|  |
| --- |
| **Overview of the Lab** |
| Sometimes we are interested in the result of aggregating multiple data items rather than in individual data items. For example, a store may be interested in the monetary amount of a single sale, but may be equally or more interested in the sum the monetary amount of all sales that occurred on a specific day. SQL provides many useful ways to aggregate data. One objective of this lab is for you to learn to aggregate data using SQL.  Other objectives include learning to normalize a schema’s tables to BCNF, and learning to visualize SQL results with data visualizations.  From a technical perspective, together, we will learn:   * how to use aggregate functions generally. * how to count and add items in a table. * how to determine minimum and maximum values. * how to filter rows based upon aggregate values. * how to use aggregation with joins together to answer more complex use cases with related data. * how to visualize SQL results with basic data visualizations. |

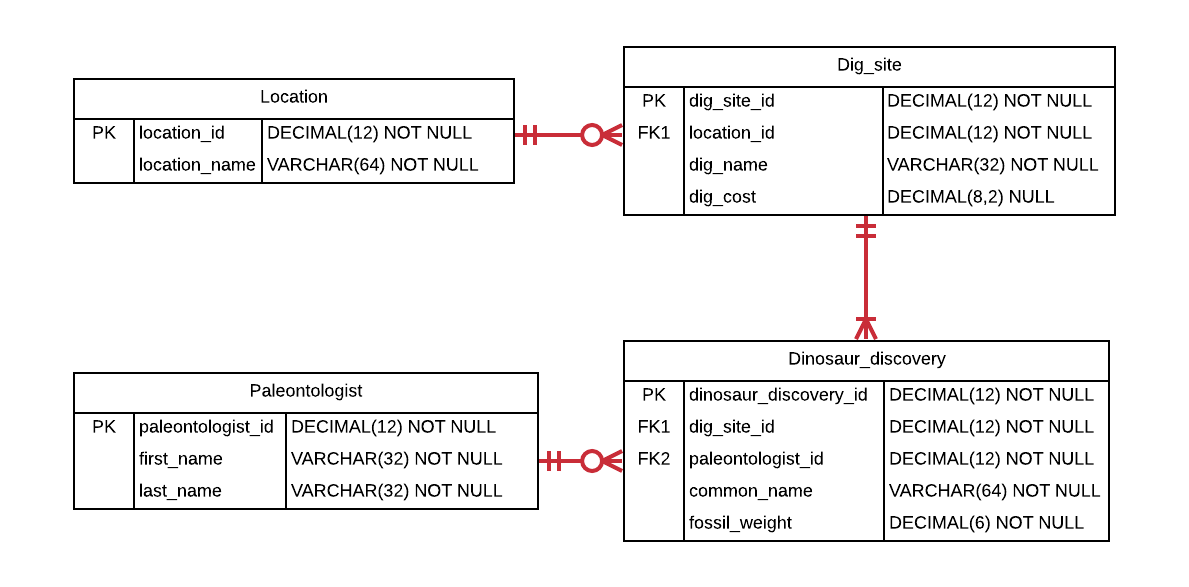
|  |
| --- |
| **Lab 3 Explanations Reminder** |
| As a reminder, it is important to read through the Lab 3 Explanation document to successfully complete this lab, available in the assignment inbox alongside this lab. The explanation document illustrates how to correctly execute each SQL construct step-by-step, and explains important theoretical and practical details. |

|  |
| --- |
| **Other Reminders** |
| * The examples in this lab will execute in modern versions of Oracle, Microsoft SQL Server, and PostgreSQL as is. * The screenshots in this lab display execution of SQL in the default SQL clients supported in the course – Oracle SQL Developer, SQL Server Management Studio, and pgAdmin – but your screenshots may vary somewhat as different version of these clients are released. * Don’t forget to commit your changes if you work on the lab in different sittings, using the “COMMIT” command, so that you do not lose your work. |

**Section One – Aggregating Data**

**Section Background**

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



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.

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 |
| 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 |
| New Mexico | Mexican dig | $12,000 | Northronychus | 6000 | Jim Kirkland |

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

Note: row with green is new added instance.

Creating 4 tables:



Location table after inserting values:

Graphical user interface, application

Description automatically generated

Dig\_site table after inserting values:

Graphical user interface, application, table

Description automatically generated

Paleontologist table after inserting values:

Graphical user interface, application

Description automatically generated

Dinosaur discovery table after inserting values:

Table

Description automatically generated

And full table after joining all four tables:

Graphical user interface, application, table

Description automatically generated

1. *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.

Answer: 9 discoveries

Graphical user interface, application

Description automatically generated

1. *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.

