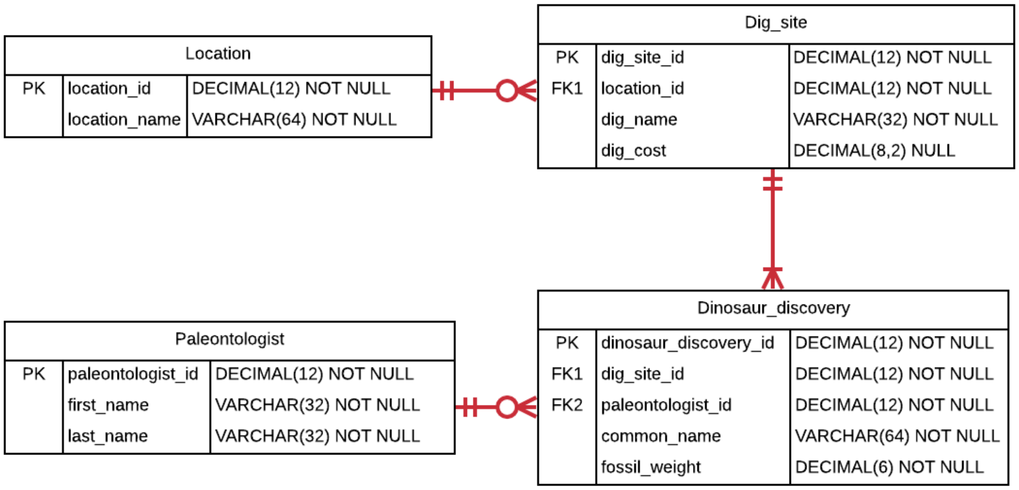
Section One – Aggregating Data

Section Background

To practice aggregating data, you will be working with the following simplified dinosaur discovery schema.



Location

PK	location_id	DECIMAL(12) NOT NULL
	location_name	VARCHAR(64) NOT NULL

Dig_site

PK	dig_site_id	DECIMAL(12) NOT NULL
FK1	location_id	DECIMAL(12) NOT NULL
	dig_name	VARCHAR(32) NOT NULL
	dig_cost	DECIMAL(8,2) NULL

Paleontologist

PK	paleontologist_id	DECIMAL(12) NOT NULL
	first_name	VARCHAR(32) NOT NULL
	last_name	VARCHAR(32) NOT NULL

Dinosaur_discovery

PK	dinosaur_discovery_id	DECIMAL(12) NOT NULL
FK1	dig_site_id	DECIMAL(12) NOT NULL
FK2	paleontologist_id	DECIMAL(12) NOT NULL
	common_name	VARCHAR(64) NOT NULL
	fossil_weight	DECIMAL(6) NOT NULL

This schema contains basic information about various dinosaur discoveries and the elements that comprise those discoveries, such as the location they were found, the dig site, and who discovered them.

In this schema, the *Location* table represents the general location of the dig site, such as "Utah" or "Arizona". The *Dig_site* table represents the specific site within the location where the dinosaur remains were discovered. Every dig site has a name, and the total cost for digging at that site. The *Dinosaur_discovery* table represents the actual discovery of the dinosaur remains. Every discovery happens at a dig site, and is discovered by a paleontologist. A discovery also has the common name of the dinosaur, and the weight of the remains. The *Paleontologist* table represents the person who made the discovery, and the first and last name is tracked in the database.

The schema is intentionally simplified compared to what you might see in a real-world production schema. Many attributes and entities that would exist in a production database are not present. Nevertheless, there is sufficient complexity in the existing relationships and attributes to challenge you to learn various aggregation scenarios you encounter in real-world schemas.

Authored by Warren Mansur and Pamela Farr.

As a reminder, for each step that requires SQL, make sure to capture a screenshot of the command and the results of its execution. *Further, make sure to eliminate unneeded columns from the result set, to name your columns something user-friendly and human readable, and to format any prices as currencies.*

Section Steps

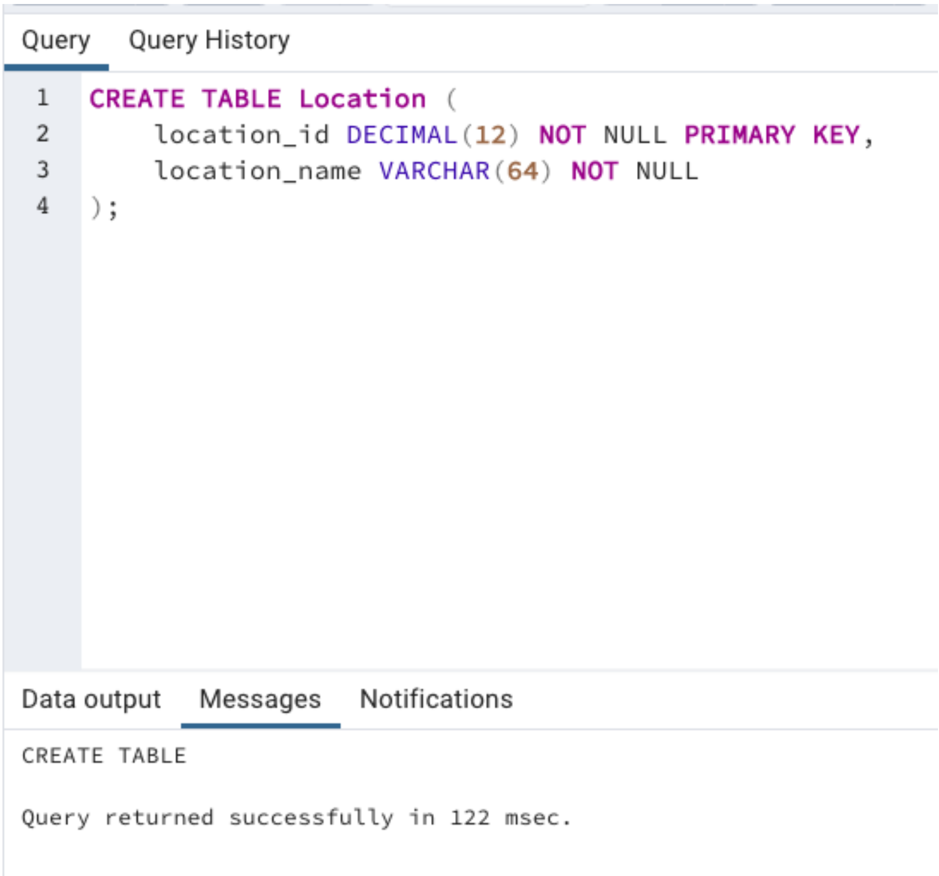
1. ***Creating Table Structure and Data*** – Create the tables in the schema, including all of their columns, datatypes, and constraints, and populate the tables with data. Most but not all the data is given to you in the table below; ***you should also insert information for one additional dinosaur discovery of your choosing.*** Although the data is in flattened representation below, you will need to insert the data relationally into the schema with foreign keys referencing the appropriate primary keys. You may choose any primary key values you would like for each table. We will learn in a later lab how to automatically generate primary key values.

Location	**Dig Name**	**Dig Cost**	**Dinosaur Common Name**	**Weight (in pounds)**	**Paleontologist**
Stonesfield	Great British Dig	$8,000	Megalosaurus	3000	William Buckland
Stonesfield	Great British Dig	$8,000	Apatosaurus	4000	William Buckland
Stonesfield	Great British Dig	$8,000	Triceratops	4500	William Buckland
Stonesfield	Great British Dig	$8,000	Stegosaurus	3500	William Buckland
Utah	Parowan Dinosaur Tracks	$10,000	Parasaurolophus	6000	John Ostrom
Utah	Parowan Dinosaur Tracks	$10,000	Tyrannosaurus Rex	5000	John Ostrom
Utah	Parowan Dinosaur Tracks	$10,000	Velociraptor	7000	John Ostrom
Arizona	Dynamic Desert Dig	$3,500	Tyrannosaurus Rex	6000	John Ostrom
Arizona	Dynamic Desert Dig	$3,500	Velociraptor	6500	John Ostrom
Stonesfield	Mission Jurassic Dig		Spinosaurus	8000	Henry Osborn
Stonesfield	Mission Jurassic Dig		Diplodocus	9000	Henry Osborn
Stonesfield	Ancient Site Dig	$5,500	Tyrannosaurus Rex	7500	Henry Osborn

Note that the Dig Cost for “Mission Jurassic Dig” is null (has no value).

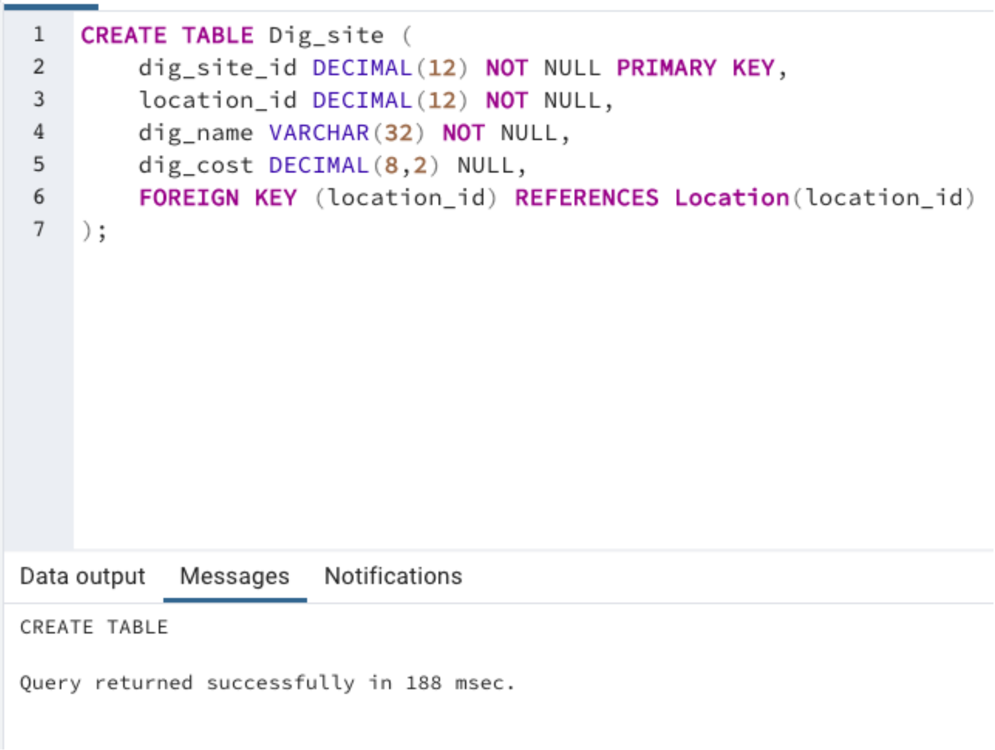
- I added one dinosaur discovery which is green color in the table above

- ***Create Tables***



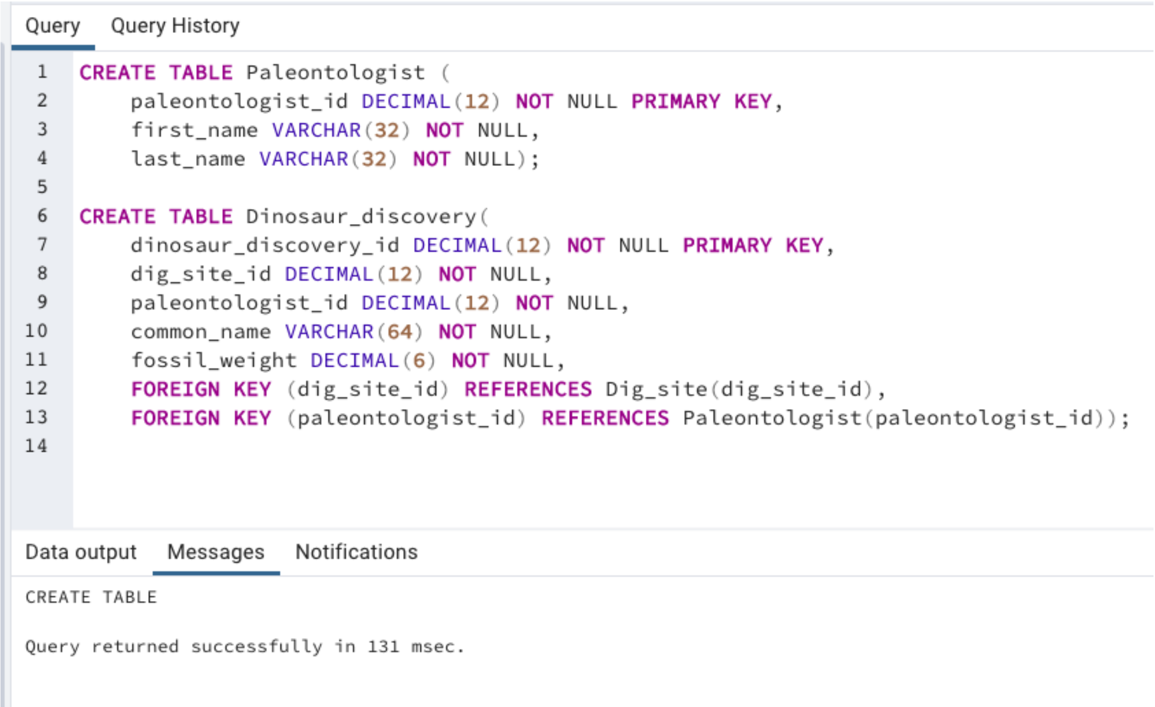
```
CREATE TABLE Location (
    location_id DECIMAL(12) NOT NULL PRIMARY KEY,
    location_name VARCHAR(64) NOT NULL
);
```

```
CREATE TABLE

Query returned successfully in 122 msec.
```



```
CREATE TABLE Dig_site (
    dig_site_id DECIMAL(12) NOT NULL PRIMARY KEY,
    location_id DECIMAL(12) NOT NULL,
    dig_name VARCHAR(32) NOT NULL,
    dig_cost DECIMAL(8,2) NULL,
    FOREIGN KEY (location_id) REFERENCES Location(location_id)
);
```

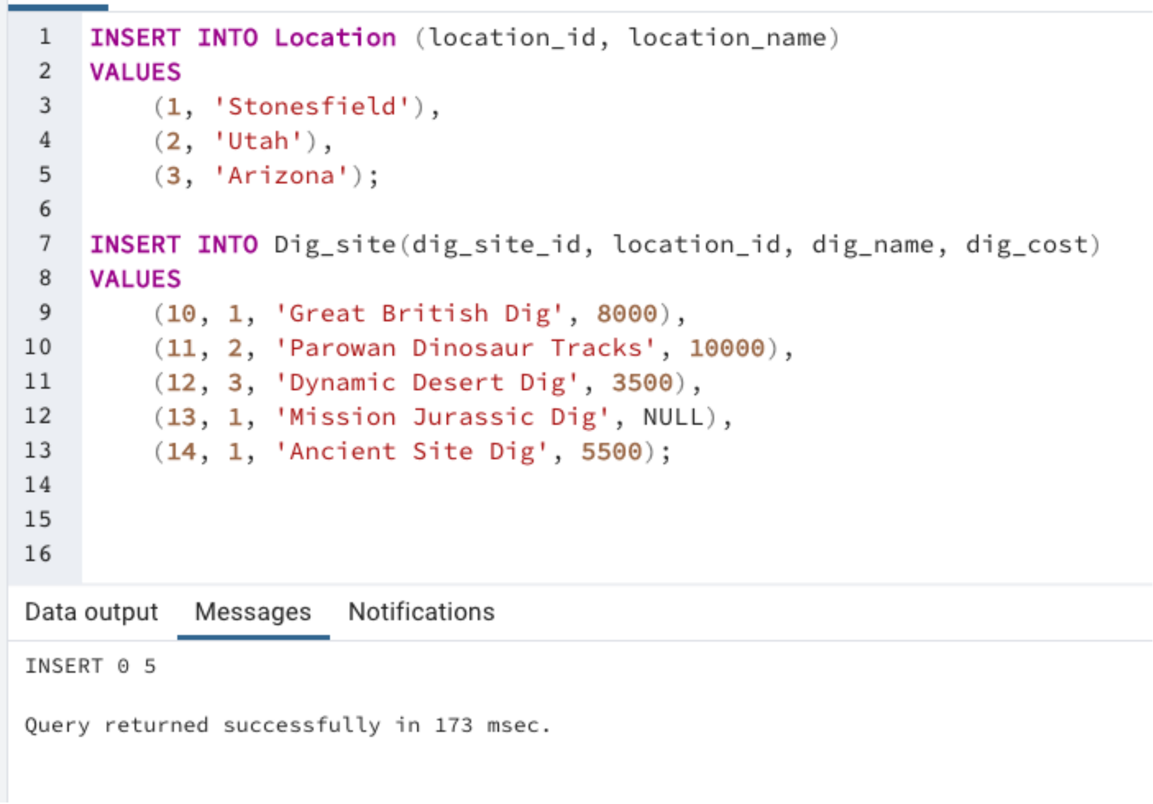
```
CREATE TABLE

Query returned successfully in 188 msec.
```



```sql
CREATE TABLE Paleontologist (
    paleontologist_id DECIMAL(12) NOT NULL PRIMARY KEY,
    first_name VARCHAR(32) NOT NULL,
    last_name VARCHAR(32) NOT NULL);

CREATE TABLE Dinosaur_discovery(
    dinosaur_discovery_id DECIMAL(12) NOT NULL PRIMARY KEY,
    dig_site_id DECIMAL(12) NOT NULL,
    paleontologist_id DECIMAL(12) NOT NULL,
    common_name VARCHAR(64) NOT NULL,
    fossil_weight DECIMAL(6) NOT NULL,
    FOREIGN KEY (dig_site_id) REFERENCES Dig_site(dig_site_id),
    FOREIGN KEY (paleontologist_id) REFERENCES Paleontologist(paleontologist_id));
```

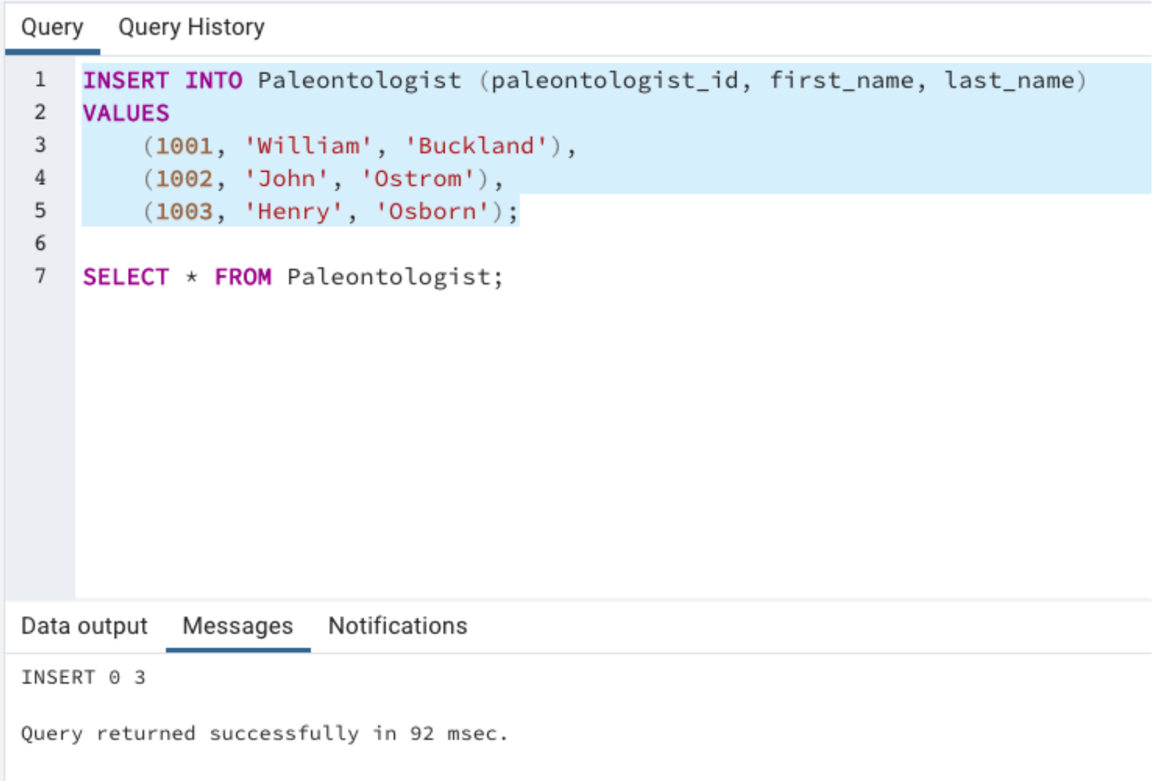
```
CREATE TABLE

Query returned successfully in 131 msec.
```

