_size is taken and hence it looks like cube.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Questions:**

1. Differentiate OLAP, ROLAP, HOLAP and MOLAP.

**Answer:**

**OLAP:**

The term online analytical processing (OLAP) usually refers to specialized tools that make warehouse data easily available. An OLAP cube is a logical structure that defines the metadata. The term cube describes existing measure groups and dimension tables and should not be interpreted as having limited dimensions. A cube is a combination of all existing measure groups. A measure group is a group of measures that match the business logic of the data and is another logical structure that defines metadata so that client tools can access the data. Each measure group contains the detail values that are stored in the fact table (copied or dynamically retrieved values). OLAP cubes contain lots of metadata; metadata in its simplest definition is data about data. Multidimensional expressions, or MDX, is a metadata-based query language that helps you query OLAP cubes.

**Difference between ROLAP, MOLAP and HOLAP :**

| **Basis** | **ROLAP** | **MOLAP** | **HOLAP** |
| --- | --- | --- | --- |
| **Storage location for summary aggregation** | Relational Database is used as storage location for summary aggregation. | Multidimensional Database is used as storage location for summary aggregation. | Multidimensional Database is used as storage location for summary aggregation. |
| **Processing time** | Processing time of ROLAP is very slow. | Processing time of MOLAP is fast. | Processing time of HOLAP is fast. |
| **Storage space requirement** | Large storage space requirement in ROLAP as compare to MOLAP and HOLAP. | Medium storage space requirement in MOLAP as compare to ROLAP and HOLAP. | Small storage space requirement in HOLAP as compare to MOLAP and ROLAP. |
| **Storage location for detail data** | Relational database is used as storage location for detail data. | Multidimensional database is used as storage location for detail data. | Relational database is used as storage location for detail data. |
| **Latency** | Low latency in ROLAP as compare to MOLAP nad HOLAP. | High latency in MOLAP as compare to ROLAP and HOLAP. | Medium latency in HOLAP as compare to MOLAP and ROLAP. |
| **Query response time** | Slow query response time in ROLAP as compare to MOLAP and HOLAP. | Fast query response time in MOLAP as compare to ROLAP and HOLAP. | Medium query response time in HOLAP as compare to MOLAP and ROLAP. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Outcomes:**

**CO4:**Apply ETL processing and Online Analytical Processing on the warehouse data.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion: (Conclusion to be based on the outcomes achieved)**

Through this experiment we executed OLAP operations and learnt how to use simple SQL operators like rollup, cube and grouping in creating easy aggregations directly inside the SQL. We executed these operators on the table created in our database and made various observations about the different results of these operators.

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of faculty in-charge with date**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**References:**

**Books/ Journals/ Websites:**

1 Elmasri and Navathe, “Fundamentals of Database Systems”, Pearson Education

2 Korth, Silberchatz, Sudarshan, “Database System Concepts” McGraw Hill

3 Raghu Ramakrishnan, Johannes Gerhke, “Database Management Systems” McGraw Hill

Page No: