```
(b) Video: Widely used SQL queries Activity overview 6 min

Submit your assignment
In previous lessons, you learned about the importance of being able to clean your data where it lives. When it comes to data stored in databases, that means using SQL queries. In this activity, you will create a custom dataset and table, import a CSV file, and use SQL queries to clean automobile data.

Receive grade
In this scenario, you are a data analyst working with a used car dealership startup venture. The investors want you to find out which cars are most popular with customers so they can make sure to stock accordingly.

By the time you complete this activity, you will be able to clean your data where it lives. When it comes to data schere it lives. When it comes to data stored in databases, that means using SQL queries. In this activity, you will create a custom dataset and table, import a CSV file, and use SQL queries to clean automobile data.

Receive grade
In this scenario, you are a data analyst working with a used car dealership startup venture. The investors want you to find out which cars are most popular with customers so they can make sure to stock accordingly.

Submit your assignment
In previous lessons, you learned about the importance of being able to clean your data where it lives. When it comes to data schere it lives. When it comes to dat
           What you will need
                                                              To get started, download the automobile_data CSV file. This is data from an external source that contains
                                                                 historical sales data on car prices and their features.
                                                                 Click the link to the automobile\_data file to download it. Or you may download the CSV file directly from the
                                                                  Link to data: automobile_data_ 🗹
                                                                    automobile_data
CSV File
                                                                                                                                                                              ----
                                                                  Upload your data
                                                                  Similarly to a previous BigQuery activity, you will need to create a dataset and a custom table to house your data.
                                                                  Then, you'll be able to use SQL queries to explore and clean it. Once you've downloaded the automobile_data file,
                                                                  Step 1: Create a dataset
                                                                 Go to the Explorer pane in your workspace and click the three dots next to your pinned project to open the menu. From here, select Create dataset.
                                                                                                       + ADD DATA DEDITOR
                                                                  Explorer
                                                                                                                                                  RUN
                                                                  Viewing pinned projects.
                                                                    airy-shuttle-315515
                                                                  From the Create dataset menu, fill out some information about the dataset. Input the Dataset ID as cars; you
                                                                  can leave the Data location as Default. Then click CREATE DATASET.
                                                                  Create dataset
                                                                  Encryption

© Google-managed encryption key
No configuration required

Customer-managed encryption key (CMEK)

Manage via Google Cloud Key Management Service
                                                                   CREATE DATASET CANCEL
                                                                  The cars dataset should appear under your project in the Explorer pane as shown below. Click on the three dots
                                                                  next to the cars dataset to open it.
                                                                       Explorer + ADD DATA
                                                                       Viewing pinned projects.
                                                                   ▼ airy-shuttle-315515

▼ iii babynames

                                                                                mames_2014
                                                                   cars
                                                                  Step 2: Create table
                                                                  After you open your newly created dataset, you will be able to add a custom table for your data.
                                                                  From the cars dataset, click CREATE TABLE.
                                                                ©, ESTOR - X □SARYAL - X □NAMES_ - X □ CARS - X □ COMPOSE NEW OURSY 
sity-shuttle 315515 cars □ CREATE TABLE ■ SHARE DATASET ■ AUTHORIZE ROUTINES ■ © COMPOSE NEW OURSY
                                                                 Description /
                                                                                                         Labels /
                                                              Under Source, upload the automobile_data CSV. Under Destination, make sure you are uploading into your cars dataset and name your table car_info. You can set the schema to Auto-detect. Then, click Create table.
                                                                  Destination

■ Search for a project ☐ Enter a project name

Project name

Late you ☐

Search for a project of name

Core ☐

Table type ☐

Table type ☐

Table name

Core, ando
                                                                 Schema
Auto detect
Schema and input parameters

Schema will be automatically generated.
                                                                 Clustering order (optionall): (ii)
Clustering order determines the cort order of the data. Clustering can be used on both partitioned and non-partitioned tables.
                                                                 After creating your table, it will appear in your Explorer pane. You can click on the table to explore the schema and preview your data. Once you have gotten familiar with your data, you can start querying it.
                                                                  Cleaning your data
                                                                Your new dataset contains historical sales data, including details such as car features and prices. You can use this data to find the top 10 most popular cars and trims. But before you can perform your analysis, you'll need to make sure your data is clean. If you analyze dirty data, you could end up presenting the wrong list of cars to the investors. That may cause them to lose money on their car inventory investment.
