MET CS 669 Database Design and Implementation for Business
Lab 3: Aggregating Data

Section One – Aggregating Data

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.

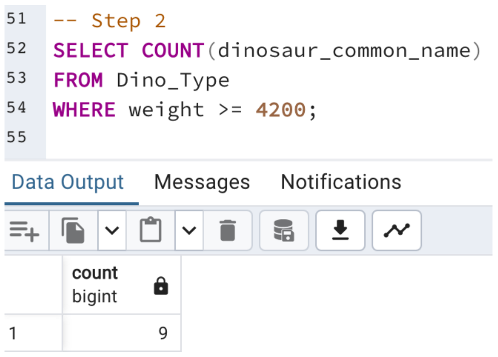
Location	**Dig Name**	**Dig Cost**	**Dinosaur Common Name**	**Weight (in pounds)**	**Paleontologist**
Stonesfield	Great British Dig	$8,000	Megalosaurus	3000	William Buckland
Stonesfield	Great British Dig	$8,000	Apatosaurus	4000	William Buckland
Stonesfield	Great British Dig	$8,000	Triceratops	4500	William 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
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).

```sql
-- Step 1
CREATE TABLE Dinos(
dinos_id DECIMAL(12) NOT NULL PRIMARY KEY,
dino_dig_location VARCHAR(64) NOT NULL);

CREATE TABLE Dino_Dig(
dino_dig_id DECIMAL(12) NOT NULL PRIMARY KEY,
dinos_id DECIMAL(12) NOT NULL,
dig_name VARCHAR(64) NOT NULL,
dig_cost DECIMAL(10,2),
paleontologist VARCHAR(64) NOT NULL,
FOREIGN KEY (dinos_id) REFERENCES Dinos(dinos_id));

CREATE TABLE Dino_Type(
dino_type_id DECIMAL(12) NOT NULL PRIMARY KEY,
dino_dig_id DECIMAL(12) NOT NULL,
dinosaur_common_name VARCHAR(128) NOT NULL,
weight DECIMAL(12) NOT NULL,
FOREIGN KEY (dino_dig_id) REFERENCES Dino_Dig(dino_dig_id));

INSERT INTO Dinos(dinos_id, dino_dig_location)
VALUES
(1, 'Stonesfield'),
(2, 'Utah'),
(3, 'Arizona');

INSERT INTO Dino_Dig(dino_dig_id, dinos_id, dig_name, dig_cost, paleontologist)
VALUES
(101, 1, 'Great British Dig', 8000, 'William Buckland'),
(102, 2, 'Parowan Dinosaur Tracks', 10000, 'John Ostrom'),
(103, 3, 'Dynamic Desert Dig', 3500, 'John Ostrom'),
(104, 1, 'Mission Jurassic Dig', NULL, 'Henry Osborn'),
(105, 1, 'Ancient Site Dig', 5500, 'Henry Osborn'),
(106, 3, 'Grand Canyon Dig', 11000, 'Katherine Rein');
```

```sql
INSERT INTO Dino_Type(dino_type_id, dino_dig_id, dinosaur_common_name, weight)
VALUES
(1001, 101, 'Megalosaurus', 3000),
(1002, 101, 'Apatosaurus', 4000),
(1003, 101, 'Triceratops', 4500),
(1004, 101, 'Stegosaurus', 3500),
(1005, 102, 'Parasaurolophus', 6000),
(1006, 102, 'Tyrannosaurus Rex', 5000),
(1007, 102, 'Velociraptor', 7000),
(1008, 103, 'Tyrannosaurus Rex', 6000),
(1009, 104, 'Spinosaurus', 8000),
(1010, 104, 'Diplodocus', 9000),
(1011, 105, 'Tyrannosaurus Rex', 7500),
(1012, 106, 'Spinosaurus', 8500);
```

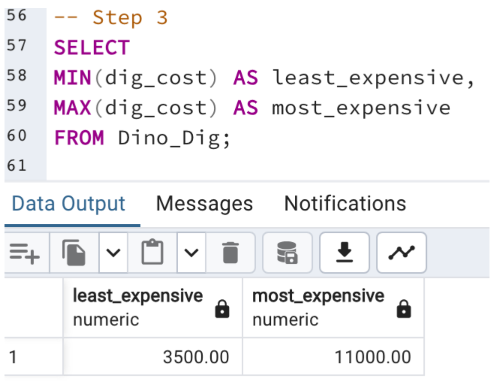
2. *Counting Matches* – A museum wants to know how many dinosaur discoveries weigh at least 4,200 pounds. Write a single query to fulfill this request.