- ***Insert Data into tables***



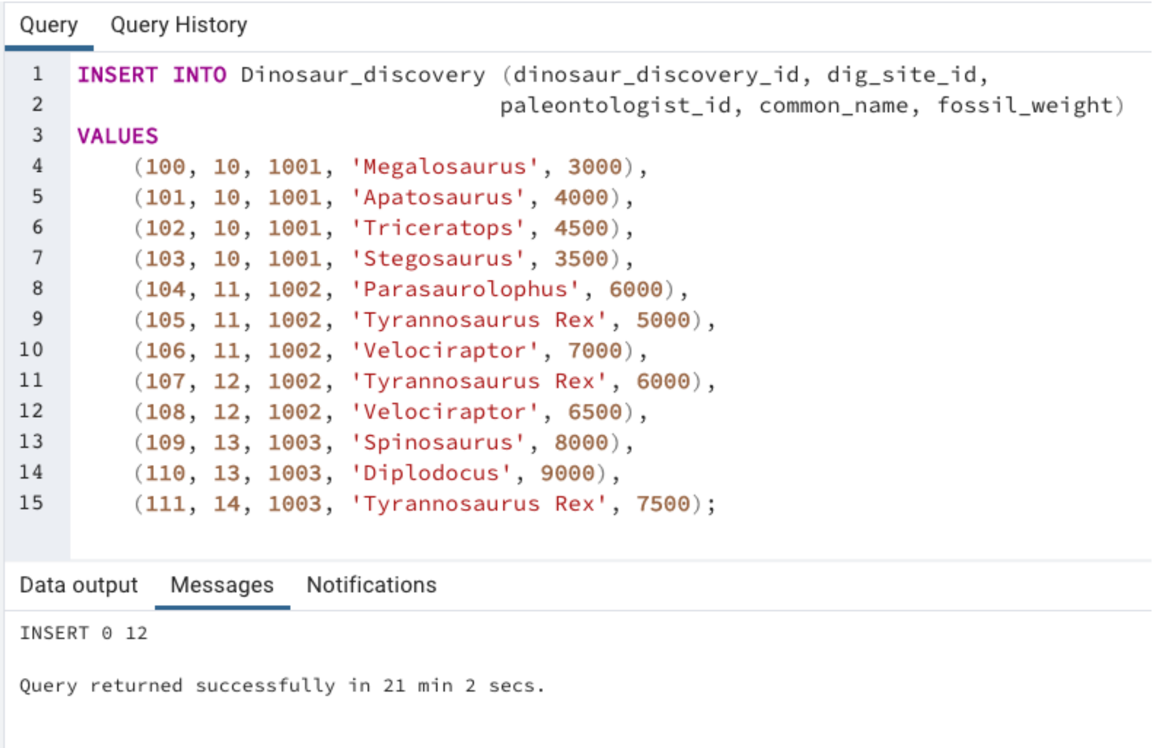
```sql
INSERT INTO Location (location_id, location_name)
VALUES
     (1, 'Stonesfield'),
     (2, 'Utah'),
     (3, 'Arizona');

INSERT INTO Dig_site(dig_site_id, location_id, dig_name, dig_cost)
VALUES
     (10, 1, 'Great British Dig', 8000),
     (11, 2, 'Parowan Dinosaur Tracks', 10000),
     (12, 3, 'Dynamic Desert Dig', 3500),
     (13, 1, 'Mission Jurassic Dig', NULL),
     (14, 1, 'Ancient Site Dig', 5500);
```

```
INSERT 0 5

Query returned successfully in 173 msec.
```



```
INSERT INTO Paleontologist (paleontologist_id, first_name, last_name)
VALUES
    (1001, 'William', 'Buckland'),
    (1002, 'John', 'Ostrom'),
    (1003, 'Henry', 'Osborn');

SELECT * FROM Paleontologist;
```

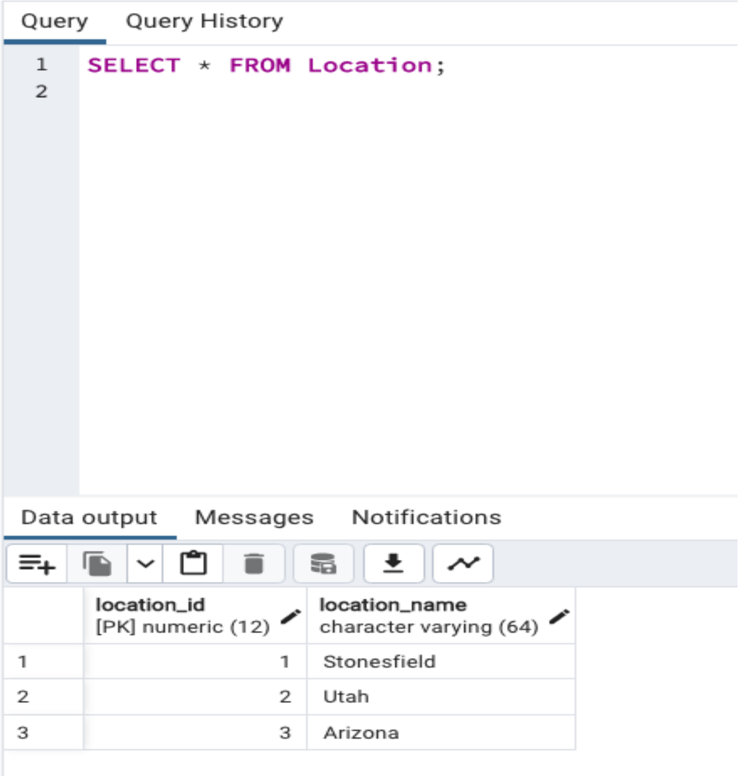
```
INSERT 0 3

Query returned successfully in 92 msec.
```



```
INSERT INTO Dinosaur_discovery (dinosaur_discovery_id, dig_site_id,
                                paleontologist_id, common_name, fossil_weight)
VALUES
    (100, 10, 1001, 'Megalosaurus', 3000),
    (101, 10, 1001, 'Apatosaurus', 4000),
    (102, 10, 1001, 'Triceratops', 4500),
    (103, 10, 1001, 'Stegosaurus', 3500),
    (104, 11, 1002, 'Parasaurolophus', 6000),
    (105, 11, 1002, 'Tyrannosaurus Rex', 5000),
    (106, 11, 1002, 'Velociraptor', 7000),
    (107, 12, 1002, 'Tyrannosaurus Rex', 6000),
    (108, 12, 1002, 'Velociraptor', 6500),
    (109, 13, 1003, 'Spinosaurus', 8000),
    (110, 13, 1003, 'Diplodocus', 9000),
    (111, 14, 1003, 'Tyrannosaurus Rex', 7500);
```

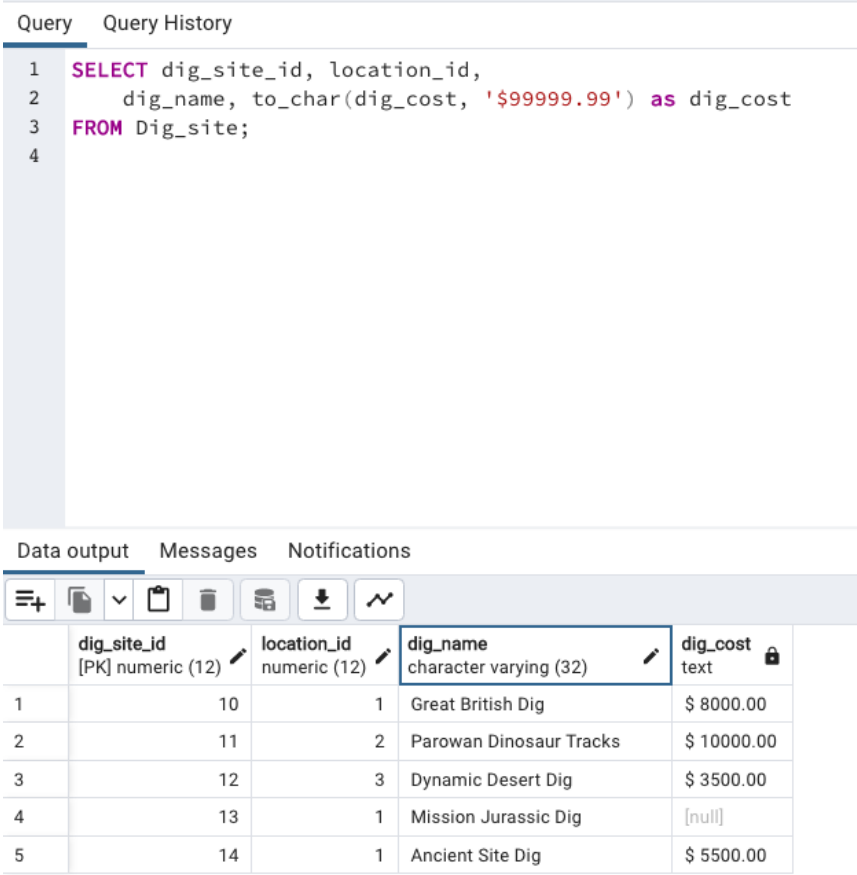
```
INSERT 0 12

Query returned successfully in 21 min 2 secs.
```

- ***View table data***



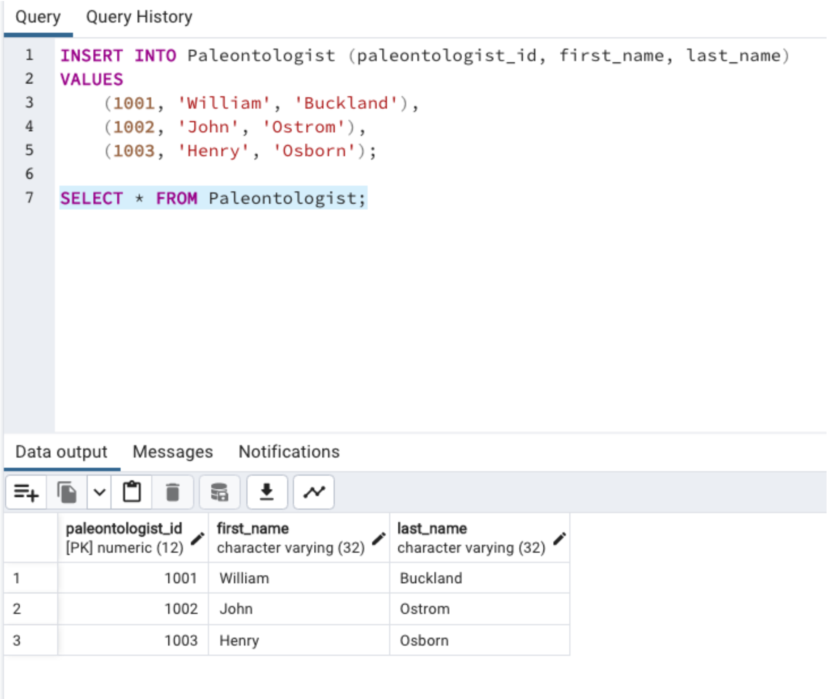
```sql
SELECT * FROM Location;
```

	location_id [PK] numeric (12)	location_name character varying (64)
1	1	Stonesfield
2	2	Utah
3	3	Arizona



