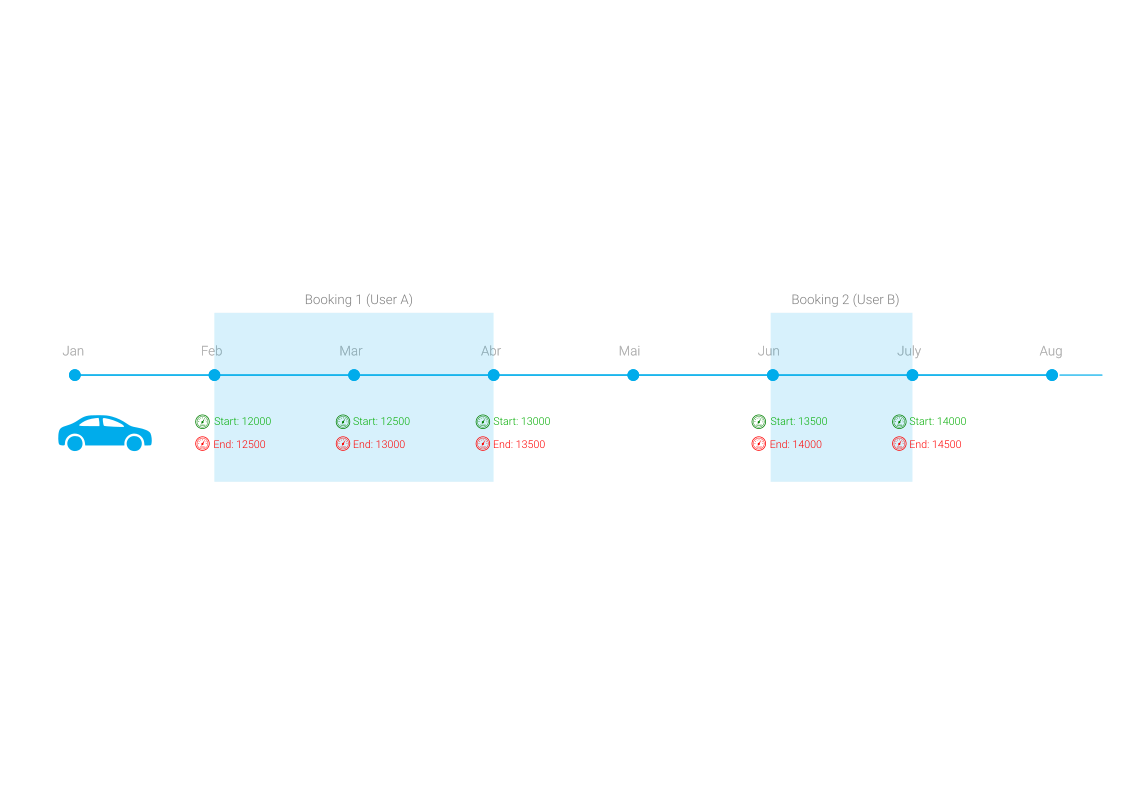
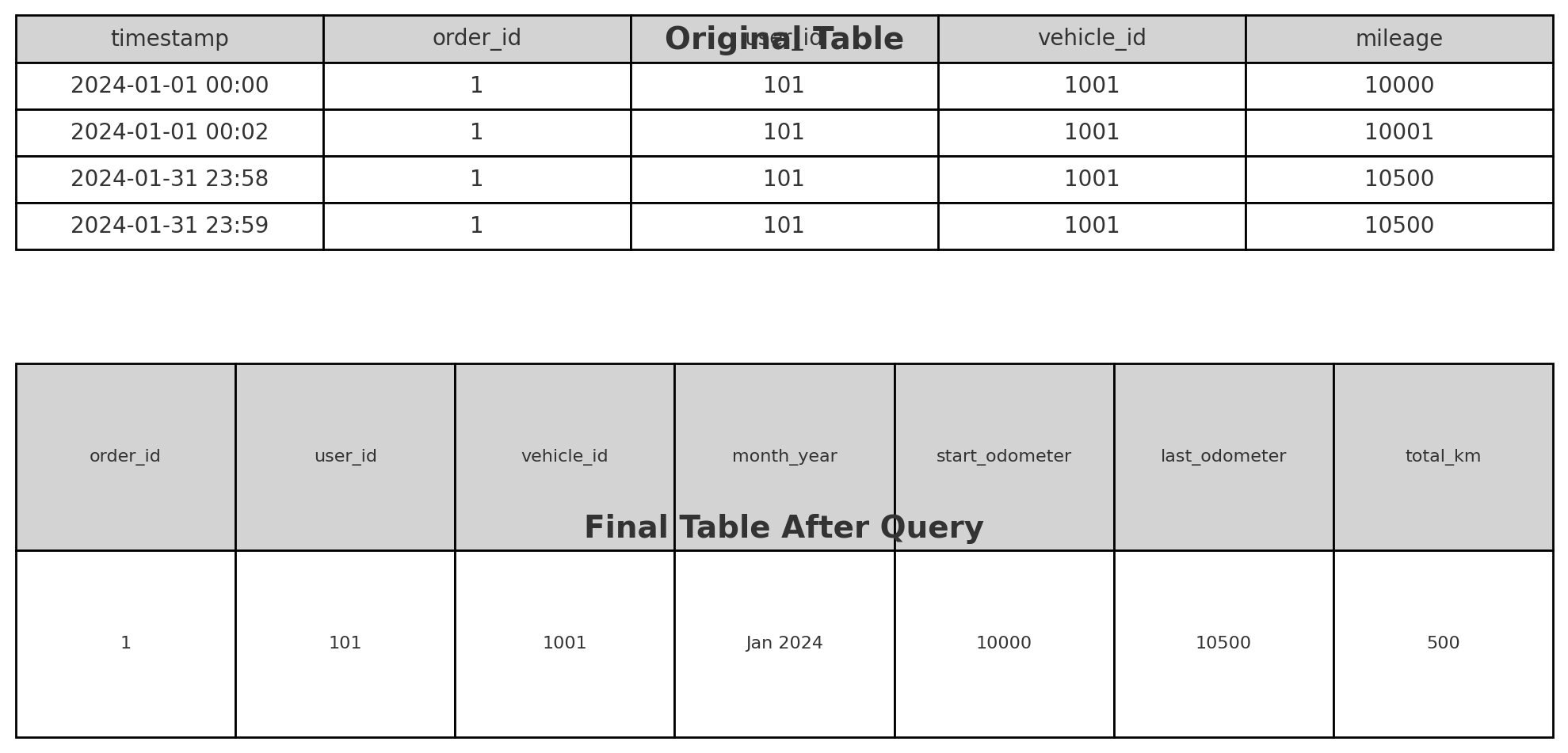
## Monitoring Vehicle Odometer Mileage for Monthly Bookings