```sql
-- Step 2
SELECT COUNT(dinosaur_common_name)
FROM Dino_Type
WHERE weight >= 4200;
```

```
   count
   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. Explain how the SQL processer treated the dig costs for the “Mission Jurassic Dig” differently than the other cost values.



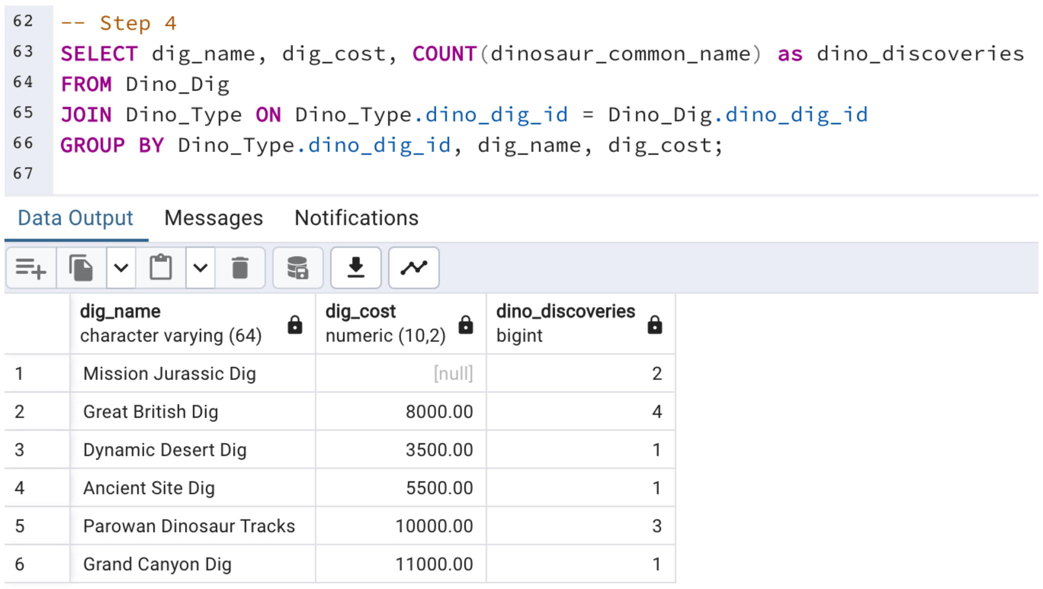
```
-- Step 3
SELECT
MIN(dig_cost) AS least_expensive,
MAX(dig_cost) AS most_expensive
FROM Dino_Dig;
```

Data Output | Messages | Notifications

	least_expensive numeric	most_expensive numeric
1	3500.00	11000.00

The SQL processor ignores null values so the cost for the “Mission Jurassic Dig” was ignored as it’s cost was input as null.

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.

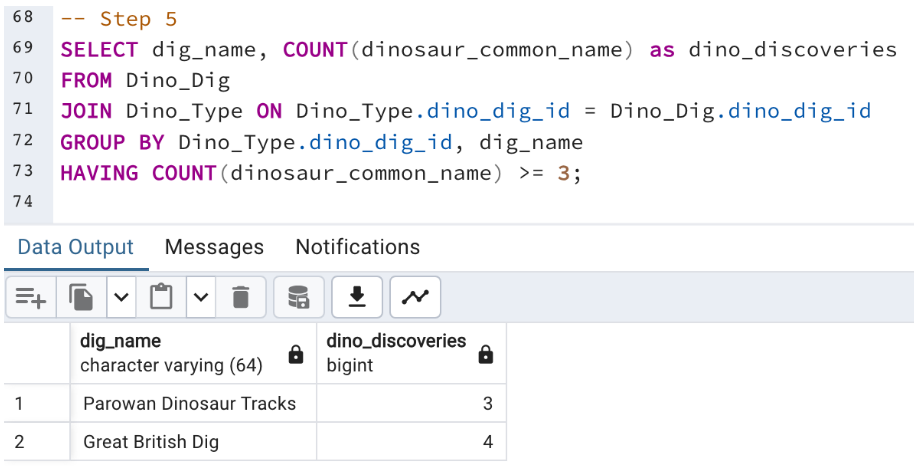


```
-- Step 4
SELECT dig_name, dig_cost, COUNT(dinosaur_common_name) as dino_discoveries
FROM Dino_Dig
JOIN Dino_Type ON Dino_Type.dino_dig_id = Dino_Dig.dino_dig_id
GROUP BY Dino_Type.dino_dig_id, dig_name, dig_cost;
```

	dig_name character varying (64)	dig_cost numeric (10,2)	dino_discoveries bigint
1	Mission Jurassic Dig	[null]	2
2	Great British Dig	8000.00	4
3	Dynamic Desert Dig	3500.00	1
4	Ancient Site Dig	5500.00	1
5	Parowan Dinosaur Tracks	10000.00	3
6	Grand Canyon Dig	11000.00	1

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.