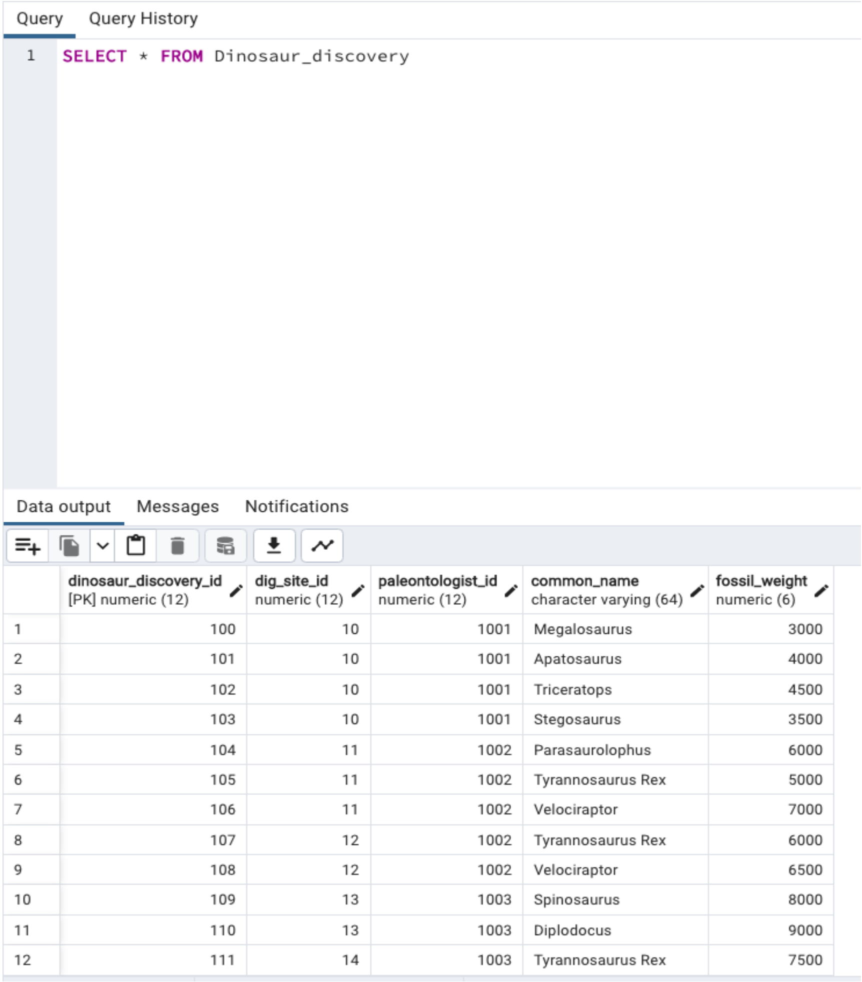
```sql
SELECT dig_site_id, location_id,
    dig_name, to_char(dig_cost, '$99999.99') as dig_cost
FROM Dig_site;
```

	dig_site_id [PK] numeric (12)	location_id numeric (12)	dig_name character varying (32)	dig_cost text
1	10	1	Great British Dig	$ 8000.00
2	11	2	Parowan Dinosaur Tracks	$ 10000.00
3	12	3	Dynamic Desert Dig	$ 3500.00
4	13	1	Mission Jurassic Dig	[null]
5	14	1	Ancient Site Dig	$ 5500.00



```sql
INSERT INTO Paleontologist (paleontologist_id, first_name, last_name)
VALUES
    (1001, 'William', 'Buckland'),
    (1002, 'John', 'Ostrom'),
    (1003, 'Henry', 'Osborn');

SELECT * FROM Paleontologist;
```

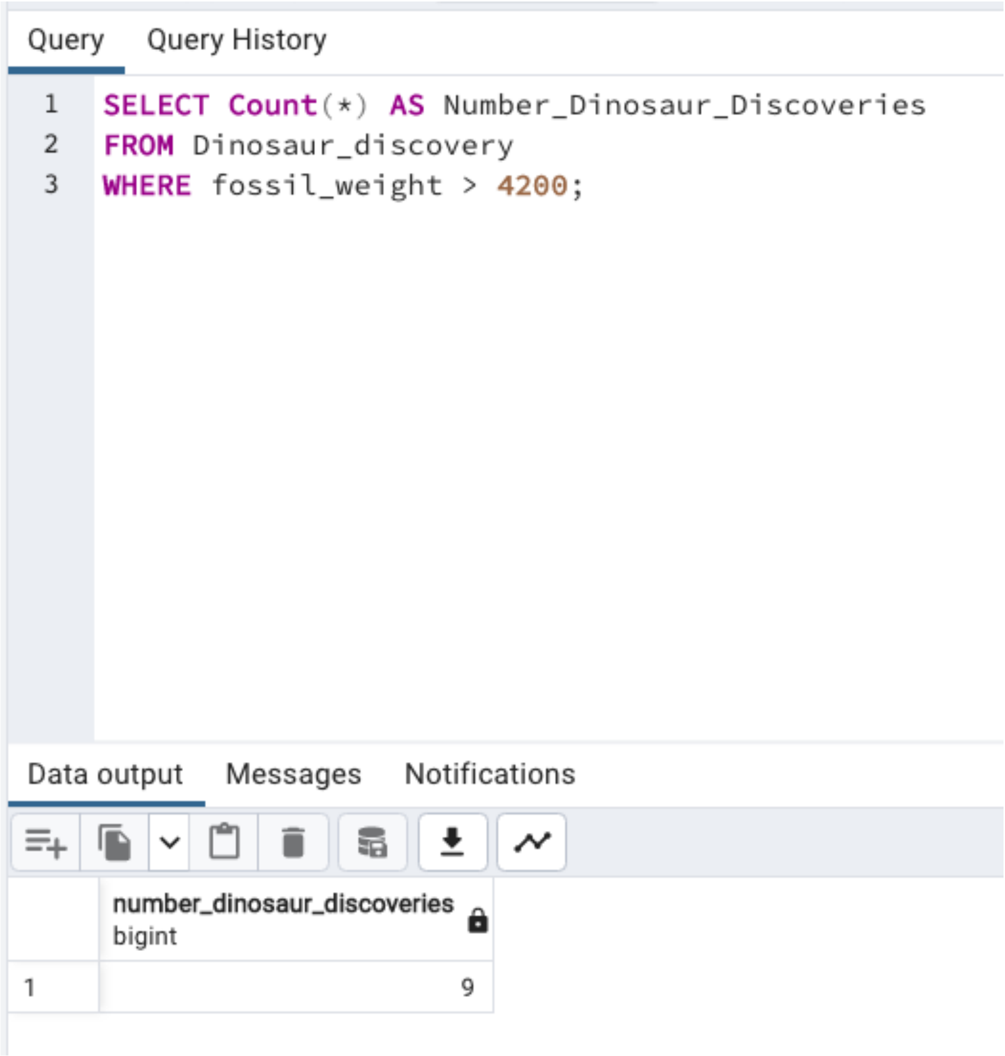
	paleontologist_id [PK] numeric (12)	first_name character varying (32)	last_name character varying (32)
1	1001	William	Buckland
2	1002	John	Ostrom
3	1003	Henry	Osborn



```sql
SELECT * FROM Dinosaur_discovery
```

	dinosaur_discovery_id [PK] numeric (12)	dig_site_id numeric (12)	paleontologist_id numeric (12)	common_name character varying (64)	fossil_weight numeric (6)
1	100	10	1001	Megalosaurus	3000
2	101	10	1001	Apatosaurus	4000
3	102	10	1001	Triceratops	4500
4	103	10	1001	Stegosaurus	3500
5	104	11	1002	Parasaurolophus	6000
6	105	11	1002	Tyrannosaurus Rex	5000
7	106	11	1002	Velociraptor	7000
8	107	12	1002	Tyrannosaurus Rex	6000
9	108	12	1002	Velociraptor	6500
10	109	13	1003	Spinosaurus	8000
11	110	13	1003	Diplodocus	9000
12	111	14	1003	Tyrannosaurus Rex	7500

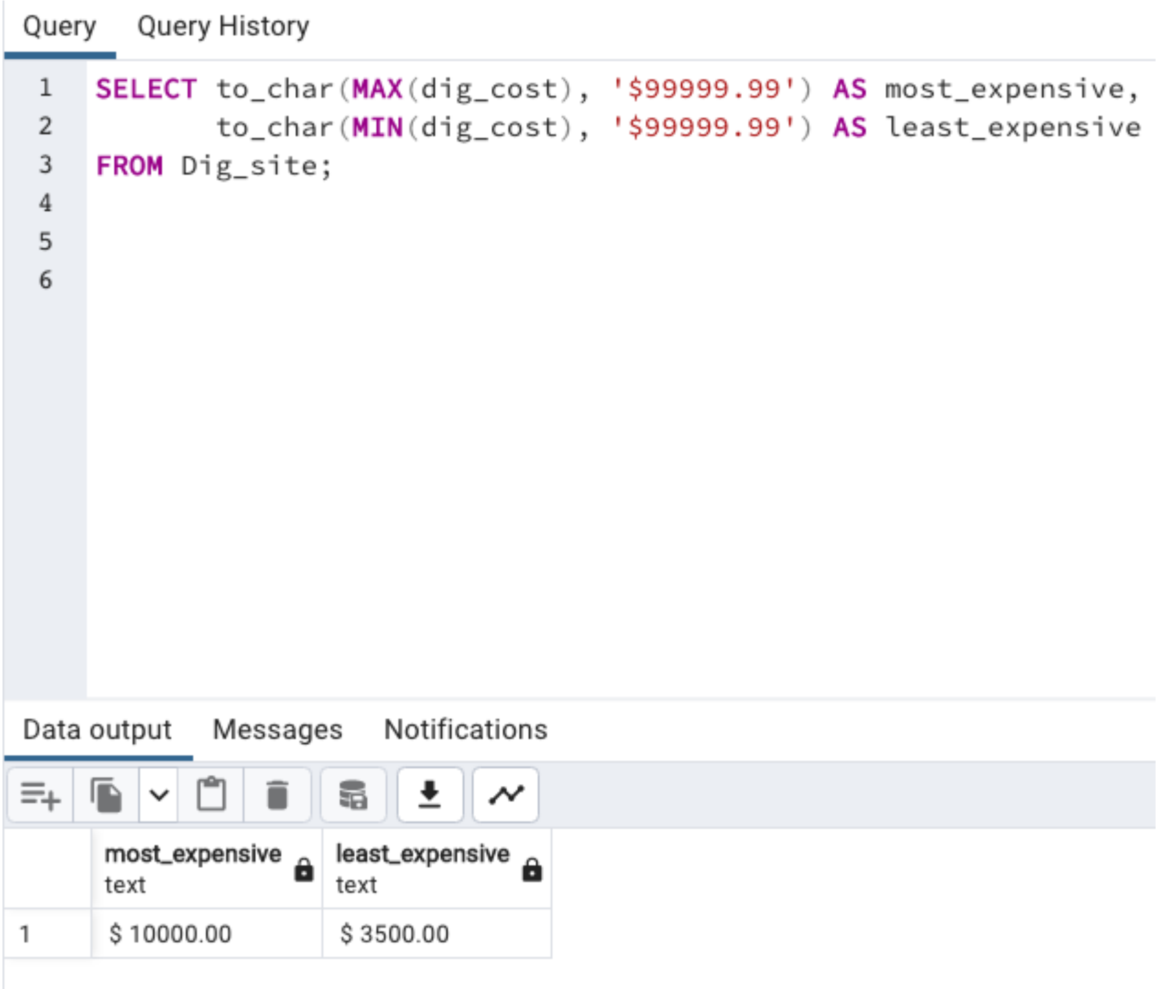
2. ***Counting Matches*** **– A museum wants to know how many dinosaur discoveries weigh at least 4,200 pounds. Write a single query to fulfill this request.**



```
SELECT Count(*) AS Number_Dinosaur_Discoveries
FROM Dinosaur_discovery
WHERE fossil_weight > 4200;
```

