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
CREATE TABLE Locations(
    location_id DECIMAL(12) PRIMARY KEY NOT NULL,
    location_name VARCHAR(64) NOT NULL
);
CREATE TABLE Paleontologist(
    paleontologist_id DECIMAL(12) PRIMARY KEY NOT NULL,
    first_name VARCHAR(32) NOT NULL,
    last_name VARCHAR(32) NOT NULL
);
CREATE TABLE Dig_site(
    dig_site_id DECIMAL(12) PRIMARY KEY NOT NULL,
    location_id DECIMAL(12) NOT NULL,
    dig_name VARCHAR(32) NOT NULL,
    dig_cost DECIMAL(8,2) NULL,
    CONSTRAINT fk_dig_site_location
    FOREIGN KEY (location_id)
    REFERENCES Locations(location_id)
);
CREATE TABLE Dinosaur_discovery(
    dinosaur_discovery_id DECIMAL(12) PRIMARY KEY NOT NULL,
    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,
    CONSTRAINT fk_discovery_site
    FOREIGN KEY (dig_site_id)
    REFERENCES Dig_site(dig_site_id),
    CONSTRAINT fk_paleontologist_discovery
    FOREIGN KEY (paleontologist_id)
    REFERENCES Paleontologist(paleontologist_id)
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 127 msec.
```

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

INSERT INTO Paleontologist(paleontologist_id, first_name, last_name)
VALUES(1, 'William', 'Buckland');
INSERT INTO Paleontologist(paleontologist_id, first_name, last_name)
VALUES(2, 'John', 'Ostrom');
INSERT INTO Paleontologist(paleontologist_id, first_name, last_name)
VALUES(3, 'Henry', 'Osborn');

INSERT INTO Dig_site(dig_site_id, location_id, dig_name, dig_cost)
VALUES(1, 1, 'Great British Dig', 8000);
INSERT INTO Dig_site(dig_site_id, location_id, dig_name, dig_cost)
VALUES(2, 2, 'Parowan Dinosaur Tracks', 10000);
INSERT INTO Dig_site(dig_site_id, location_id, dig_name, dig_cost)
VALUES(3, 3, 'Dynamic Desert Dig', 3500);
INSERT INTO Dig_site(dig_site_id, location_id, dig_name, dig_cost)
VALUES(4, 1, 'Mission Jurassic Dig', NULL);
INSERT INTO Dig_site(dig_site_id, location_id, dig_name, dig_cost)
VALUES(5, 1, 'Ancient Site Dig', 5500);

INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(1, 1, 1, 'Megalosaurus', 3000);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(2, 1, 1, 'Apatosaurus', 4000);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(3, 1, 1, 'Triceratops', 4500);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(4, 1, 1, 'Stegosaurus', 3500);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(5, 2, 2, 'Parasaurolophos', 6000);
```

Data Output | Messages | Notifications

```
INSERT 0 1

Query returned successfully in 89 msec.
```

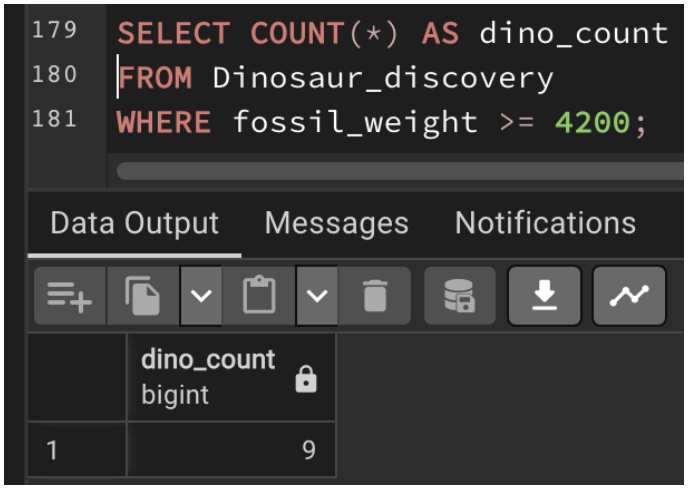
```
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(5, 2, 2, 'Parasaurolophos', 6000);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(6, 2, 2, 'Tyrannosaurus Rex', 5000);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(7, 2, 2, 'Velociraptor', 7000);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(8, 3, 2, 'Tyrannosaurus Rex', 6000);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(9, 4, 3, 'Spinosaurus', 8000);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(10, 4, 3, 'Diplodocus', 9000);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(11, 5, 3, 'Tyrannosaurus Rex', 7500);
INSERT INTO Dinosaur_discovery(dinosaur_discovery_id, dig_site_id, paleontologist_id, common_name, fossil_weight)
VALUES(12, 3, 2, 'Abrosaurus', 4500);
```