                                                                  Step 1: Inspect the fuel_type column
                                                                The first thing you want to do is inspect the data in your table so you can find out if there is any specific cleaning that needs to be done. According to the <a href="https://description">https://description</a> L², the <a href="fuel-type column">fuel-type column</a> should only have <a href="https://description">two</a> unique string values: diesel and <a href="mailto:gas-type">gas-type://description</a> Lock and make sure that's true, <a href="mailto:run">run</a> the following <a href="mailto:queen">queen</a>:
                                                                 SELECT
DISTINCT fuel_type
                                                                 FROM cars.car_info;
                                                                 This returns the following results:
                                                                      Query complete (0.6 sec elapsed, 1 KB processed)
                                                                 Job information Results JSON Execution details

Row fuel_type
                                                                 2 diesel
                                                                 This confirms that the fuel_type column doesn't have any unexpected values.
                                                                 Step 2: Inspect the length column
                                                              Next, you will inspect a column with numerical data. The length column should contain numeric measurements of the cars. So you will check that the minimum and maximum lengths in the dataset align with the data description [\mathcal{C}], which states that the lengths in this column should range from 141.1 to 208.1. Run this query to confirm
                                                                 SELECT
MIN(length) AS min_length,
                                                                   MAX(length) AS max_length
                                                                  cars.car_info;
                                                                 Your \ results \ should \ confirm\ that\ 141.1\ and\ 208.1\ are\ the\ minimum\ and\ maximum\ values\ respectively\ in\ this
                                                                  Row min_length max_length
                                                                 1 141.1 208.1
                                                                 Step 3: Fill in missing data
                                                                Missing values can create errors or skew your results during analysis. You're going to want to check your data for null or missing values. These values might appear as a blank cell or the word null in BigQuery.
                                                                 You can check to see if the num_of_doors column contains null values using this query:
                                                                  SELECT
*
                                                                  cars.car_info
                                                                  WHERE
                                                                 num_of_doors IS NULL;
                                                               This will select any rows with missing data for the num_of_doors column and return them in your results table. You should get two results, one Mazda and one Dodge:
                                                                  Row make fuel_type num_of_doors body_style
                                                                 1 dodge gas null sedan
                                                                 2 mazda diesel null sedan
                                                                In order to fill in these missing values, you check with the sales manager, who states that all Dodge gas sedans and all Mazda diesel sedans sold had four doors. If you are using the BigQuery free trial, you can use this query to update your table so that all Dodge gas sedans have four doors:
                                                                 cars.car_info
                                                                   num_of_doors = "four"
                                                                 WHERE
make = "dodge"
                                                                   AND fuel_type = "gas"
                                                                    AND body_style = "sedan";
                                                                  You should get a message telling you that three rows were modified in this table. To make sure, you can run the
                                                                  previous query again:
                                                                  SELECT *
                                                                  cars.car_info
                                                                  WHERE
                                                                   num_of_doors IS NULL;
                                                                 Now, you only have one row with a NULL value for num_of_doors. Repeat this process to replace the null value for the Mazda.
                                                                  If you are using the BigQuery Sandbox, you can skip these UPDATE queries; they will not affect your ability to
                                                                  Step 4: Identify potential errors
                                                                 Once you have finished ensuring that there aren't any missing values in your data, you'll want to check for other potential errors. You can use SELECT DISTINCT to check what values exist in a column. You can run this query to
                                                                  check the num\_of\_cylinders column:
                                                              SELECT
DISTINCT num_of_cylinders
                                                                   cars.car_info;
                                                                 After running this, you notice that there are one too many rows. There are two entries for two cylinders: rows 6 and 7. But the two in row 7 is misspelled.
                                                                  Row num_of_cylinders
                                                                 1 four
                                                                  2 six
                                                                 3 five
                                                                 4 three
                                                                  5 twelve
                                                                  6 two
                                                                 7 tow
                                                                8 eight
                                                                  To correct the misspelling for all rows, you can run this query if you have the BigQuery free trial:
                                                                   cars.car_info
                                                                   num_of_cylinders = "two"