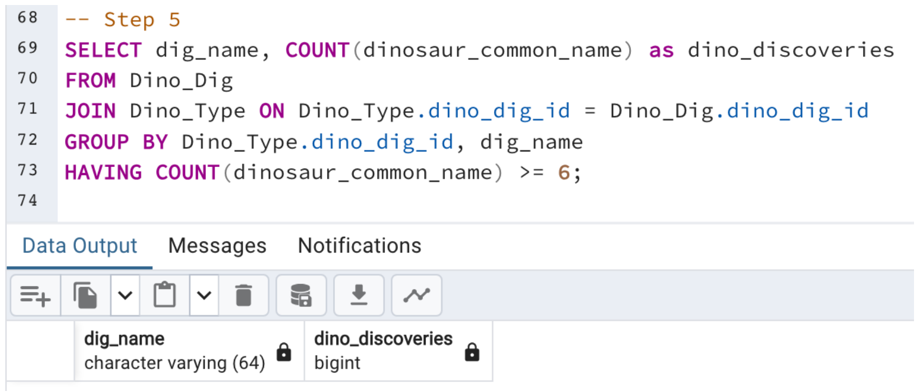
I wrote this for 3 discoveries so that I can show it works and then again for 6.



```sql
-- Step 5
SELECT dig_name, COUNT(dinosaur_common_name) as dino_discoveries
FROM Dino_Dig
JOIN Dino_Type ON Dino_Type.dino_dig_id = Dino_Dig.dino_dig_id
GROUP BY Dino_Type.dino_dig_id, dig_name
HAVING COUNT(dinosaur_common_name) >= 3;
```

Data Output Messages Notifications

	dig_name character varying (64)	dino_discoveries bigint
1	Parowan Dinosaur Tracks	3
2	Great British Dig	4

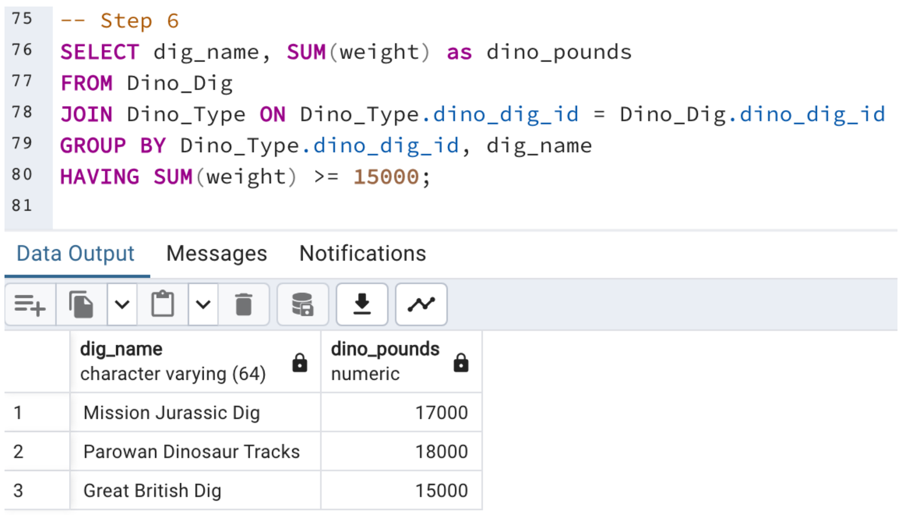


```sql
-- Step 5
SELECT dig_name, COUNT(dinosaur_common_name) as dino_discoveries
FROM Dino_Dig
JOIN Dino_Type ON Dino_Type.dino_dig_id = Dino_Dig.dino_dig_id
GROUP BY Dino_Type.dino_dig_id, dig_name
HAVING COUNT(dinosaur_common_name) >= 6;
```

Data Output Messages Notifications

	dig_name character varying (64)	dino_discoveries bigint

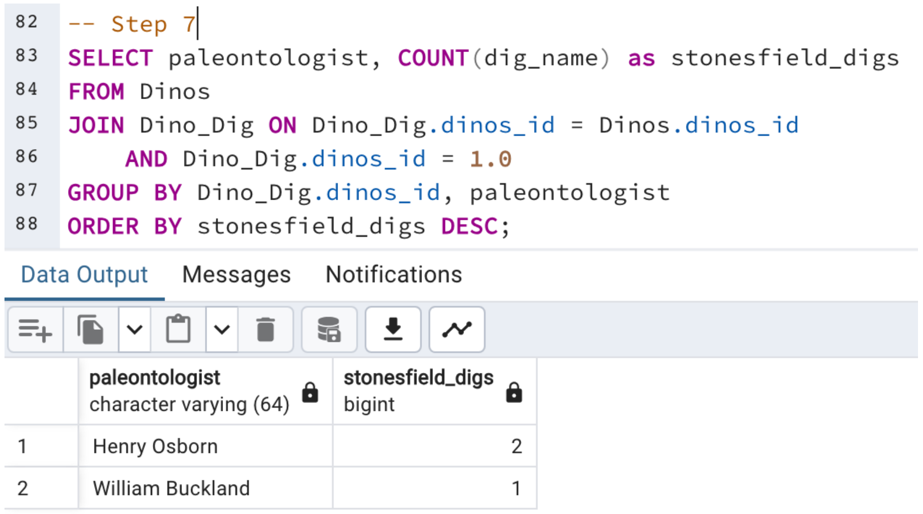
6. *Adding Up Values* – A museum needs to know which digs (not locations) had at least 15,000 pounds of discovered dinosaur remains. Write a single query that gives this information, with useful columns.