Data Output | Messages | Notifications

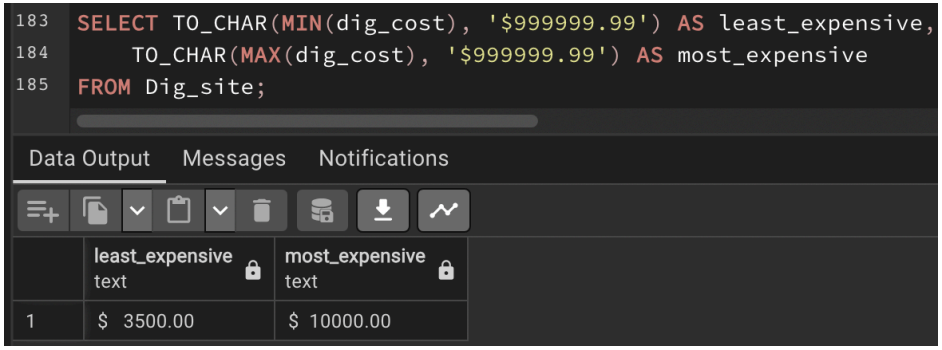
```
INSERT 0 1

Query returned successfully in 99 msec.
```

The last insert statement is the additional dinosaur discovery I added.

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.



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.



Because the Mission Jurassic Dig has no cost associated with it, the SQL processor ignores it in the case of the MIN and MAX functions. Each DBMS has different ways of treating null values in a table, and aggregate functions vary from ignoring null values, putting them at the beginning or end of results, or allowing the user to determine how they are treated.

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.

```
187 SELECT dig_name, dig_cost, COUNT(*) AS discovery_count
188 FROM Dig_site
189 JOIN Dinosaur_discovery ON Dinosaur_discovery.dig_site_id = Dig_site.dig_site_id
190 GROUP BY dig_name, dig_cost;
```

Data Output | Messages | Notifications

	dig_name character varying (32)	dig_cost numeric (8,2)	discovery_count 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.

```
192 SELECT location_name, COUNT(*) AS discovery_count
193 FROM Locations
194 JOIN Dig_site ON Dig_site.location_id = Locations.location_id
195 JOIN Dinosaur_discovery on Dinosaur_discovery.dig_site_id = Dig_site.dig_site_id
196 GROUP BY location_name
197 HAVING COUNT(*) >= 6;
198
```

Data Output | Messages | Notifications

	location_name character varying (64)	discovery_count bigint
1	Stonesfield	7

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.

```
199 SELECT dig_name, SUM(fossil_weight) AS total_weight
200 FROM Dig_site
201 JOIN Dinosaur_discovery ON Dinosaur_discovery.dig_site_id = Dig_site.dig_site_id
202 GROUP BY dig_name
203 HAVING SUM(fossil_weight) >= 15000;
204
```

Data Output | Messages | Notifications

	dig_name character varying (32)	total_weight numeric
1	Great British Dig	15000
2	Parowan Dinosaur Tracks	18000
3	Mission Jurassic Dig	17000