                                                                 WHERE
num_of_cylinders = "tow";
                                                                You will get a message alerting you that one row was modified after running this statement. To check that it worked, you can run the previous query again:
SELECT
                                                                   DISTINCT num_of_cylinders
                                                                 Next, you can check the compression_ratio column. According to the <u>data description</u> [2], the compression_ratio column values should range from 7 to 23. Just like when you checked the length values,
                                                                  MIN(compression_ratio) AS min_compression_ratio,
MAX(compression_ratio) AS max_compression_ratio
                                                                   cars.car_info;
                                                                Notice that this returns a maximum of 70. But you know this is an error because the maximum value in this column should be 23, not 70. So the 70 is most likely a 7.0. Run the above query again without the row with 70 to make sure that the rest of the values fall within the expected range of 7 to 23.
                                                                  MIN(compression_ratio) AS min_compression_ratio,
MAX(compression_ratio) AS max_compression_ratio
                                                                 FROM cars.car_info
                                                                   compression_ratio <> 70;
                                                                Now the highest value is 23, which aligns with the data description. So you'll want to correct the 70 value. You check with the sales manager again, who says that this row was made in error and should be removed. Before you delete anything, you should check to see how many rows contain this erroneous value as a precaution so that you don't end up deleting 50% of your data. If there are too many (for instance, 20% of your rows have the incorrect 70 value), then you would want to check back in with the sales manager to inquire if these should be deleted or if the 70 should be updated to another value. Use the query below to count how many rows you would be deleting:
                                                                  SELECT
                                                                     COUNT(*) AS num_of_rows_to_delete
                                                                      cars.car_info
                                                                      compression_ratio = 70;
                                                                  Turns out there is only one row with the erroneous 70 value. So you can delete that row using this query:
                                                                 DELETE cars.car_info
                                                                  If you are using the {\tt BigQuery} \ sandbox, you can {\tt replace} \ {\tt DELETE} \ with {\tt SELECT} \ to see which row would be deleted.
                                                                 Step 5: Ensure consistency
                                                                 Finally, you want to check your data for any inconsistencies that might cause errors. These inconsistencies can be tricky to spot — sometimes even something as simple as an extra space can cause a problem.
                                                                  Check the drive_wheels column for inconsistencies by running a query with a SELECT DISTINCT statement:
                                                                      DISTINCT drive_wheels
                                                                  cars.car_info;
                                                                It appears that 4wd appears twice in results. However, because you used a SELECT DISTINCT statement to return unique values, this probably means there's an extra space in one of the 4wd entries that makes it different from the other 4wd.
                                                                  Row drive_wheels
                                                                  2 fwd
                                                                 4 4wd
                                                                  To check if this is the case, you can use a LENGTH statement to determine the length of how long each of these
                                                                  string variables:
                                                                SELECT
DISTINCT drive_wheels,
                                                                LENGTH(drive_wheels) AS string_length
                                                                 According to these results, some instances of the 4wd string have four characters instead of the expected three (4wd has 3 characters). In that case, you can use the TRIM function to remove all extra spaces in the drive_wheels column if you are using the BigQuery free trial:
                                                                  UPDATE
                                                                  cars.car_info
                                                                  drive_wheels = TRIM(drive_wheels)
                                                                 WHERE TRUE;
                                                                  Then, you \ \textbf{run the SELECT DISTINCT statement again} \ to \ ensure \ that \ there \ are \ only \ three \ distinct \ values \ in \ the
                                                                  drive_wheels column:
                                                                   DISTINCT drive_wheels
                                                                   cars.car_info;
                                                                  And now there should only be three unique values in this column! Which means your data is clean, consistent,
                                                                  and ready for analysis!
                                                                 Confirmation and reflection
                                                                 What is the maximum value in the price column of the car_info table?
                                                              O 16,430
                                                                 O 12,978
                                                                O 45,400
                                                              O 5,1180
                                                            2. In the text box below, write 2-3 sentences (40-60 words) in response to each of the following questions:

    Which of these cleaning techniques do you think will be most useful for you in the future?

                                                                 Your answer cannot be more than 10000 characters.
                                                             Coursera Honor Code Learn more
                                                            I, Terris Tan, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.
```

Submit Save draft

🖒 Like 🖓 Dislike 🏳 Report an issue

Using SQL to clean data

Transforming data

Weekly challenge 3