```sql
-- Step 6
SELECT dig_name, SUM(weight) as dino_pounds
FROM Dino_Dig
JOIN Dino_Type ON Dino_Type.dino_dig_id = Dino_Dig.dino_dig_id
GROUP BY Dino_Type.dino_dig_id, dig_name
HAVING SUM(weight) >= 15000;
```

	dig_name character varying (64)	dino_pounds numeric
1	Mission Jurassic Dig	17000
2	Parowan Dinosaur Tracks	18000
3	Great British Dig	15000

7. *Integrating Aggregation with Other Constructs* – A research institution wants the names of all paleontologists. For each paleontologist, the institution also wants the number of digs they participated in at the "Stonesfield" location, even if the number is 0. 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.



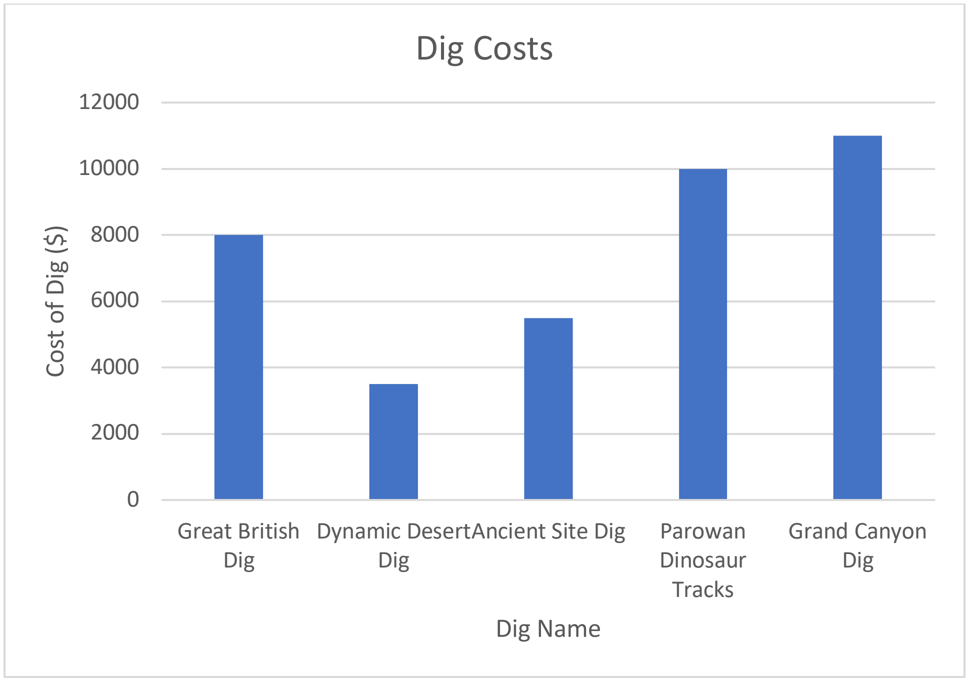
```
-- Step 7
SELECT paleontologist, COUNT(dig_name) as stonesfield_digs
FROM Dinos
JOIN Dino_Dig ON Dino_Dig.dinos_id = Dinos.dinos_id
    AND Dino_Dig.dinos_id = 1.0
GROUP BY Dino_Dig.dinos_id, paleontologist
ORDER BY stonesfield_digs DESC;
```

	paleontologist character varying (64)	stonesfield_digs bigint