	number_dinosaur_discoveries bigint
1	9

3. ***Determining Highest and Lowest*** **– The same museum needs to know the cost of the most expensive and least expensive dinosaur digs. Write a single query to fulfill this request.**



```sql
SELECT to_char(MAX(dig_cost), '$99999.99') AS most_expensive,
       to_char(MIN(dig_cost), '$99999.99') AS least_expensive
FROM Dig_site;
```

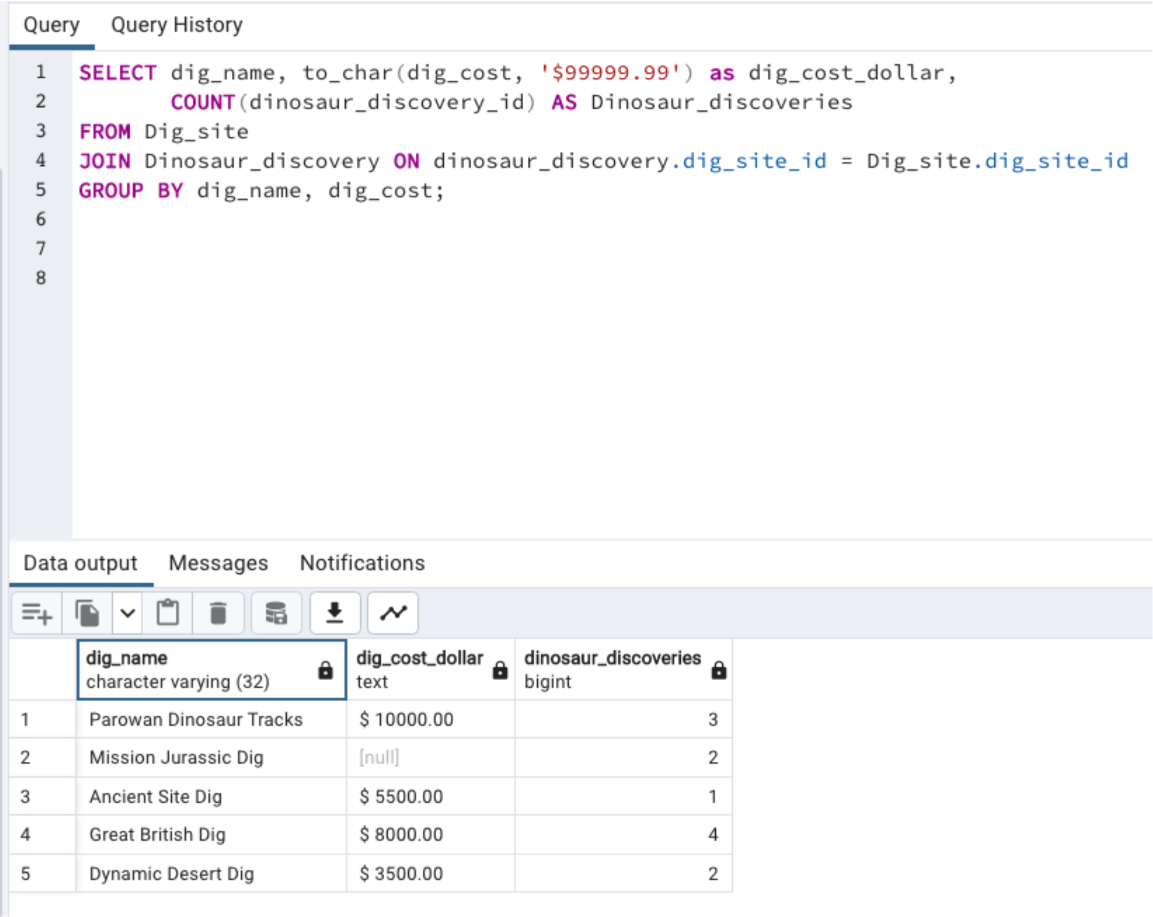
	most_expensive text	least_expensive text
1	$ 10000.00	$ 3500.00

Explain how the SQL processer treated the dig costs for the "Mission Jurassic Dig" differently than the other cost values.

The SQL processer treated the dig costs for the "Mission Jurassic Dig" differently than the other cost values because the Dig costs data is given for the "Mission Jurassic Dig" is null (has no value). Therefore, they are not considered, when we use aggregation functions such as MIN, MAX because these functions don't consider the data that are null values, In contrast, other dig names have a value for dig cost, so the value of MIN, MAX will be from these dig name's cost but not include dig costs for the "Mission Jurassic Dig" due to its null values.

4. ***Grouping Aggregate Results*** **– A museum is considering supporting their own paleontological expedition and needs to know the dig site name and cost, along with the number of dinosaur discoveries at each site. Write a single query to fulfill**

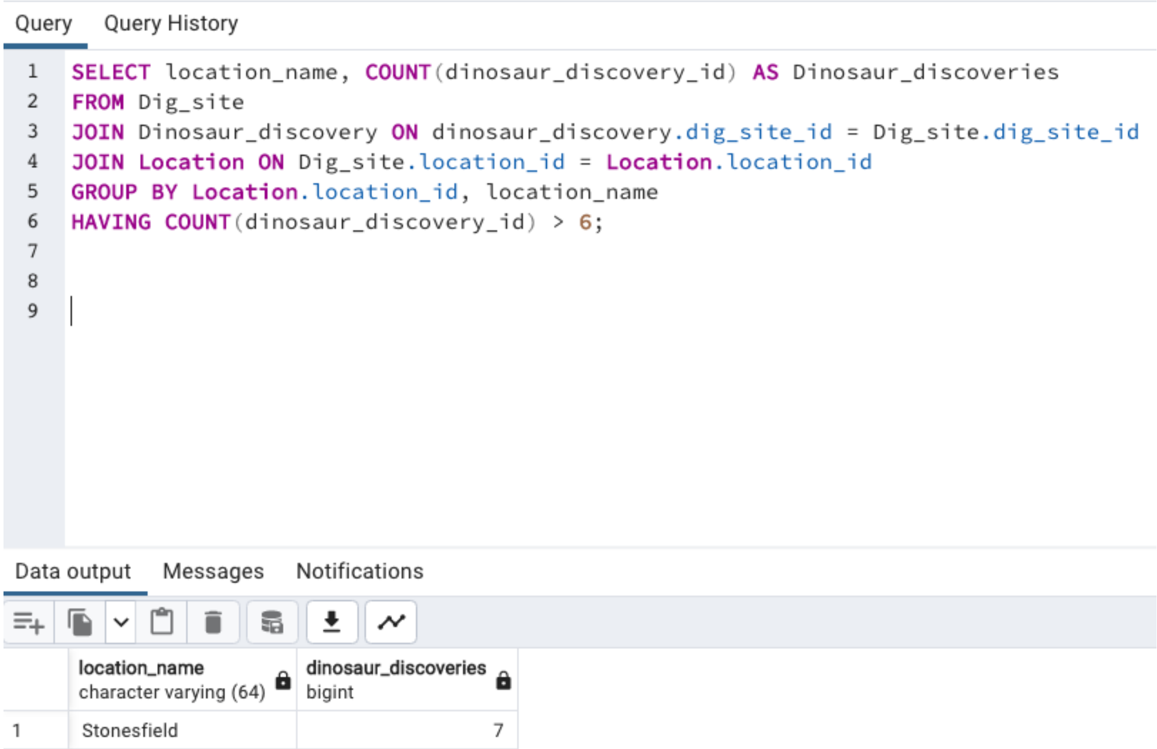
this request.



```sql
SELECT dig_name, to_char(dig_cost, '$99999.99') as dig_cost_dollar,
       COUNT(dinosaur_discovery_id) AS Dinosaur_discoveries
FROM Dig_site
JOIN Dinosaur_discovery ON dinosaur_discovery.dig_site_id = Dig_site.dig_site_id
GROUP BY dig_name, dig_cost;
```

	dig_name character varying (32)	dig_cost_dollar text	dinosaur_discoveries bigint
1	Parowan Dinosaur Tracks	$ 10000.00	3
2	Mission Jurassic Dig	[null]	2
3	Ancient Site Dig	$ 5500.00	1
4	Great British Dig	$ 8000.00	4
5	Dynamic Desert Dig	$ 3500.00	2

