```sql
-- Les noms des colonnes pour le premier trimestre 2020 diffèrent de ceux de 2019.
Pour simplifier les opérations, il est nécessaire d'uniformiser les noms en
attributant ceux de 2019 à 2020, tout en créant une nouvelle TABLE dédiée au
premier trimestre 2020.
CREATE TABLE
  `caduran2025.Projet_certif_google.Trajet_Q1_2020_v2` AS
SELECT
  ride_id AS trip_id,
  started_at AS start_time,
  ended_at AS end_time,
  start_station_id AS from_station_id,
  start_station_name AS from_station_name,
  end_station_id AS to_station_id,
  end_station_name AS to_station_name,
  member_casual AS usertype,
  rideable_type,
  start_lat,
  start_lng,
  end_lat,
  end_lng
FROM
  `caduran2025.Projet_certif_google.Trajet_Q1_2020`;

--Convertir le format trip_id, bikeid,from_station_id,to_station_id (integer) en
string
SELECT
  CAST(trip_id AS string) AS trip_id,
  cast(bikeid as string) as bikeid,
  cast(from_station_id as string) as from_station_id,
  cast(to_station_id AS string) as to_station_id,
  *
FROM
  `caduran2025.Projet_certif_google.Trajet2019_Q1`


SELECT
  CAST(trip_id AS string) AS trip_id,
  cast(bikeid as string) as bikeid,
  cast(from_station_id as string) as from_station_id,
  cast(to_station_id AS string) as to_station_id,
  *
FROM
  `caduran2025.Projet_certif_google.Trajet2019_Q4`


SELECT
  cast(from_station_id as string) as from_station_id,
  cast(to_station_id AS string) as to_station_id,
  *
FROM
  `caduran2025.Projet_certif_google.Trajet2020_Q1_v1`

-- Créer une nouvelle table en ajoutant les données Q1 et Q4 2019, Q1 2020
CREATE TABLE
  `caduran2025.Projet_certif_google.Trajet` AS
SELECT
  trip_id,
  start_time,
```

```sql
  end_time,
  from_station_id,
  from_station_name,
  to_station_id,
  to_station_name,
  usertype,

FROM
  `caduran2025.Projet_certif_google.Trajet2019_Q1_V1`

UNION ALL

SELECT
  trip_id,
  start_time,
  end_time,
  from_station_id,
  from_station_name,
  to_station_id,
  to_station_name,
  usertype,

FROM
  `caduran2025.Projet_certif_google.Trajet2019_Q4_v1`

UNION ALL

SELECT
  trip_id,
  start_time,
  end_time,
  from_station_id,
  from_station_name,
  to_station_id,
  to_station_name,
  usertype,
FROM
  `caduran2025.Projet_certif_google.Trajet2020_Q1_v2`


-- Ajouter les colonnes supplémentaires de 2019 et 2020 à la table Trajet
-- Avant toute chose, combinons les tables du 1er et 4e trimestre de 2019

CREATE TABLE
  `caduran2025.Projet_certif_google.Trajet2019_Q1_Q4` AS
SELECT
  trip_id,
  start_time,
  end_time,
  from_station_id,
  from_station_name,
  to_station_id,
  to_station_name,
  usertype,

FROM
  `caduran2025.Projet_certif_google.Trajet2019_Q1_V1`
```

```sql
UNION ALL

SELECT
  trip_id,
  start_time,
  end_time,
  from_station_id,
  from_station_name,
  to_station_id,
  to_station_name,
  usertype,

FROM
  `caduran2025.Projet_certif_google.Trajet2019_Q4_v1`


-- Ajoutons les colonnes bikeid, gender et birthyear à la table Trajet
ALTER TABLE
  `caduran2025.Projet_certif_google.Trajet`
ADD COLUMN
  bikeid1 STRING,
ADD COLUMN
  gender1 STRING,
ADD COLUMN
  birthyear1 INTEGER;

---Mettre à jour les colonnes avec les données de Trajet2019
UPDATE
  `caduran2025.Projet_certif_google.Trajet` AS Trajet
SET
  Trajet.bikeid1 = Trajet2019_Q1_Q4.bikeid,
  Trajet.gender1 = Trajet2019_Q1_Q4.gender,
  Trajet.birthyear1 = Trajet2019_Q1_Q4.birthyear
FROM
  `caduran2025.Projet_certif_google.Trajet2019_Q1_Q4` AS Trajet2019_Q1_Q4
WHERE
  Trajet.trip_id = Trajet2019_Q1_Q4.trip_id;rideable_type

-- Ajouter les colonnes rideable, latitude, longitude à la table Trajet
ALTER TABLE
  `caduran2025.Projet_certif_google.Trajet`
ADD COLUMN
  rideable_type2 STRING,
ADD COLUMN
  start_lat2 Float64,
ADD COLUMN
  start_lng2 Float64,
ADD COLUMN
  end_lat2 Float64,
ADD COLUMN
  end_lng2 Float64;

--Mettre à jour les colonnes avec les données de Trajet2020_Q1_v2
UPDATE
  `caduran2025.Projet_certif_google.Trajet` AS Trajet
SET
  Trajet.rideable_type2 = Trajet2020_Q1_v2.rideable_type,
  Trajet.start_lat2 = Trajet2020_Q1_v2.start_lat,
```

```
  Trajet.start_lng2 = Trajet2020_Q1_v2.start_lng,
  Trajet.end_lat2 = Trajet2020_Q1_v2.end_lat,
  Trajet.end_lng2 = Trajet2020_Q1_v2.end_lng
FROM
  `caduran2025.Projet_certif_google.Trajet2020_Q1_v2` AS Trajet2020_Q1_v2
WHERE
  Trajet.trip_id = Trajet2020_Q1_v2.trip_id;

-- Supprimer les colonnes vides
ALTER TABLE
  `caduran2025.Projet_certif_google.Trajet`
DROP COLUMN
  end_lng1,
DROP COLUMN
  end_lng,
  ...
