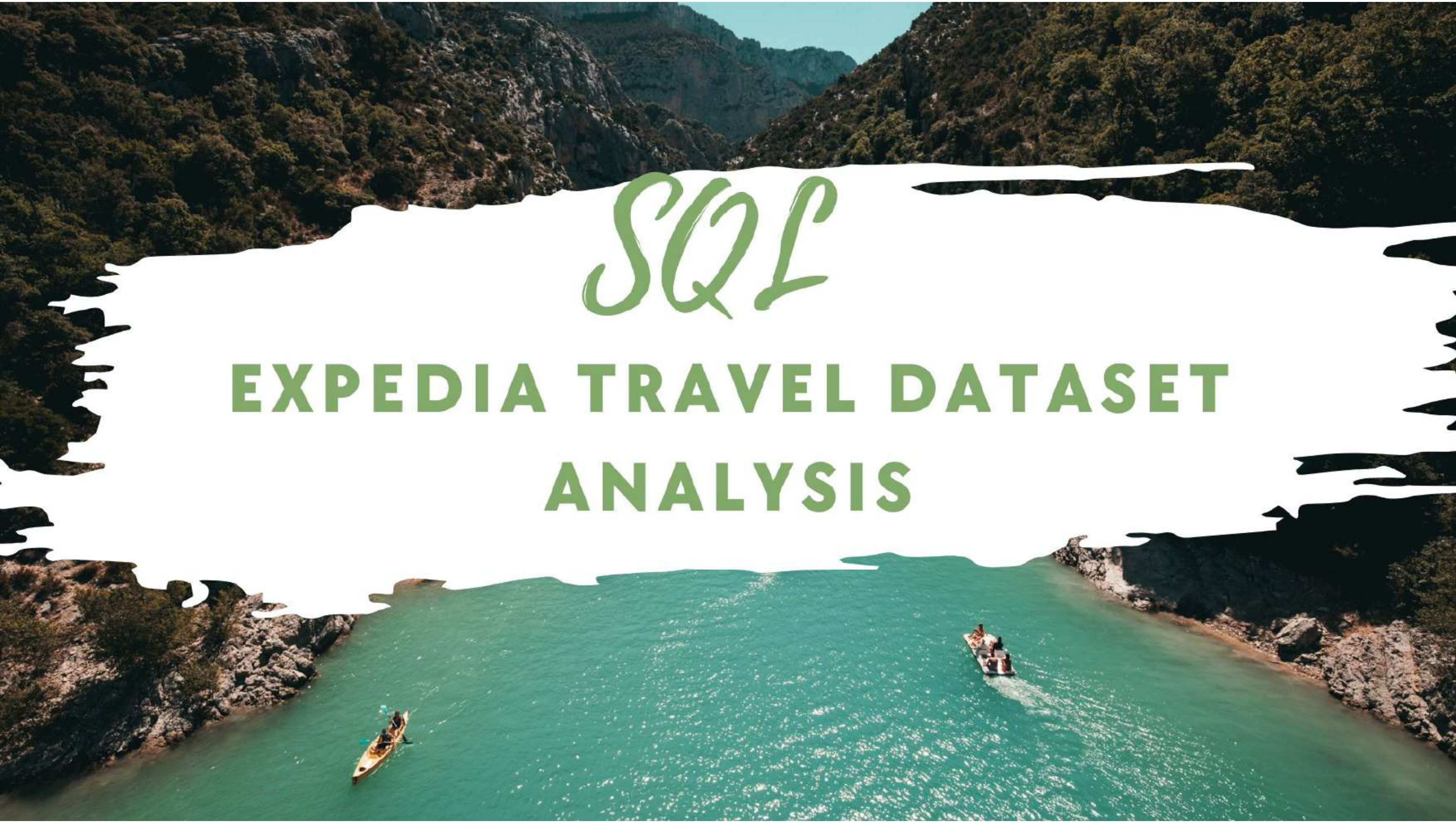
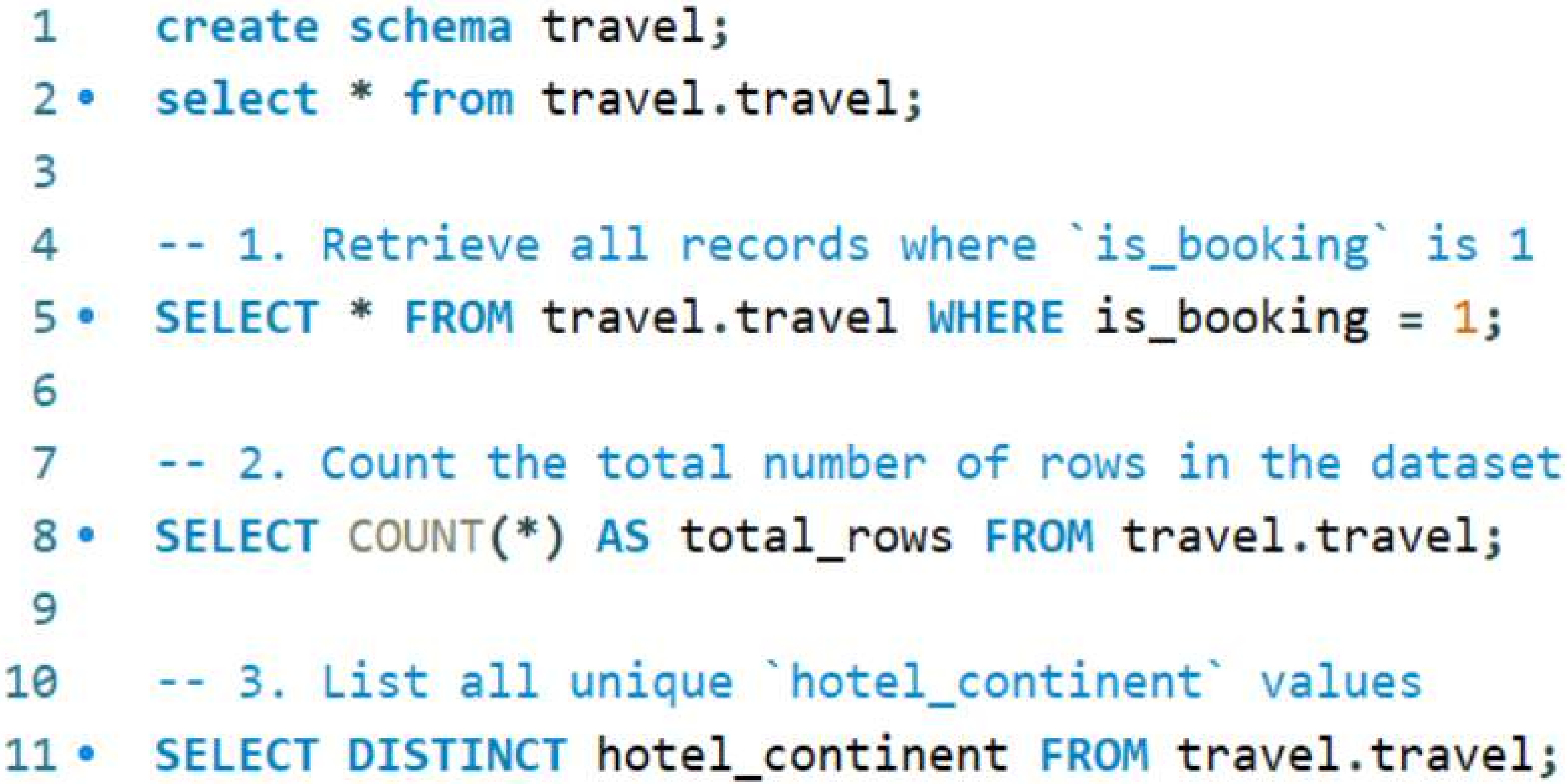
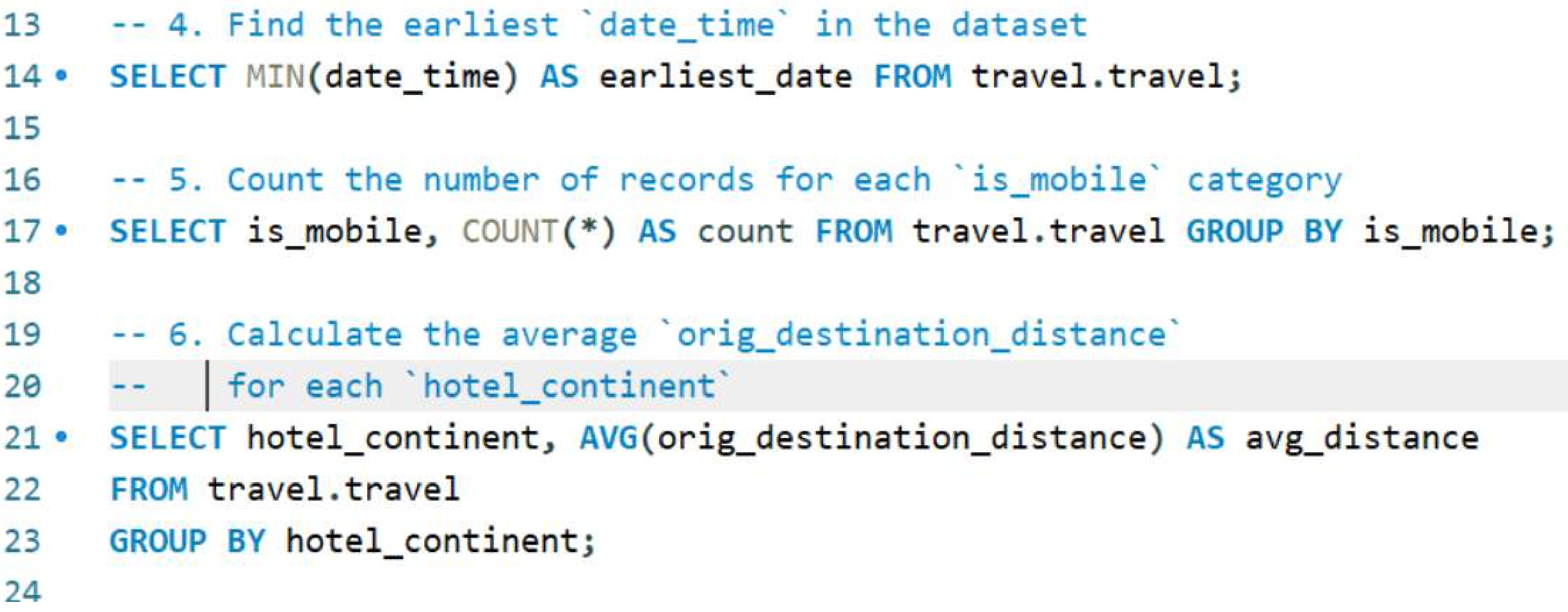


SQL

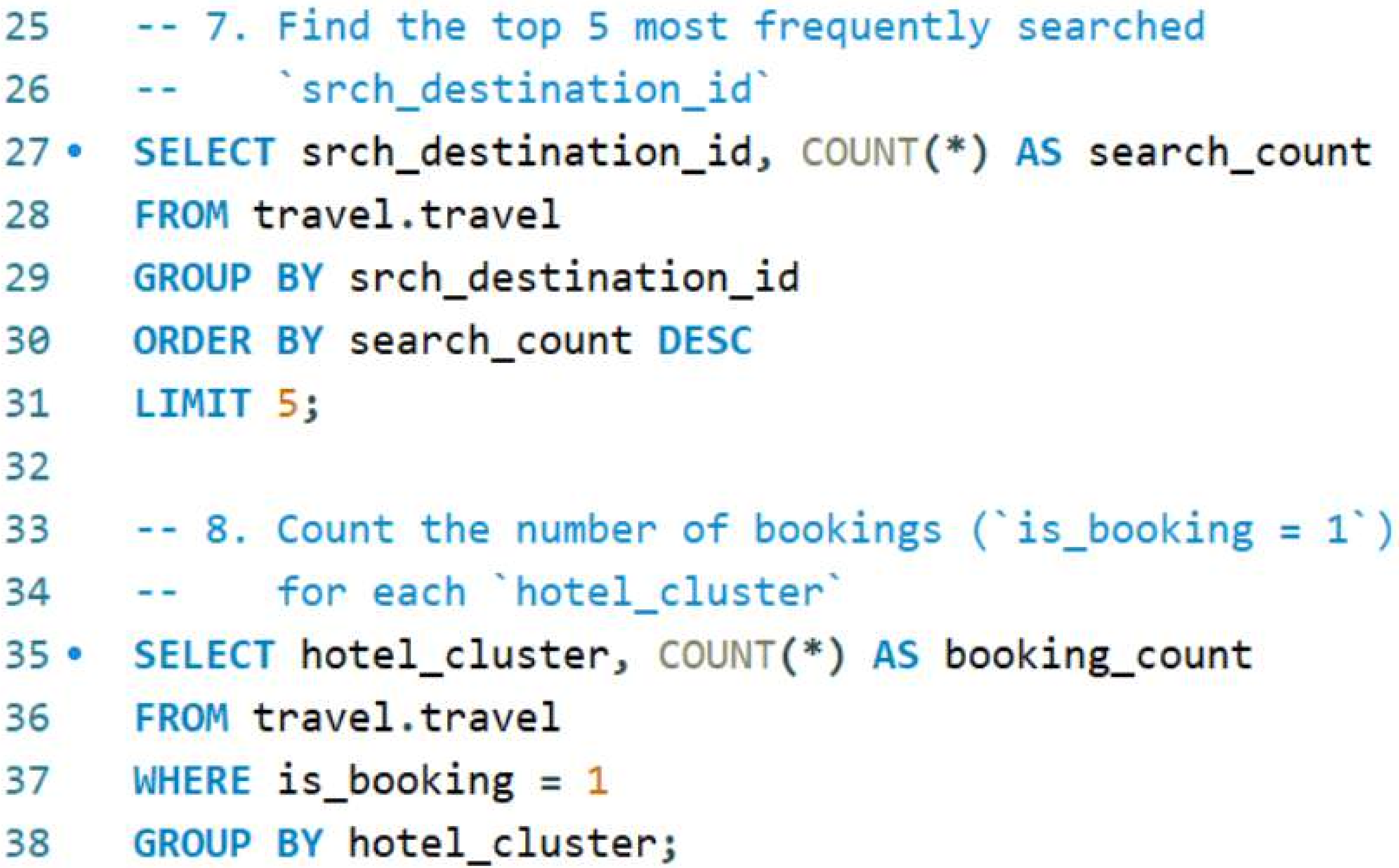
EXPEDIA TRAVEL DATASET ANALYSIS



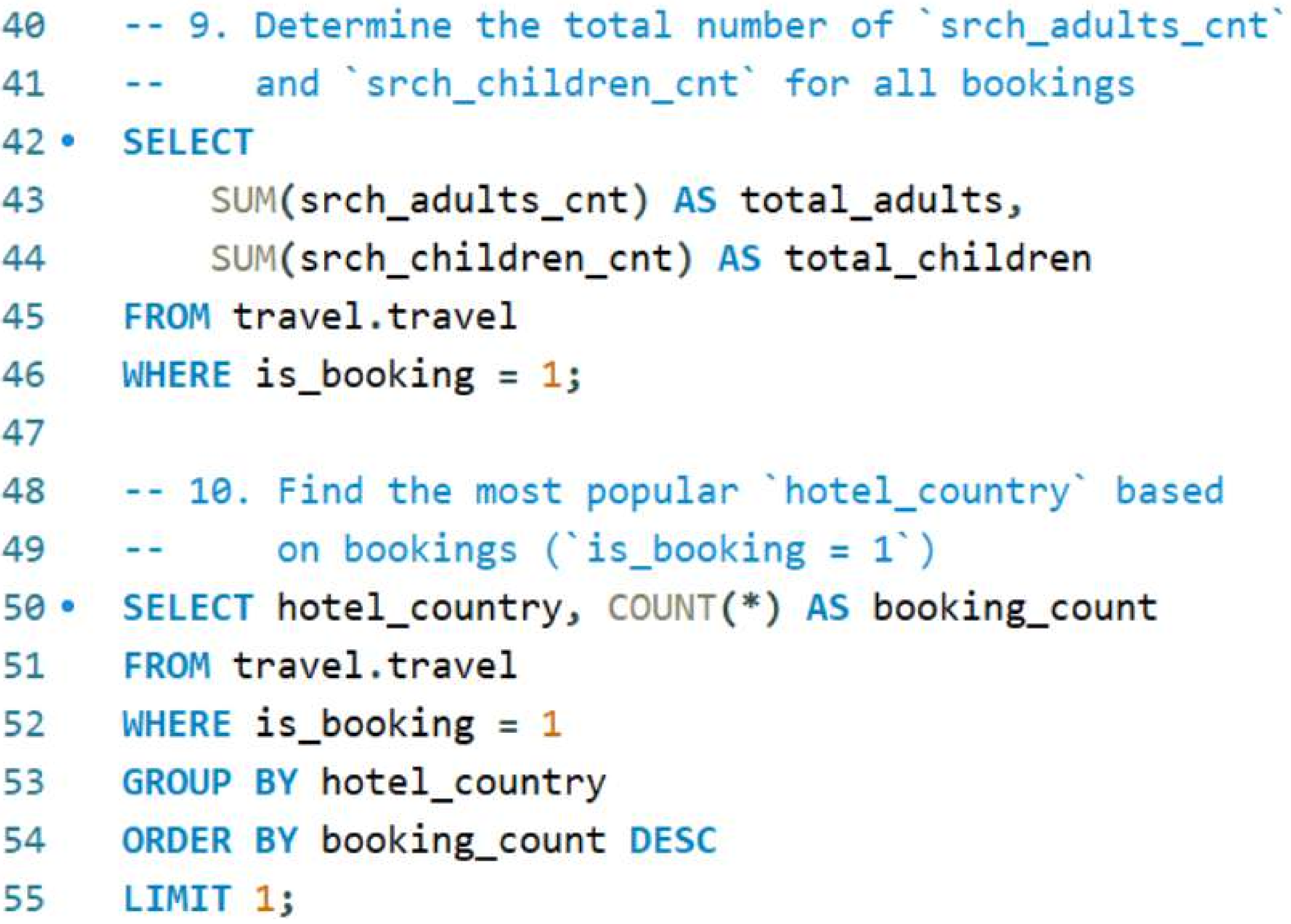
```sql
create schema travel;
select * from travel.travel;

-- 1. Retrieve all records where `is_booking` is 1
SELECT * FROM travel.travel WHERE is_booking = 1;

