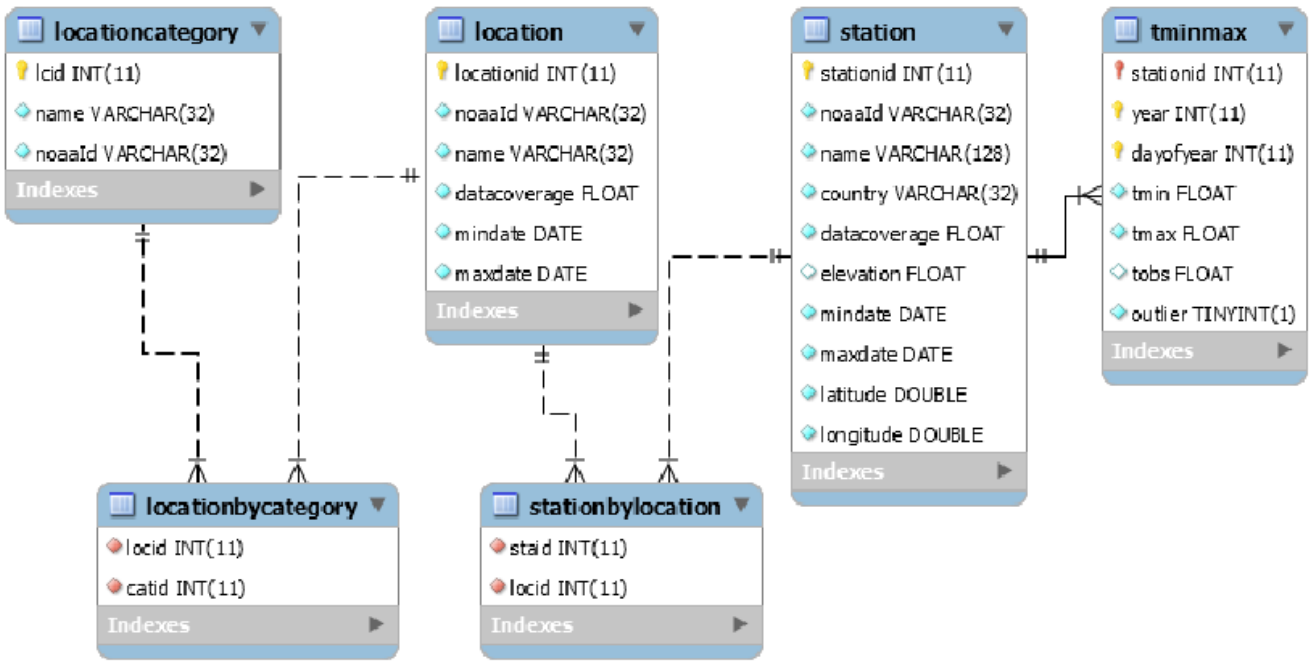
Mysql Assignment.

Data Analysis using SQL

The **noaacdo** database is a small subset of the vast database collected by NOAA through international collaborations going back to 1824. The entire database is accessible via aweb-service api. Since web-service access can be slow and does not support SQL querying, I have accumulated some of this data into **noaacdo** so that it can be more easily analyzed. The schema is shown here:



This database can be found on the remote server that we connected to in class. In case you have lost that connection, the hostname is 10.0.0.16, the user name is *dbreader*, and the password is *stiletto*.
The following problems ask you to exercise your skills in crafting SELECT statements to investigate trends in the data and generate output conforming to specific requests. As you craft your queries, please be aware that these tables are large. The tminmax table has over 200 million rows, and most other tables have over 100,000 rows. Given that the server hosting this database is not especially powerful, queries which utilize all rows of tminmax can be quite slow. For example, the query "select count(*) from tminmax" takes ~108 seconds to return.
To improve performance, the tminmax table is partitioned into 10 time ranges, each with an approximately equal number of entries. This means that queries which restrict their scope by using a year range in their where clauses can run faster since they pull data from only a subset of the full table. The queries in this exercise that pull data from tminmax all utilize a year range and should not require more the ~30 seconds to run.
The **deliverable** from this assignment is a single script file named .sql. Your scripts should appear in the same order as the problems below, and since all of the problems below are identified numbers and

sometimes letters, you should also add comments to your script file indicating which problem each query addresses. One question below asks you to complete a table and offer ideas on next steps - again, you should include these as a comment in your script file.

1. Write a query to list how many stations are found in each location. Output should list the location name and the station count, and the columns should have the headers 'Location' and '# Stations'. Only report locations with 100 or more stations, and list locations with the most stations first.

2. Write a query to list location name, the minimum elevation of its stations, the maximum elevation of its stations, and the average elevation of its stations. Include only those locations with 100 or more stations, and round the average elevation to just 1 decimal place. Locations with the highest average elevation should be listed first.

3. Note that a “location” is a geographic area, such as a city, county, state or country (Query the locationcategory table to see the full list). As a result, the location/station relationship is many to many - locations include multiple stations, and a station can be included in multiple overlapping locations. Typically the several locations in which a single station is grouped fall into different location categories.

Write a query to list the location category name, the location name, the station name, and elevation of the locations that include the one station in the entire database that has the highest elevation. Your column headers should be “Category”, “Location”, “Station”, and “Elevation”. HINT: the station and elevation will be the same for all five rows of your output.

4. You want to test a hypothesis that mean daily temperatures are highest at stations nearest the equator and at lowest elevations. As a preliminary analysis, you can calculate a few bulk statistics.