Business Problem:

A client needs to monitor the total mileage recorded on vehicle odometers at the end of each month for every booking/user. Vehicles can be booked for periods extending over several months, and the client wants to identify both bookings and users who exceed a specific mileage threshold each month. The goal is to determine the odometer readings at the beginning and end of each month during the booking period.



The data source captures vehicle information every 2 minutes, including the booking status (Order ID), user ID (if booked), and the vehicle's odometer reading. To meet the client's requirements, we need to extract and calculate the odometer readings at the start and end of each month for each booking and subtract the last measure by the first.



The following BigQuery SQL query generates a table with the required calculations:

WITH

cte AS (

SELECT

order\_id,

user\_id,

vehicle\_id,

time,

FIRST\_VALUE(mileage) OVER w1 AS start\_odometer,

LAST\_VALUE(mileage) OVER w1 AS last\_odometer,

ROW\_NUMBER() OVER(PARTITION BY order\_id, EXTRACT(MONTH FROM time) ORDER BY time ASC) AS rank\_order

FROM

`data-analyst-326910.vehicle\_km.rt2\_corrected3`

WINDOW

w1 AS(

PARTITION BY order\_id, EXTRACT(MONTH FROM time)

ORDER BY time ASC ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) )

SELECT

\* EXCEPT(rank\_order)

FROM (

SELECT

\*,

ROUND(last\_odo - start\_odo,0) AS total\_km

FROM

cte

WHERE

rank\_order = 1

AND order\_id IS NOT NULL

ORDER BY

order\_id)



Explanation:

1. CTE (Common Table Expression):

* Retrieves necessary details from the main table, including order\_id, user\_id, vehicle\_id, time, and mileage.
* Calculates the first and last odometer readings for each booking within each month using window functions.
* Ranks the records within each order and month to ensure we are capturing the start and end readings correctly.

1. Final Query:

* Computes the total kilometers traveled within the month (total\_km) by subtracting the starting odometer reading from the ending odometer reading.
* Filters the records to include only the first-ranked entries for each order within each month (to get the initial and final readings).
* Ensures only non-null order\_id entries are considered.
* Orders the results by order\_id.