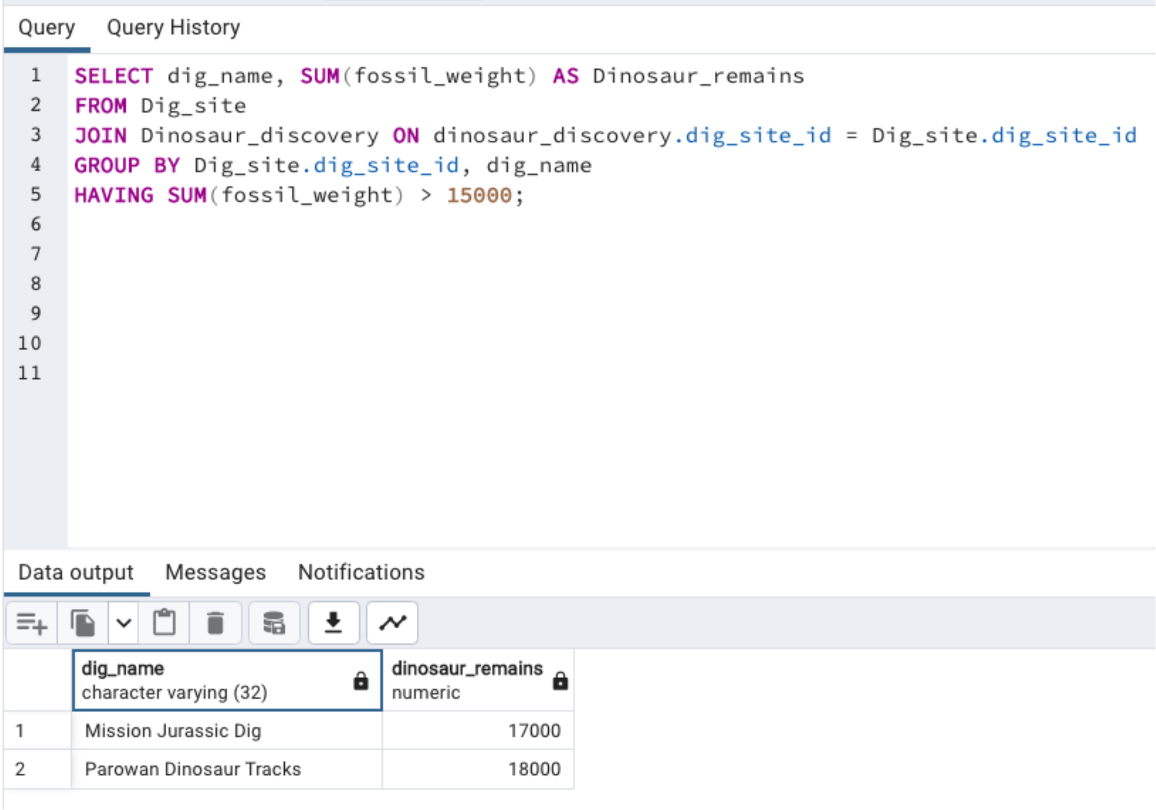
5. ***Limiting Results by Aggregation*** – **A paleontologist, looking to dig at a location ripe with discoveries, wants to search for locations with at least 6 dinosaur discoveries. Write a single query to fulfill this request.**



```sql
SELECT location_name, COUNT(dinosaur_discovery_id) AS Dinosaur_discoveries
FROM Dig_site
JOIN Dinosaur_discovery ON dinosaur_discovery.dig_site_id = Dig_site.dig_site_id
JOIN Location ON Dig_site.location_id = Location.location_id
GROUP BY Location.location_id, location_name
HAVING COUNT(dinosaur_discovery_id) > 6;
```

	location_name character varying (64)	dinosaur_discoveries bigint
1	Stonesfield	7

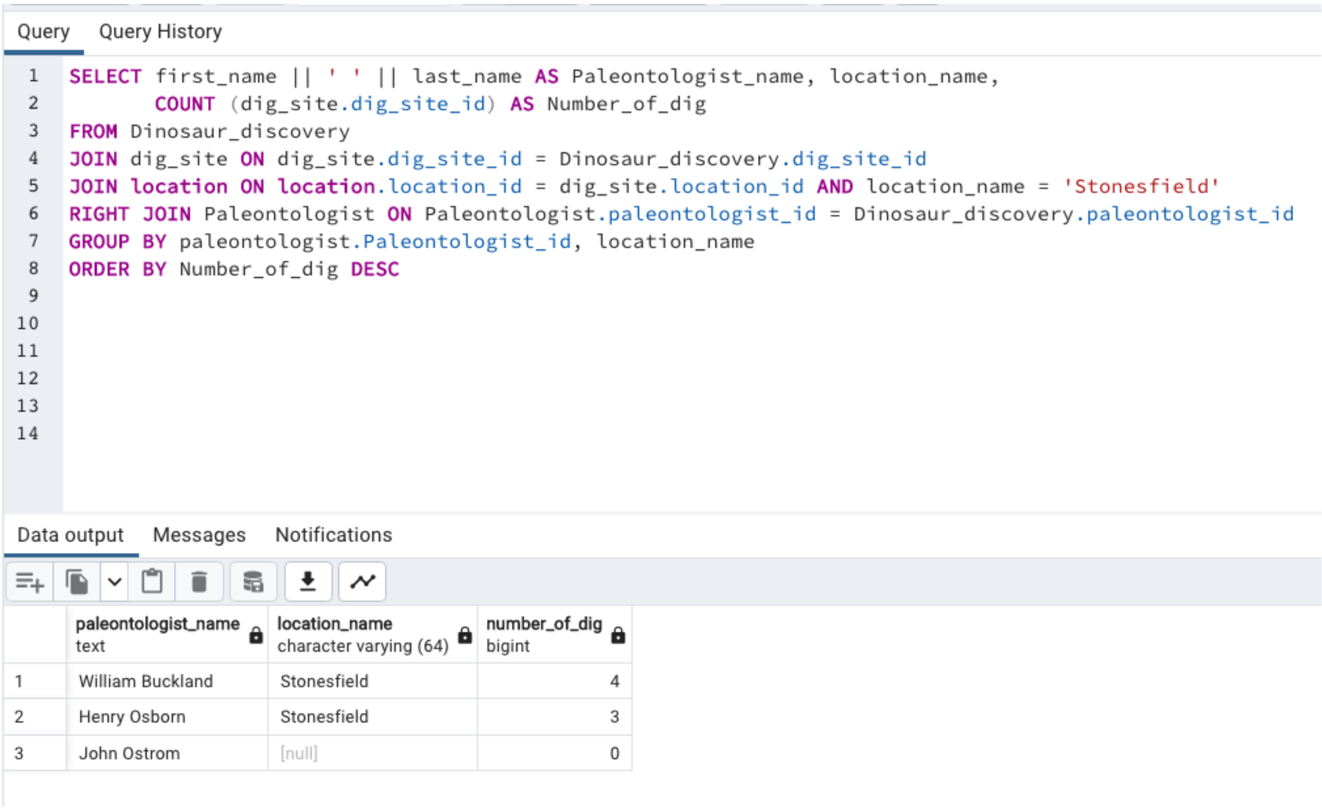
6. ***Adding Up Values* – A museum needs to know which dig sites had at least 15,000 pounds of discovered dinosaur remains. Write a single query that gives this information, with useful columns.**



```sql
SELECT dig_name, SUM(fossil_weight) AS Dinosaur_remains
FROM Dig_site
JOIN Dinosaur_discovery ON dinosaur_discovery.dig_site_id = Dig_site.dig_site_id
GROUP BY Dig_site.dig_site_id, dig_name
HAVING SUM(fossil_weight) > 15000;
```

	dig_name character varying (32)	dinosaur_remains numeric
1	Mission Jurassic Dig	17000
2	Parowan Dinosaur Tracks	18000

7. ***Integrating Aggregation with Other Constructs*** **– A research institution requests the names of all paleontologists, as well as the number of digs they participated in at the “Stonesfield” location (even if they participated in no Stonesfield digs). The institution wants the list to be ordered from most to least; the paleontologist who discovered the most Stonesfield dinosaurs will be at the top of the list, and the one with the least will be at the bottom. Write a single query that gives this information, with useful columns.**



```
SELECT first_name || ' ' || last_name AS Paleontologist_name, location_name,
       COUNT (dig_site.dig_site_id) AS Number_of_dig
FROM Dinosaur_discovery
JOIN dig_site ON dig_site.dig_site_id = Dinosaur_discovery.dig_site_id
JOIN location ON location.location_id = dig_site.location_id AND location_name = 'Stonesfield'
RIGHT JOIN Paleontologist ON Paleontologist.paleontologist_id = Dinosaur_discovery.paleontologist_id
GROUP BY paleontologist.Paleontologist_id, location_name
ORDER BY Number_of_dig DESC
```

	paleontologist_name text	location_name character varying (64)	number_of_dig bigint
1	William Buckland	Stonesfield	4
2	Henry Osborn	Stonesfield	3
3	John Ostrom	[null]	0

Section Two –Data Visualization

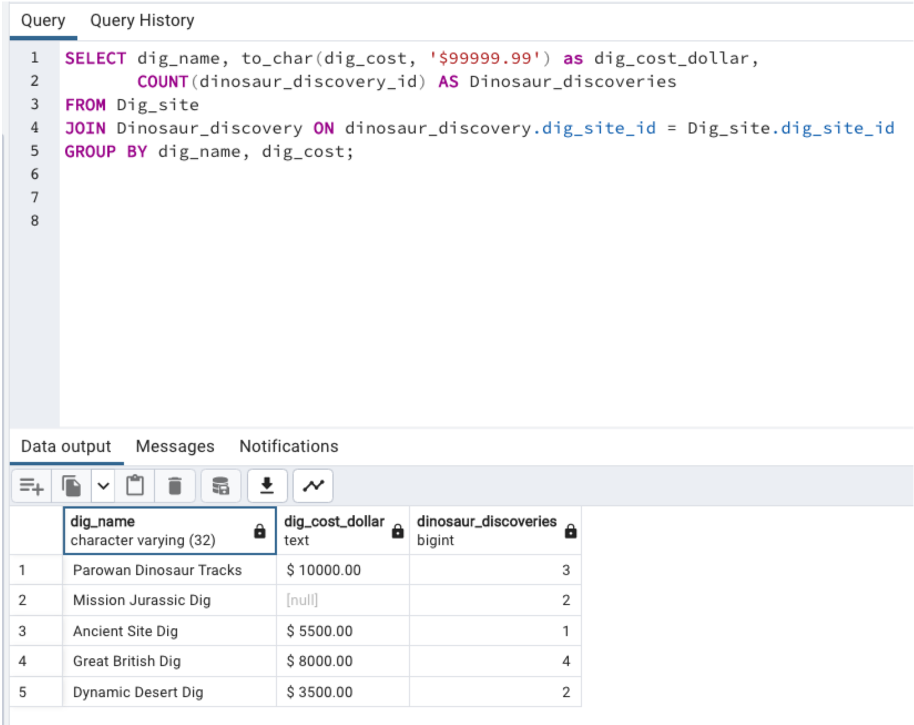
Section Background

Data visualization is presenting information in visual form, commonly with charts and graphs. People are adept at recognizing patterns, trends, and differences visually. Visual data stories are understood accurately and quickly; recognition comes much more slowly with pages and pages of text and tables.