--Changer les noms de deux observations pour avoir les mêmes noms d'observations
UPDATE
  `caduran2025.Projet_certif_google.Trajet`
SET
  usertype =
  CASE
    WHEN usertype = 'Subscriber' THEN 'member'
    WHEN usertype = 'Customer' THEN 'casual'
    ELSE usertype
END
WHERE
  usertype IN ('Subscriber', 'Customer');

  --Déterminer s'il y a de doublon
SELECT
  trip_id,
  COUNT(*)
FROM
  `caduran2025.Projet_certif_google.Trajet`
GROUP BY
  trip_id
HAVING
  COUNT (*) > 1;
--Il n'ya pas de doublon dans la base du Trajet


--Identifier les cellules vides de la base
SELECT
  *
FROM
  `caduran2025.Projet_certif_google.Trajet`
WHERE
  trip_id IS NULL
  OR start_time IS NULL
  OR end_time IS NULL
  OR from_station_name IS NULL
  OR from_station_id IS NULL
  OR to_station_id IS NULL
  OR to_station_name IS NULL
  OR usertype IS NULL
--Il y a 2 cellules vides dont l'une dans la colonne 'to_station_id' et l'autre
'to_station_name'
```

```sql
-- Je n'ai pas pris en compte les autres colonnes dans cette partie car s'ils devraient faire une analyse plus poussée sur le genre, la station et autres ils vont devoir ajouter ces variables dans la base de données de 2019

-- Compléter les espaces vides par analogie
UPDATE
  `caduran2025.Projet_certif_google.Trajet`
SET
  to_station_name = "HQ QR",
  to_station_id = '675'
WHERE
  trip_id = '157EAA4C4A3C8D36'

--Calculer la durée du trajet en seconde
SELECT
  trip_id,
  start_time,
  end_time,
  from_station_id,
  from_station_name,
  to_station_id,
  to_station_name,
  usertype,
  bikeid1,
  gender1,
  birthyear1,
  rideable_type2,
  start_lat2,
  start_lng2,
  end_lat2,
  end_lng2,
  CASE
    WHEN end_time < start_time THEN NULL
    ELSE TIMESTAMP_DIFF(end_time, start_time, SECOND)
END
  AS Trip_duration
FROM
  `caduran2025.Projet_certif_google.Trajet`

-- Calculer ride_length sur le format hh:mn:ss
SELECT
  *,
  CASE
    WHEN end_time < start_time THEN NULL
    ELSE FORMAT('%02d:%02d:%02d', DIV(TIMESTAMP_DIFF(end_time, start_time, SECOND),
3600), MOD(DIV(TIMESTAMP_DIFF(end_time, start_time, SECOND), 60), 60),
MOD(TIMESTAMP_DIFF(end_time, start_time, SECOND), 60))
END
  AS ride_length
FROM
  `caduran2025.Projet_certif_google.Trajet1`;

-- Filtrer les résultats NULLS
SELECT
  *
FROM
  `caduran2025.Projet_certif_google.Trajet2`
```

```sql
WHERE
  Trip_duration IS NULL

--Il y a 130 cellules nulles

-- Supprimer ces 130 lignes nulles
DELETE
FROM
  `caduran2025.Projet_certif_google.Trajet2`
WHERE
  Trip_duration IS NULL

-- Calculer le nombre de trajet dont leurs durées sont inférieures à 120 secondes
SELECT
  *
FROM
  `caduran2025.Projet_certif_google.Trajet2`
WHERE
  Trip_duration < 120

--Il y a 22934 trajets qui sont inférieurs à 120 secondes

--Supprimer les trajets inférieurs à 120 secondes
DELETE
FROM
  `caduran2025.Projet_certif_google.Trajet2`
WHERE
  Trip_duration < 120


--la durée maximum est de 10632022 secondes, il y a une erreur humaine, selon le projet les vélos sont en libre service, donc une personne ne peut rentrer avec le vélo chez elle, et je considère qu'un trajet ne peut pas durer plus de 3 heures ou 10800 secondes, donc cest une erreur dans la base, je vais supprimer tous les trajets au dessus de 3 heures ou 10800 secondes

--Filtrer les durées supérieures à 3 h ou 10800s
SELECT
  *
FROM
  `caduran2025.Projet_certif_google.Trajet2`
WHERE
  Trip_duration > 10800

-- Supprimer les 4334 trajets qui ont une durée supérieure à 10800 secondes
DELETE
FROM
  `caduran2025.Projet_certif_google.Trajet2`
WHERE
  Trip_duration > 10800


--Calculer jour de la semaine où chaque trajet a commencé