-- 2. Count the total number of rows in the dataset
SELECT COUNT(*) AS total_rows FROM travel.travel;

-- 3. List all unique `hotel_continent` values
SELECT DISTINCT hotel_continent FROM travel.travel;
```



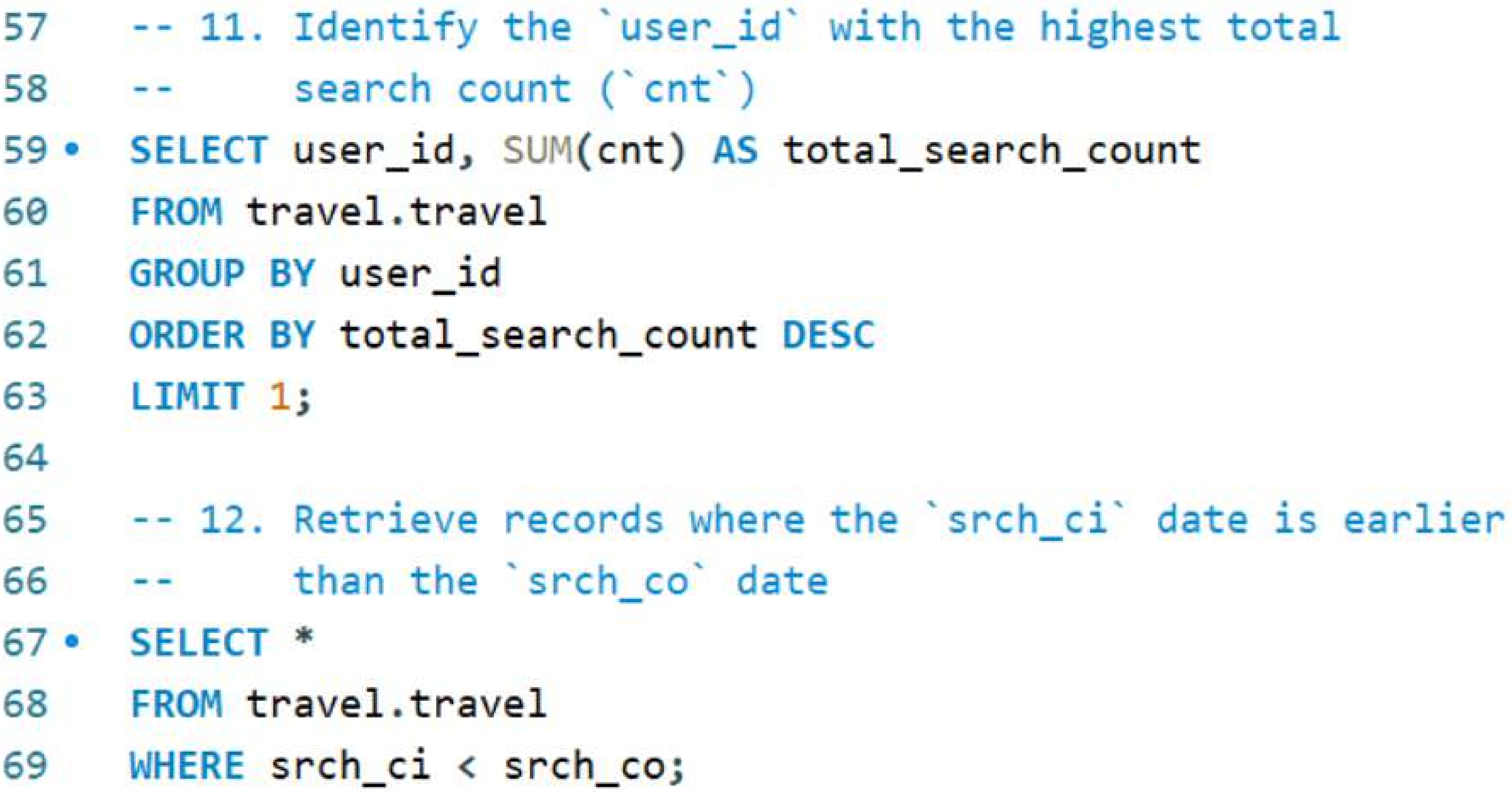
```sql
-- 4. Find the earliest `date_time` in the dataset
SELECT MIN(date_time) AS earliest_date FROM travel.travel;

-- 5. Count the number of records for each `is_mobile` category