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.

```
205 SELECT first_name ||' '|| last_name AS paleontologist_name, COUNT(Locations.location_id) AS number_digs
206 FROM Dinosaur_discovery
207 JOIN Paleontologist ON Paleontologist.paleontologist_id = Dinosaur_discovery.paleontologist_id
208 JOIN Dig_site ON Dig_site.dig_site_id = Dinosaur_discovery.dig_site_id
209 LEFT JOIN Locations on Locations.location_id = Dig_site.location_id
210     AND Locations.location_id = 1
211 GROUP BY first_name ||' '|| last_name
212 ORDER BY number_digs DESC;
```

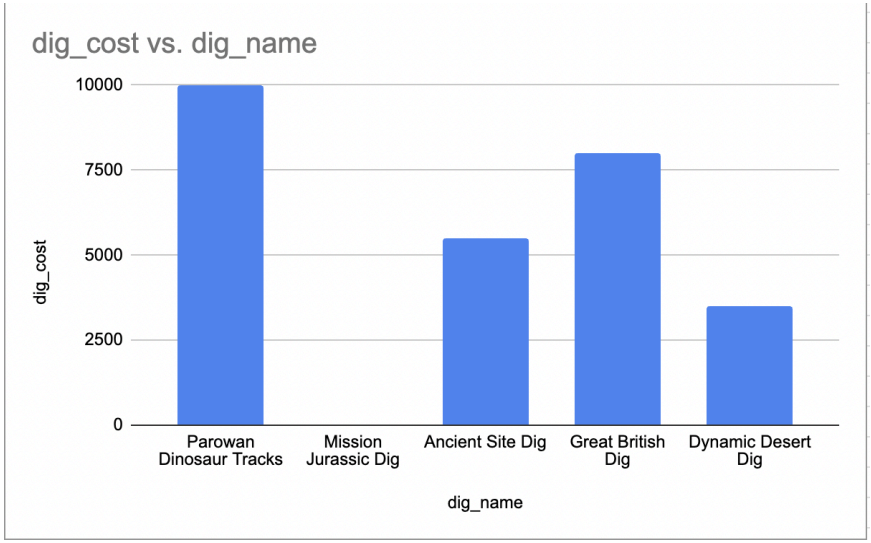
Data Output | Messages | Notifications

	paleontologist_name text	number_digs bigint
1	William Buckland	4
2	Henry Osborn	3
3	John Ostrom	0

Section Two – Data Visualization

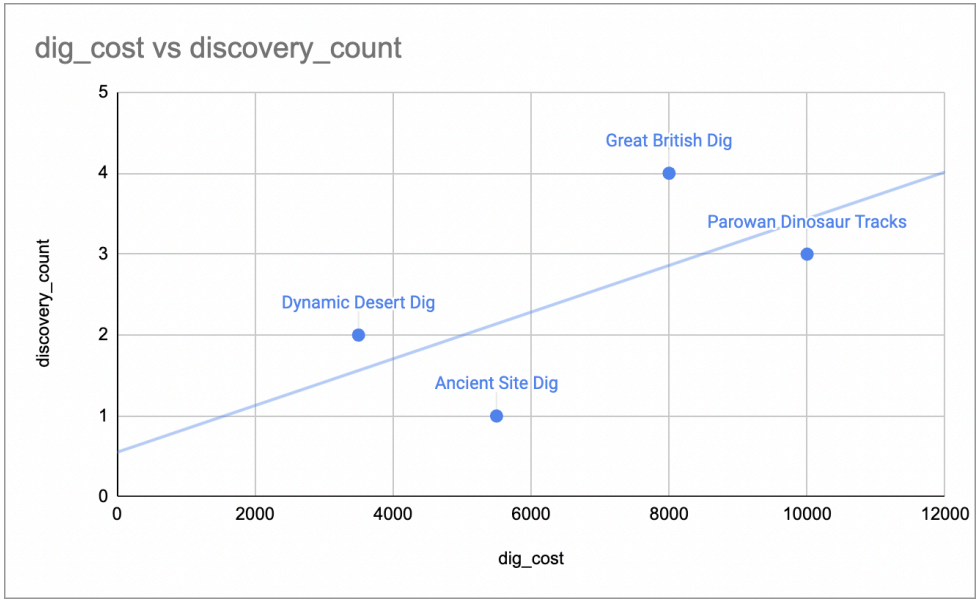
8. *Visualizing Data with One or Two Measures* – Use the SQL results obtained for Step #4 to address the following.