1	Henry Osborn	2
2	William Buckland	1

Section Two – Data Visualization

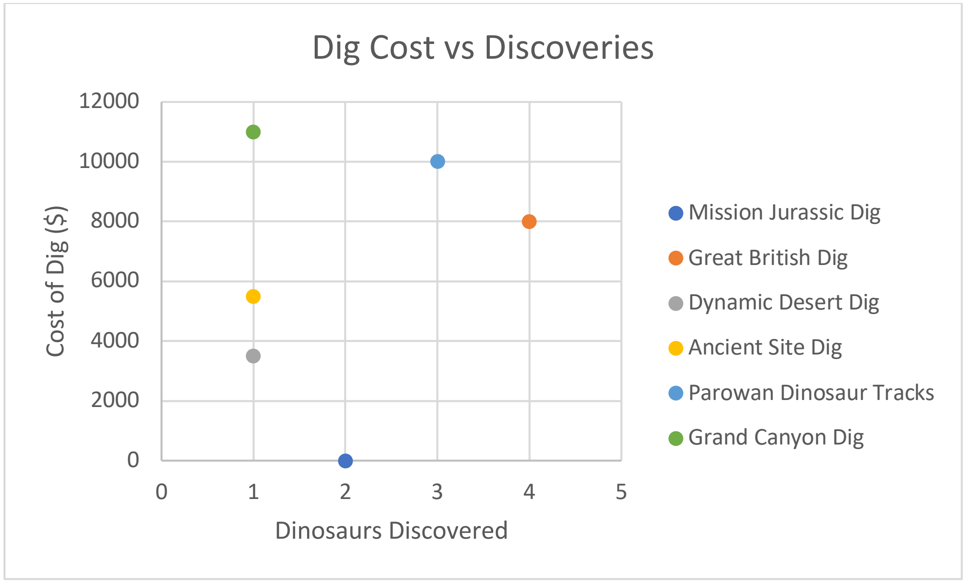
8. *Visualizing Data with One or Two Measures* – Use the SQL results obtained for Step #4 to address the following.

a. Create a bar chart with the dig name as one axis, and the dig cost as another axis. Explain the story this visualization describes.



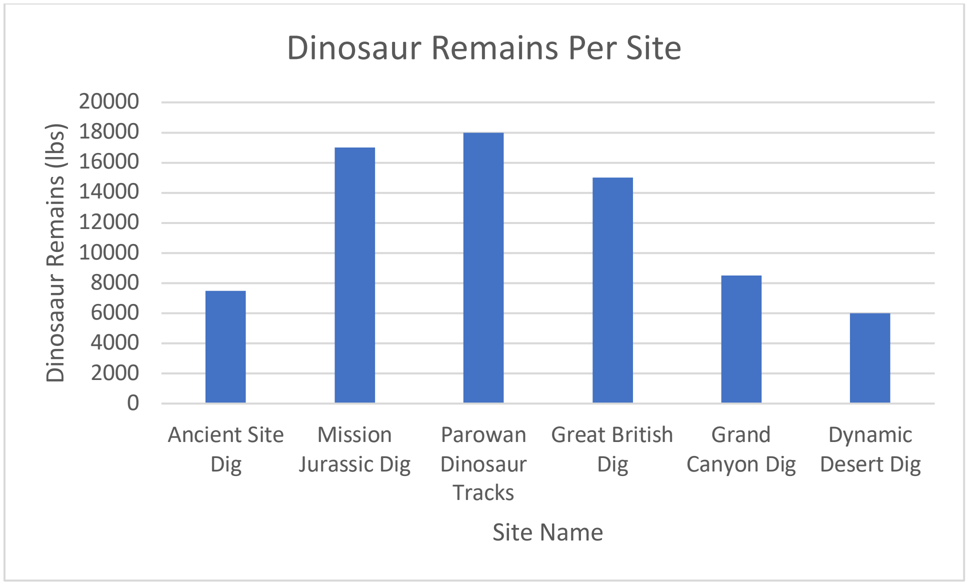
This tells us how much more expensive certain digs are than others. This emphasizes how cheap the Dynamic Desert Dig was.

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.



This shows how much more expensive digs are when you find more dinosaurs. From this data we can gather that there isn't much correlation between cost and dinosaurs found.

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.



This graph shows us how successful each site was in finding dinosaur remains. Pounds is the best measure of this because counting the dinosaurs does not show how much of the dinosaur they were able to get.