SELECT is_mobile, COUNT(*) AS count FROM travel.travel GROUP BY is_mobile;

-- 6. Calculate the average `orig_destination_distance`
--    for each `hotel_continent`
SELECT hotel_continent, AVG(orig_destination_distance) AS avg_distance
FROM travel.travel
GROUP BY hotel_continent;

```



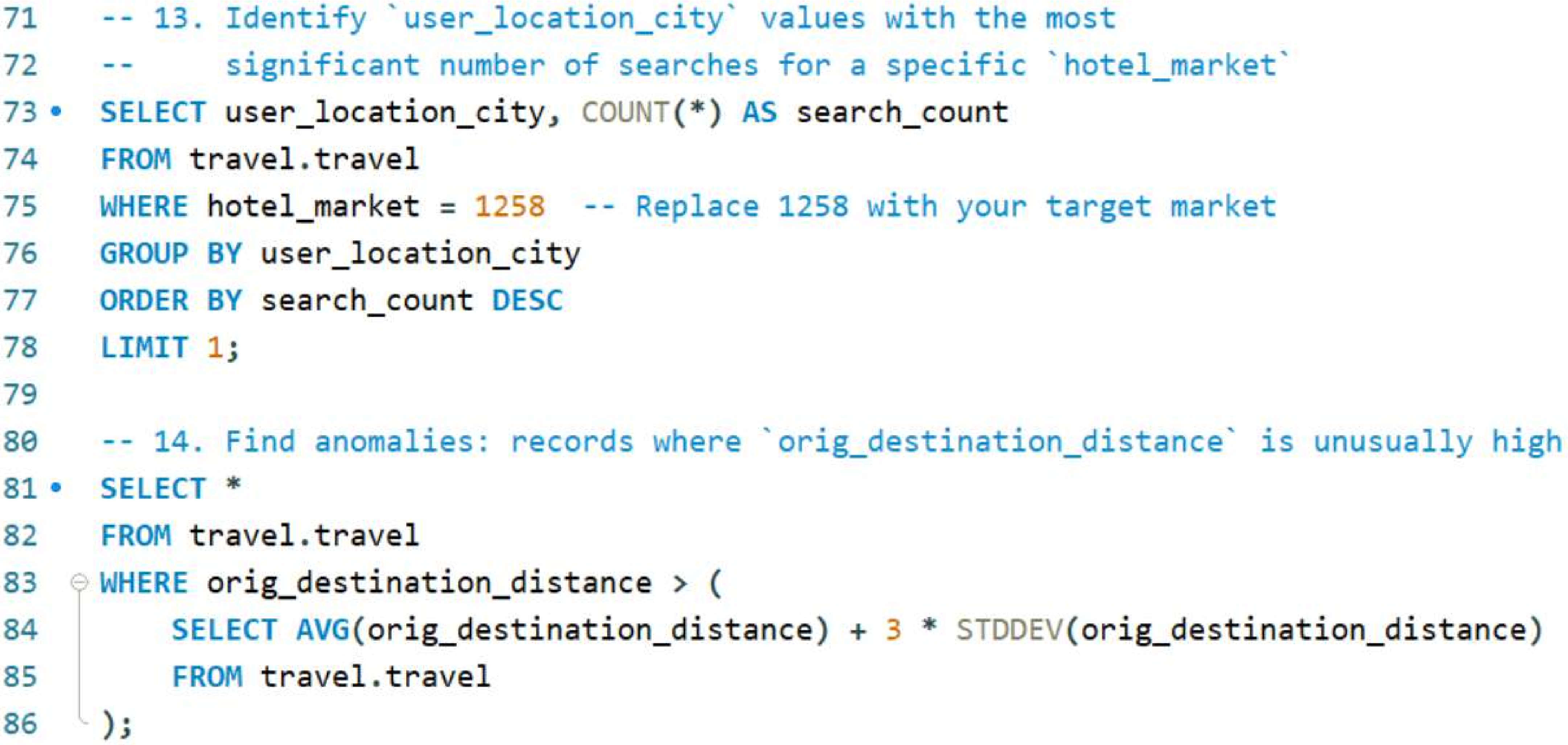
```sql
-- 7. Find the top 5 most frequently searched