In the modern age of data driven decision-making, data stories are important for any field – sales, finance, human resources, engineering, information technology, just to name a few. Conveying those data stories effectively is just as important. If you can design and implement effective databases, and also build visualizations from your database to tell data stories, you will have a skillset desired by organizations worldwide. In this section, you have a chance to visualize data by writing queries to obtain results, and using those results to create commonly used charts.

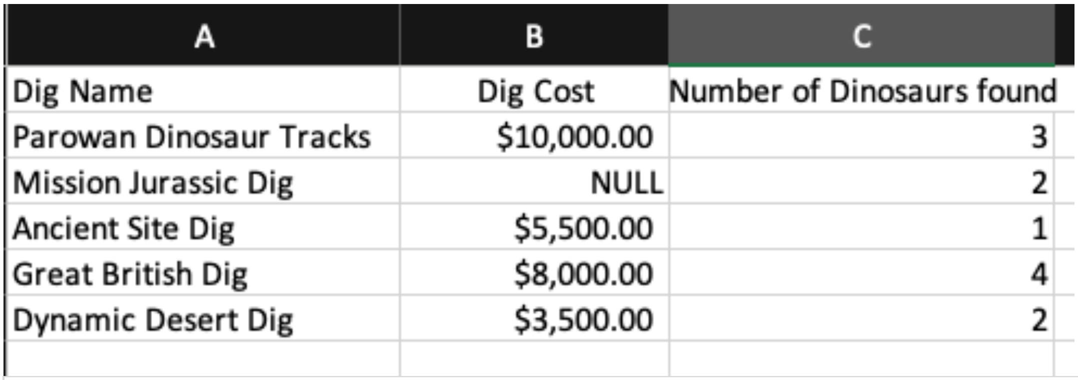
Section Steps

8. *Visualizing Data with One or Two Measures* – Use the SQL results obtained for Step #4 to address the following.
   a. **Create a bar chart with the dig name as one axis, and the dig cost as another axis.**

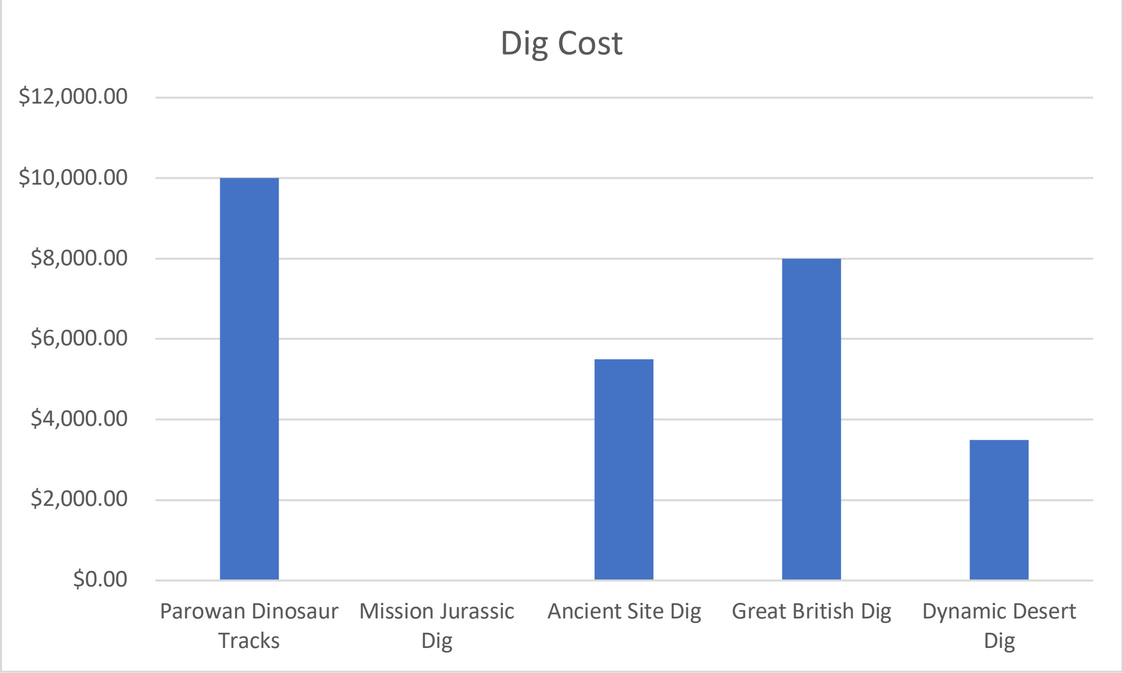


```
SELECT dig_name, to_char(dig_cost, '$99999.99') as dig_cost_dollar,
       COUNT(dinosaur_discovery_id) AS Dinosaur_discoveries
FROM Dig_site
JOIN Dinosaur_discovery ON dinosaur_discovery.dig_site_id = Dig_site.dig_site_id
GROUP BY dig_name, dig_cost;
```

	dig_name character varying (32)	dig_cost_dollar text	dinosaur_discoveries bigint
1	Parowan Dinosaur Tracks	$ 10000.00	3
2	Mission Jurassic Dig	[null]	2
3	Ancient Site Dig	$ 5500.00	1
4	Great British Dig	$ 8000.00	4
5	Dynamic Desert Dig	$ 3500.00	2



A	B	C
Dig Name	Dig Cost	Number of Dinosaurs found
Parowan Dinosaur Tracks	$10,000.00	3
Mission Jurassic Dig	NULL	2
Ancient Site Dig	$5,500.00	1
Great British Dig	$8,000.00	4
Dynamic Desert Dig	$3,500.00	2

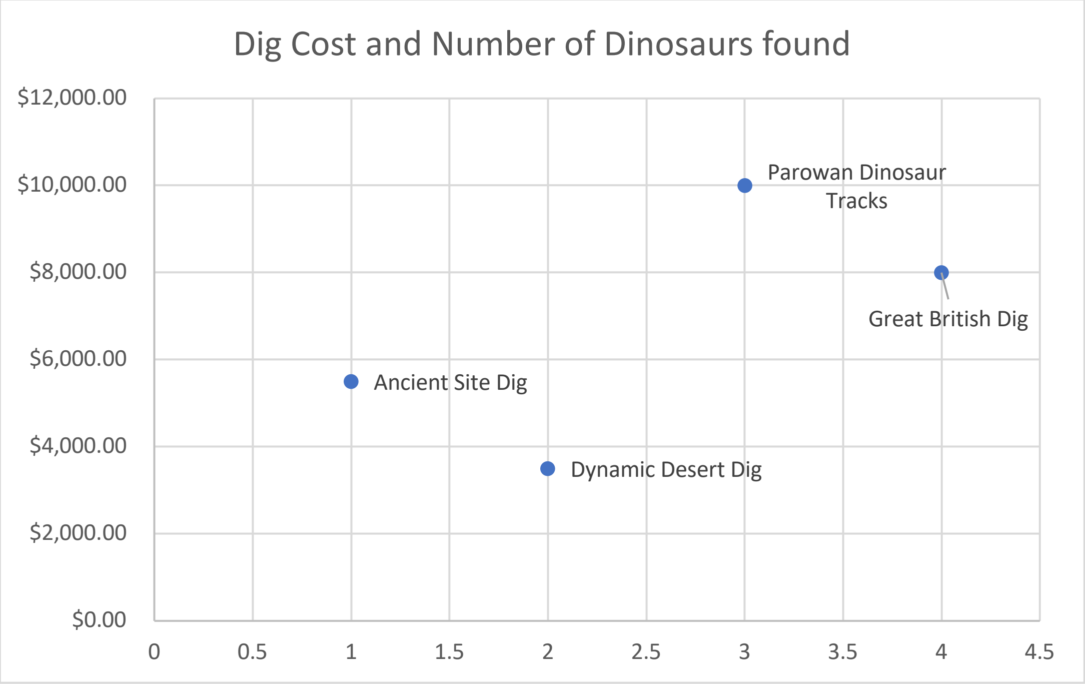


Explain the story this visualization describes.

The bar chart illustrates the amount of money spent on dinosaur digs on six dig sites such as Parowan Dinosaur, Mission Jurassic Dig, Ancient Site Dig, Great British Dig, Dynamic Desert Dig.

Although, we do not have any data for Mission Jurassic Dig cost, we still can recognize that different dig site has different dinosaur cost. Overall, Parowan Dinosaur Tracks has the highest dig cost with $10,000, The lowest dig cost is Dynamic Desert Dig about $3,500. Meanwhile, the dig cost for Great British Dig and Ancient Site Dig is $8,000 and $ 5,500 respectively.

b. **Create a scatterplot with the dig cost as one axis, and the number of dinosaurs found as another axis. Ensure that each dig name is labeled with its name, either directly or with a legend.**

