Partie 2 — Advanced SQL Queries Only

20) Give the number of vehicles per vehicle type

```
SELECT type, COUNT(*) AS number_of_vehicles
FROM Vehicle
GROUP BY type;
```

21) List the number of clients per city

```
SELECT city, COUNT(*) AS number_of_clients
FROM Client
GROUP BY city;
```

22) Give the total number of rentals per client

```
SELECT c.id_client,
    c.name,
    COUNT(r.id_rental) AS number_of_rentals
FROM Client c
LEFT JOIN Rental r ON c.id_client = r.id_client
GROUP BY c.id_client, c.name;
```

23) Give the number of rentals per vehicle type

```
SELECT v.type,
    COUNT(r.id_rental) AS number_of_rentals
FROM Vehicle v
JOIN Rental r ON v.id_vehicle = r.id_vehicle
GROUP BY v.type;
```

24) Calculate the average duration of rentals in days

```
SELECT AVG(end_date - start_date) AS avg_duration_days
FROM Rental;
```

25) Calculate the average duration of rentals per agency in days

```
SELECT a.id_agency,  a.agency_name,  AVG(r.end_date - r.start_date) AS avg_duration_days
FROM Agency a
JOIN Vehicle_Agency va ON a.id_agency = va.id_agency
JOIN Rental r ON va.id_vehicle = r.id_vehicle
GROUP BY a.id_agency, a.agency_name;
```

26) Calculate the average daily rate for each vehicle brand

```
SELECT brand,    AVG(daily_rate) AS avg_daily_rate
FROM Vehicle
GROUP BY brand;
```

27) Give the number of rentals per rental start year

```
SELECT EXTRACT(YEAR FROM start_date) AS rental_start_year, COUNT(*) AS number_of_rentals
FROM Rental
GROUP BY EXTRACT(YEAR FROM start_date)
ORDER BY rental_start_year;
```

28) For each client, give the total number of kilometers traveled

```
SELECT c.id_client, c.name, c.surname, COALESCE(SUM(r.km_end - r.km_start), 0) AS total_km
FROM Client c
LEFT JOIN Rental r ON c.id_client = r.id_client
GROUP BY c.id_client, c.name, c.surname;
```

29) For each agency, give the number of assigned vehicles and the average daily rate of available vehicles

Comme le schéma ne contient pas d'attribut available, j'interprète ici les véhicules disponibles comme les véhicules affectés à l'agence et **non loués actuellement**.

```
SELECT a.id_agency, a.agency_name,  COUNT(DISTINCT va.id_vehicle) AS number_of_assigned_vehicles,
    AVG( CASE
        WHEN NOT EXISTS (
          SELECT 1
          FROM Rental r
          WHERE r.id_vehicle = va.id_vehicle
            AND CURRENT_DATE BETWEEN r.start_date AND r.end_date
        )
        THEN v.daily_rate
      END
    ) AS avg_daily_rate_available
FROM Agency a
LEFT JOIN Vehicle_Agency va ON a.id_agency = va.id_agency
LEFT JOIN Vehicle v ON va.id_vehicle = v.id_vehicle
GROUP BY a.id_agency, a.agency_name;
```

30) Give the number of distinct clients who rented a vehicle of type "Utilitaire"

```
SELECT COUNT(DISTINCT r.id_client) AS number_of_clients
FROM Rental r
JOIN Vehicle v ON r.id_vehicle = v.id_vehicle
WHERE v.type = 'Utilitaire';
```

31) For each client, give the average rate of the vehicles they rented

```
SELECT c.id_client, c.name, c.surname, AVG(v.daily_rate) AS avg_rented_vehicle_rate
FROM Client c
LEFT JOIN Rental r ON c.id_client = r.id_client
LEFT JOIN Vehicle v ON r.id_vehicle = v.id_vehicle
GROUP BY c.id_client, c.name, c.surname;
```

32) Calculate the average number of rental days per client

```
SELECT c.id_client,  c.name,   c.surname,  AVG(r.end_date - r.start_date) AS avg_rental_days
FROM Client c
LEFT JOIN Rental r ON c.id_client = r.id_client
GROUP BY c.id_client, c.name, c.surname;
```

33) List the most rented vehicles, indicating how many times each vehicle was rented

```
SELECT v.id_vehicle,  v.brand,  v.model,   COUNT(r.id_rental) AS rental_count
FROM Vehicle v
```

```
JOIN Rental r ON v.id_vehicle = r.id_vehicle
GROUP BY v.id_vehicle, v.brand, v.model
HAVING COUNT(r.id_rental) = (
   SELECT MAX(nb_rentals)
   FROM (
      SELECT COUNT(*) AS nb_rentals
      FROM Rental
      GROUP BY id_vehicle
   ) t
);
```

34) Display the vehicles whose rate is higher than the average rate of rented vehicles

```
SELECT *
FROM Vehicle
WHERE daily_rate > (
   SELECT AVG(v.daily_rate)
   FROM Vehicle v
   JOIN Rental r ON v.id_vehicle = r.id_vehicle
);
```

35) Find the vehicles that have never been rented using NOT EXISTS

```
SELECT v.id_vehicle,
      v.brand,
      v.model,
      v.type,
      v.daily_rate
FROM Vehicle v
WHERE NOT EXISTS (
   SELECT *
   FROM Rental r
   WHERE r.id_vehicle = v.id_vehicle
);
```