--    `srch_destination_id`
SELECT srch_destination_id, COUNT(*) AS search_count
FROM travel.travel
GROUP BY srch_destination_id
ORDER BY search_count DESC
LIMIT 5;

-- 8. Count the number of bookings (`is_booking = 1`)
--    for each `hotel_cluster`
SELECT hotel_cluster, COUNT(*) AS booking_count
FROM travel.travel
WHERE is_booking = 1
GROUP BY hotel_cluster;
```



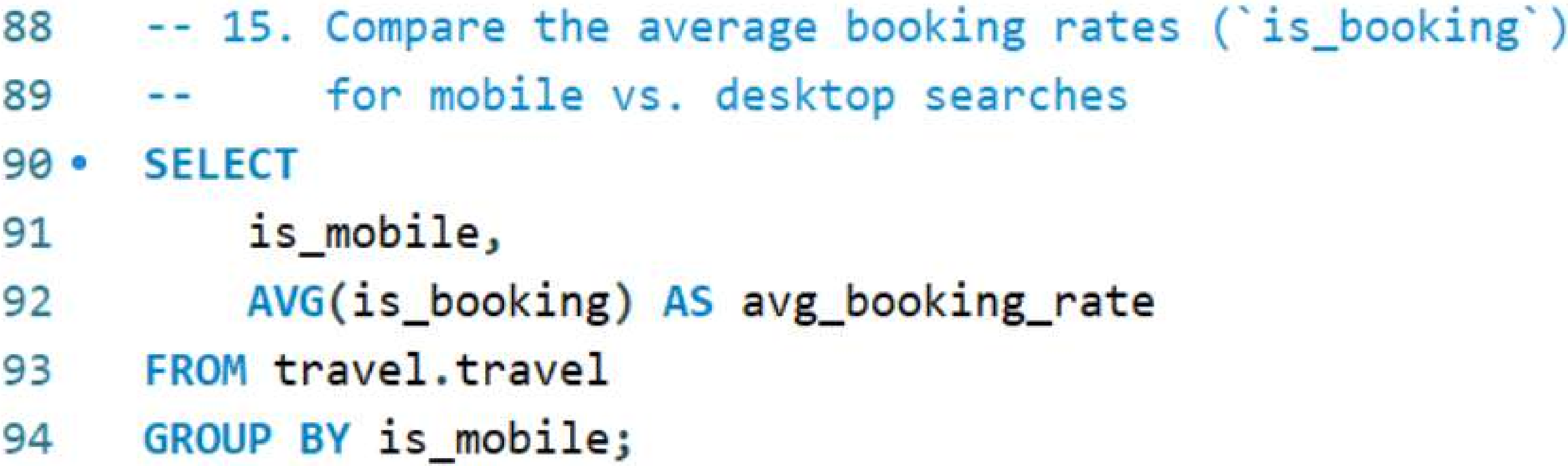
```sql