A. Write a query to report station elevation, absolute value of the latitude, and average of the mean daily temperature measured at the station. Restrict you query to the year 2008 and later. Order by elevation, and limit the query to just 50 results. NOTE: The “mean daily temperature” at a station should be calculated as (tmin + tmax)/2. Do not use TObs, as it is reported only by a small subset of the stations.

B. Write a query to report the average of each of the fields in the previous query. Write this query in 4 different ways:

1) Average over the 50 highest elevations
2) Average over the 50 lowest elevations
3) Average over the 50 lowest latitudes (remember - latitude ranges from -90 to 90, so use the absolute value)
4) Average over the 50 highest latitudes (again, use absolute values of latitudes)

Use the query results to complete this table:

Station Category	Average Elevation	Average Latitude	Average Temperature
High Elevation			
Low Elevation			
Low Latitudes			

High Latitudes

C. Do the results suggest that the hypothesis has merit? Suggest how you might be able to quantify the variation in temperature with latitude.

5. This database is not fully normalized. The station fields mindate and maxdate ideally should reflect the minimum and maximum dates for which temperature data is available in tminmax. Similarly, the location fields mindate and maxdate should reflect the minimum and maximum dates for which temperature data is available in tminmax taken over all of the stations in each location.

A. Write a query to return the number of stations for which the station's maxdate's year is less than the maximum year in tminmax for that station. Use only those entries in tminmax where year >= 2000.

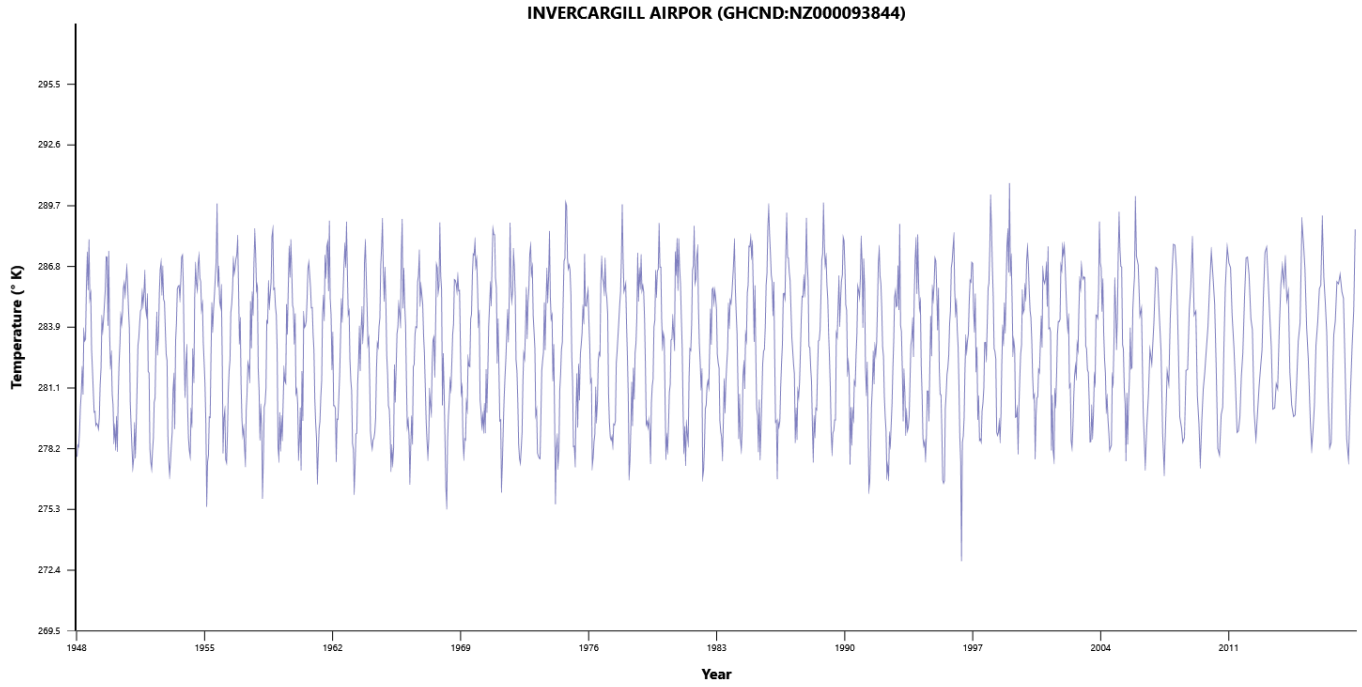
B. Write a query to return the count of locations for which the location's maxdate's year is less than the maximum year for any station in that location. Again, use only those entries in tminmax where year >= 2000.

6. If you plot daily temperature for any particular station measured over several years, you will find that it is perfectly periodic due to earth's unwavering orbit around the sun. This periodicity can be modeled with a function:

$$T(t) = T_{mean} + A*\sin(2*\pi*(t-\phi))$$

where t is time measured in days, T_{mean} is the yearly mean temperature, A is the amplitude of the seasonal fluctution, and ϕ is a phase offset representing seasonal "lag" - i.e., the difference between the date of the winter solstice and the day at which lowest temperature is typically observed.

Daily Temperatures at Invercargill Aiport (New Zealand) from 1948-present



Finding the values for T_{mean}, A and ϕ which best fit to the data can be done by making educated guesses for their values, then systematically tweaking the values up or down until the root mean square deviation between the data and the model is minimized.

For the station data shown above (stationid=1115), write the following queries:

A. Write a query to estimate both T_{mean} and A.

B. Estimating ϕ requires two steps:

1. Write a query to report the mean daily temperature averaged over the years 2008 to present.

2. Using the above in a sub query, select the day at which the minimum average temperature is observed.