   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.



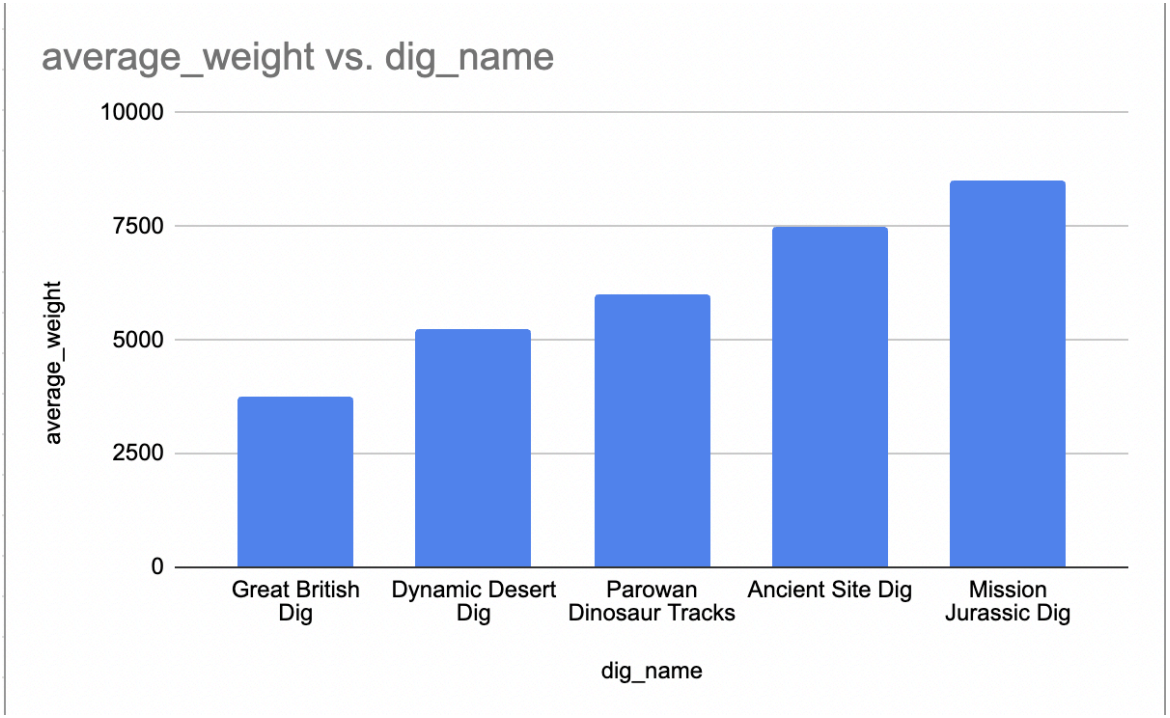
   Different dig sites have different costs associated with them, and can vary by a lot. Parowan Dinosaur tracks is the most expensive dig site, while the Dynamic Desert Dig is the least expensive. The cost for the Mission Jurassic Dig is not mentioned in the data.

   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.



There is a correlation between the cost of a dig site and the number of discoveries made at the site. The Great British Dig and the Parowan Dinosaur Tracks cost the most and have a greater number of discoveries made compared to the Dynamic Desert Dig and the Ancient Dig Site, which were less expensive and had less discoveries made.

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.



Certain dig sites had heavier dinosaur discoveries made compared to others. At the Mission Jurassic Dig site, the average fossil weight of the dinosaur remains was the greatest, while the average fossil weight at the Great British Dig was the lowest. It can be inferred that smaller dinosaurs inhabited the area of the Great British Dig, while larger ones roamed where the Mission Jurassic Dig site is. I chose a bar graph to represent this data because it made the most sense and displayed information for both axes in a clear manner.