-- 9. Determine the total number of `srch_adults_cnt`
--    and `srch_children_cnt` for all bookings
SELECT
    SUM(srch_adults_cnt) AS total_adults,
    SUM(srch_children_cnt) AS total_children
FROM travel.travel
WHERE is_booking = 1;

-- 10. Find the most popular `hotel_country` based
--     on bookings (`is_booking = 1`)
SELECT hotel_country, COUNT(*) AS booking_count
FROM travel.travel
WHERE is_booking = 1
GROUP BY hotel_country
ORDER BY booking_count DESC
LIMIT 1;
```



```sql
-- 11. Identify the `user_id` with the highest total
--      search count (`cnt`)
SELECT user_id, SUM(cnt) AS total_search_count
FROM travel.travel
GROUP BY user_id
ORDER BY total_search_count DESC
LIMIT 1;

-- 12. Retrieve records where the `srch_ci` date is earlier
--      than the `srch_co` date
SELECT *
FROM travel.travel
WHERE srch_ci < srch_co;
```



```sql
-- 13. Identify `user_location_city` values with the most
--     significant number of searches for a specific `hotel_market`
SELECT user_location_city, COUNT(*) AS search_count
FROM travel.travel
WHERE hotel_market = 1258  -- Replace 1258 with your target market
GROUP BY user_location_city
ORDER BY search_count DESC
LIMIT 1;

-- 14. Find anomalies: records where `orig_destination_distance` is unusually high
SELECT *
FROM travel.travel
WHERE orig_destination_distance > (
    SELECT AVG(orig_destination_distance) + 3 * STDDEV(orig_destination_distance)
    FROM travel.travel
);
```



```sql
-- 15. Compare the average booking rates (`is_booking`)
--      for mobile vs. desktop searches
SELECT
    is_mobile,
    AVG(is_booking) AS avg_booking_rate
FROM travel.travel
GROUP BY is_mobile;
```