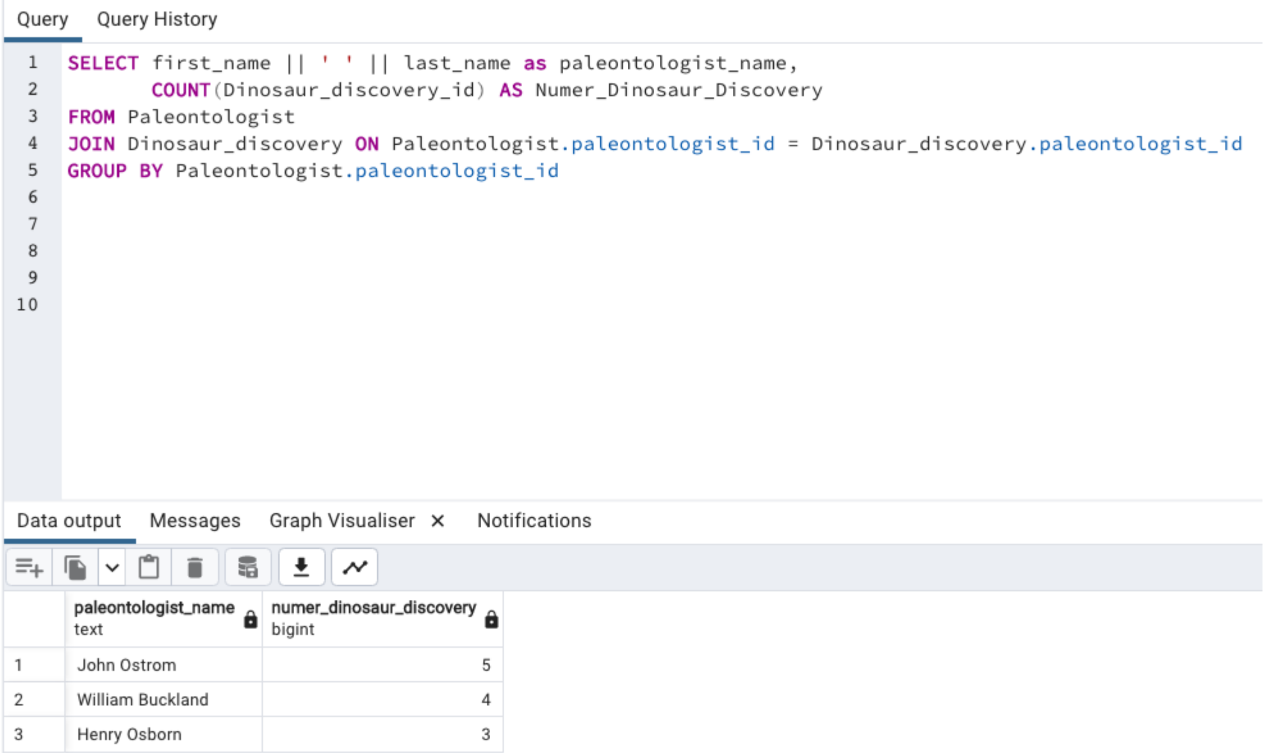


Explain the story this visualization describes.

The number of dinosaur discovery is displayed the X axis, the dig cost in the Y axis, and the name of dig site are shown as labels in scatterplot.

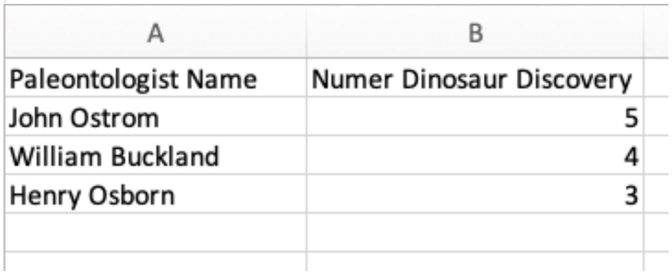
From the visual glance, we can see that Parowan Dinosaur Tracks has 3 dinosaur discoveries and dig cost is the highest $10,000. Dynamic Desert Dig has the lowest dig cost $3,500 and 3 dinosaur discoveries. Great British Dig cost is $8,000, it's not the highest dig cost but it has the highest dinosaur discoveries, 4 dinosaur discoveries. Ancient Site Dig cost is $5,500 and it has the lowest dinosaur discoveries, 1 dinosaur discoveries.

9. *Another Data Visualization* – Create a visualization of your choosing for data in the Dinosaur schema. The visualization should tell a useful story. If you find that you need more dinosaurs in the schema to tell the story well, feel free to add them. Make sure to explain the data story, and to explain why you chose that particular chart or visualization.

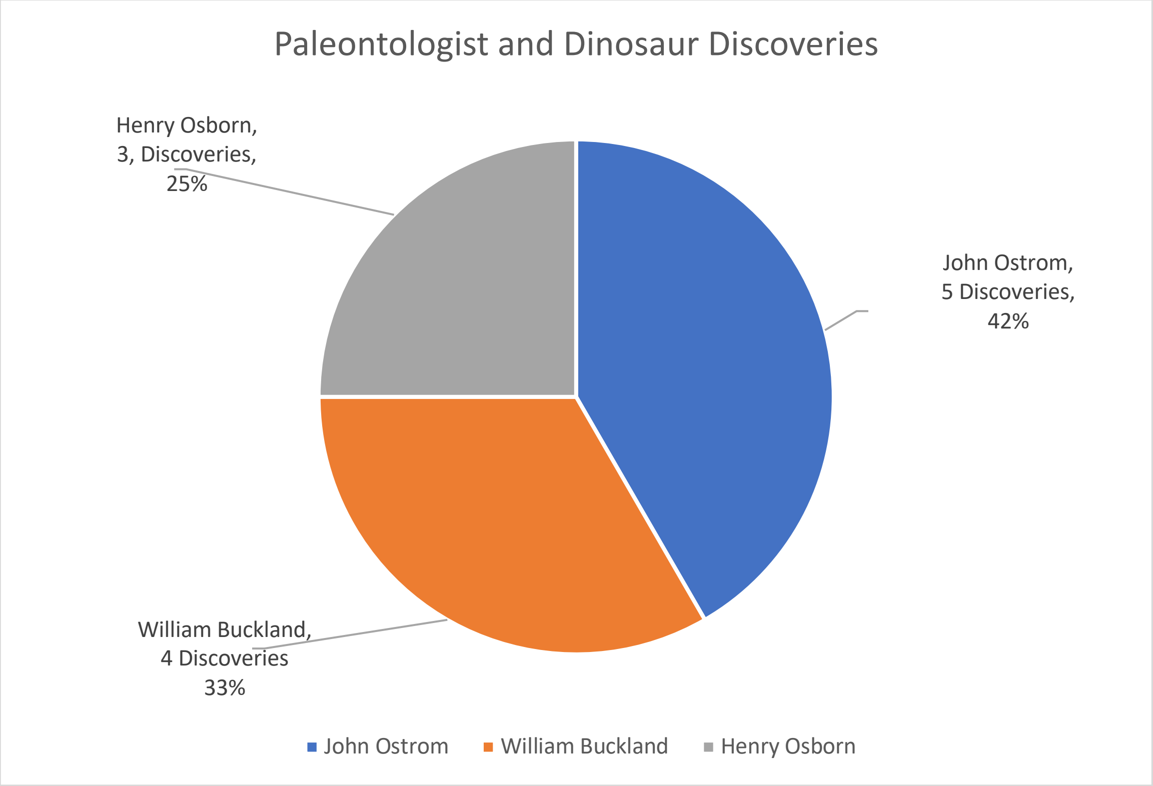


```sql
SELECT first_name || ' ' || last_name as paleontologist_name,
       COUNT(Dinosaur_discovery_id) AS Numer_Dinosaur_Discovery
FROM Paleontologist
JOIN Dinosaur_discovery ON Paleontologist.paleontologist_id = Dinosaur_discovery.paleontologist_id
GROUP BY Paleontologist.paleontologist_id
```

	paleontologist_name text	numer_dinosaur_discovery bigint
1	John Ostrom	5
2	William Buckland	4
3	Henry Osborn	3



A	B
Paleontologist Name	Numer Dinosaur Discovery
John Ostrom	5
William Buckland	4
Henry Osborn	3



- The pie chart above gives information about number of dinosaurs has been found by paleologists. And the percentage of dinosaur discoveries of each paleontologist.
- It is noticeable that there are 12 dinosaurs' discoveries in total and three paleologists such as William Buckland, Henry Osborn and John Ostrom got involved. It is clear that John Ostrom has the highest discoveries, 5 discoveries which is 42%. The figure for William Buckland is 4 discoveries which is 33 % and Henry Osborn has the lowest number of discoveries, 3 discoveries which is 25 %.
- The reason I choose pie chart because the data is not big, and I want to display the data in a circular shaped graph to have quick visualization and Pie chart works best with few data. I also want to show the percentage of dinosaur discoveries of each paleologist compared to overall.

Evaluation

Your lab will be reviewed by your facilitator or instructor with the criteria outlined in the table below. Note that the grading process:

- involves the grader assigning an appropriate letter grade to each criterion.
- uses the following letter-to-number grade mapping – A+=100,A=96,A-=92,B+=88,B=85,B-=82,C+=88,C=85,C-=82,D=67,F=0.
- provides an overall grade for the submission based upon the grade and weight assigned to each criterion.
- allows the grader to apply additional deductions or adjustments as appropriate for the submission.
- applies equally to every student in the course.

5 points per day will be subtracted for late submissions. Submissions beyond 5 days late will not be accepted. Please contact your facilitator for any exceptions.

Criterion	A	B	C	D	F
Section 1: Quality (70%)	The results for all steps in Section 1 are complete and correct. Appropriate SQL constructs have been used for all steps, and supporting explanations are present and accurate. All screenshots in Section 1 are legible. The section is well organized. All supporting explanations are clear and understandable.	The results for most steps in Section 1 are complete and correct. The appropriate SQL constructs have been used for most steps, and supporting explanations are mostly present and accurate. Most screenshots in Section 1 are legible. The section is organized. Most supporting explanations are clear and understandable.	The results for some steps in Section 1 are complete and correct. Appropriate SQL constructs have been used for some steps, and some supporting explanations are present and accurate Some screenshots in Section 1 are legible. Some supporting explanations are clear and understandable.	The results for most steps in Section 1 are incomplete or incorrect. Appropriate SQL constructs have not been used for most steps. The screenshots in Section 1 are mostly illegible or missing. Most supporting explanations are unclear or missing. The section is disorganized.	The results for virtually all steps in Section 1 are incomplete or incorrect. Appropriate SQL constructs have not been used. Virtually all screenshots in Section 1 are illegible or missing. Virtually all supporting explanations are unclear or missing. The section is disorganized.
Section 2 Presentation (20%)	The visualizations present the SQL results entirely accurately. The visualizations are labeled well, use appropriate ranges, and are clearly understood.	The visualizations present the SQL results mostly accurately. The visualizations are labeled, use reasonable ranges, and are mostly clear.	The visualizations present the SQL results somewhat accurately. The visualizations are partly labeled, use somewhat reasonable ranges, and are somewhat clear.	The visualizations present the SQL results mostly inaccurately. The visualizations may not be labeled well, may not use reasonable ranges, and may be unclear.	The visualizations are missing, or represent the SQL results entirely inaccurately. The visualizations may not be labeled, may not use ranges, and may be entirely unclear.

Section 2 Data Stories (10%)	The data stories given are entirely clear and useful. The data stories accurately characterize the visualizations.	The data stories given are mostly clear and useful. The data stories mostly characterize the visualizations.	The data stories given are somewhat clear and useful. The data stories somewhat characterize the visualizations.	The data stories given are mostly unclear and not useful. The data stories do not characterize the visualizations well.	The data stories are missing, or are entirely unclear and not useful. The data stories do not characterize the visualizations.

Use the **Ask the Teaching Team Forum** if you have any questions regarding how to approach this lab. Make sure to include your name in the filename and submit it in the *Assignments* section of the course.