Answer:

Graphical user interface, text, application

Description automatically generated

As there is missing value for the Mission Jurassic Dig, SQL processer disregard the two null values present in the dig\_cost column. Also, dig\_cost should be a numeric value as per data types but there is no value for Mission Jurassic dig. DBMS ignores null values. As we can see from our max and min queries, DBMS look for minimum and maximum values of dig cost and ignore null values.

1. *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.

*Answer:*

Graphical user interface, application, Teams

Description automatically generated

1. *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.

Answer: Stonesfield with 7 discoveries

Graphical user interface, application

Description automatically generated

1. *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.

*Answer:*

Graphical user interface, application

Description automatically generated

1. *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.

Answer:

Graphical user interface, text, application, email

Description automatically generated

**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**

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

Answer: - This bar visualization method is easy to understand. As there is null value for “mission Jurassic dig” I removed it for the visualization. I made the graph in decreasing order of dig cost.From graph it easy to see that cost for Mexican dig is highest which is $12,000 and cost for dynamic desert dig lowest which is $3,500. Graph is given below.



* 1. 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.

Answer:

It is easy to visualize and describe just looking into this scatter plot. From the plot, there are three types of dig name namely Mexican dig, ancient site digs and dynamic desert dig each of which were able to discover one type of dinosaur. Similarly, Parowan dinosaur Tracks were able to discover 3 types of dinosaurs and Great British digs were able to discover four different types of dinosaurs. Also, from same graph it can be easily visualize that cost for Mexican dig is highest which is $12,000 and cost of dynamic desert dig is lowest which is $3,500.



1. *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.

Answer: -

Data for my visualization was extracted using this command from database.

Graphical user interface, application

Description automatically generated



The reason I choose this graph versus other is I wanted to compare fossil weights of different dinosaurs and the location of their discovery. There are three types of TyrannnosaurusRex which is differentiated among them by their weight here and named them which has its end letter came from the name of location they were discovered. For example, TyrannnosaurusRex\_U was discovered in Utah. There might be different features for each of these dinosaurs, but we can simply differentiate among them by their weight. For example, diplodocus’s fossil is heaviest among them while Megalosaurus’s fossils are the lightest. Also, three of the fossils namely Parasaurolophus, TyrannosaurusRex\_A and Northronychus have same weight that is 600 pounds. I arranged the dinosaurs in graph in decreasing order of their fossil’s weight. Second heaviest fossil is of Spinosaurus which has 8000 pounds of weight. The graph also gives insight of the location of its discovery for example first three heaviest fossils namely Diplodocus, Spinosaurus and Tyrannosaurus was discovered in Stonesfield. Similarly, light fossils named Megalosaurus and stegosaurus also discovered in Stonesfield.

# 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.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criterion** | **What it Measures** | **A+ Excellent** | **B Good** | **C Fair/Satisfactory** | **D Insufficient** | **F Failure** |
| **Section 1 Technical Mastery (50%)** | For section 1, this measures the correctness of the SQL results and the quality of the construction of the SQL queries. Excellent queries provide complete and correct results, and are constructed with suitable, relevant SQL constructs. | Entirely correct Excellent construction | Mostly correct Good construction | Somewhat correct Satisfactory construction | Mostly incorrect Insufficient construction | Queries and results missing *or* Entirely incorrect Unacceptable construction |
| **Section 1 Intelligibility (20%)** | This measures how intelligible your section 1 solutions are. Excellent solutions have entirely legible screenshots with excellent supporting explanations (where needed) that are organized well and clearly presented. | Excellent intelligibility | Good intelligibility | Satisfactory intelligibility | Insufficient intelligibility | Section 1 missing *or* Unacceptable intelligibility |
| **Section 2 Visualizations (20%)** | For section 2, this measures the accuracy and clarity of the data visualizations. Excellent data visualizations present the SQL results entirely accurately, are labeled well, use appropriate ranges, are legible and organized, and are clearly understood. | Entirely accurate Entirely clear | Mostly accurate Mostly clear | Somewhat accurate Somewhat clear | Mostly inaccurate Mostly unclear | The data visualizations are missing  *or* Entirely inaccurate Entirely unclear |
| **Section 2 Data Stories (10%)** | This measures the accuracy and clarity of the data stories. Excellent data stores are entirely clear and useful, are organized well, and accurately describe the data and visualizations. | Entirely accurate Entirely clear | Mostly accurate Mostly clear | Somewhat accurate Somewhat clear | Mostly inaccurate Mostly unclear | The data stories are missing  *or* Entirely inaccurate Entirely unclear |
|  |  | **Preliminary Grade:** |  | **Lateness Deduction:** 5 points per day4 days maximumContact your facilitator for any exceptions |  | **Lab Grade:** |

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.