SELECT
  *,
  EXTRACT(DAYOFWEEK
  FROM
    start_time) AS day_of_week
```

```sql
FROM
  `caduran2025.Projet_certif_google.Trajet2`;


--Calculer les mois de chaque début de trajet
ALTER TABLE
  `caduran2025.Projet_certif_google.Trajet3`
ADD COLUMN IF NOT EXISTS month STRING;
UPDATE
  `caduran2025.Projet_certif_google.Trajet3`
SET
  month = FORMAT_DATE("%B", start_time)
WHERE
  start_time IS NOT NULL;


-- Création de la colonne season
ALTER TABLE
  `caduran2025.Projet_certif_google.Trajet3` ADD COLUMN season STRING;
UPDATE
  `caduran2025.Projet_certif_google.Trajet3`
SET
  season =
  CASE
    WHEN EXTRACT(MONTH FROM SAFE_CAST(start_time AS DATETIME)) IN (12, 1, 2) THEN
'Hiver'
    WHEN EXTRACT(MONTH FROM SAFE_CAST(start_time AS DATETIME)) IN (3, 4, 5) THEN
'Printemps'
    WHEN EXTRACT(MONTH FROM SAFE_CAST(start_time AS DATETIME)) IN (6, 7, 8) THEN
'Été'
    WHEN EXTRACT(MONTH FROM SAFE_CAST(start_time AS DATETIME)) IN (9, 10, 11) THEN
'Automne'
    ELSE NULL
END
WHERE TRUE;

-- Calcul statistique descriptive sur la durée du trajet des utilisateurs
SELECT
  AVG(Trip_duration) AS average_trip_duration,
  STDDEV(Trip_duration) AS ecart_type_trip_duration,
  VAR_POP(Trip_duration) AS variance_trip_duration,
  count(*) as n
FROM
  `caduran2025.Projet_certif_google.Trajet3`;

-- en moyenne les utilisateurs ont fait une durée de 804 secondes c'est-à-dire
moins d'une heure de trajet, avec un total de 1468612 trajets distincts.


-- Statistiques descriptives par type d'utilisateur
SELECT
  usertype,
  AVG(Trip_duration) AS average_trip_duration,
  APPROX_QUANTILES(Trip_duration, 2)[OFFSET (1)] AS median_trip_duration,
  MAX(Trip_duration) AS max_trip_duration,
  MIN(Trip_duration) AS min_trip_duration,
  STDDEV(Trip_duration) AS ecartType_trip_duration,
  VAR_POP(Trip_duration) AS variance_trip_duration,
```

```
  COUNT(*) AS n
FROM
  `caduran2025.Projet_certif_google.Trajet3`
GROUP BY
  usertype;



--Avec une population de 1468612, les clients occasionnels représentent 11,60% et 88,40% les membres, il y a un fort écart entre les deux groupes. En catégorisant les durées de trajet, il semblerait que les clients occasionnels font en moyenne plus de temps avec les vélos que les membres soient 672 secondes contre 1803 secondes. La médiane nous montre que 50% des clients occasionnels a fait une durée de 1290 secondes contre 50% des membres avec 527 secondes.  Cette tendance montre que les clients occasionnels font plus de temps avec le vélo que les membres. Mais puisqu'il y a un écart important entre les deux groupes, le mieux c'est d'avancer les calculs statistiques pour tirer une conclusion réelle et logique.

-- Mode de day_of_week
SELECT
  day_of_week,
  COUNT(*) AS count
FROM
  `caduran2025.Projet_certif_google.Trajet3`
GROUP BY
  day_of_week
ORDER BY
  count DESC

-- Mode de day_of_week par usertype
SELECT
  day_of_week,usertype,
  COUNT(*) AS n
FROM
  `caduran2025.Projet_certif_google.Trajet3`
GROUP BY
  day_of_week, usertype
ORDER BY
  n DESC

--Statistiques descriptives par jour de la semaine et type d'utilisateur
SELECT
  day_of_week,
  usertype,
  AVG(Trip_duration) AS trip_duration,
  COUNT(*) AS n
FROM
  `caduran2025.Projet_certif_google.Trajet3`
GROUP BY
  day_of_week, usertype
ORDER BY
  day_of_week, usertype;

--Statistiques descriptives par saison et type d'utilisateur
SELECT
  season,
  usertype,
```

```sql
  AVG(Trip_duration) AS moy_trip_duration,
  COUNT(*) AS n
FROM
  `caduran2025.Projet_certif_google.Trajet3`
GROUP BY
  season, usertype
ORDER BY
  season, usertype;

--Nombres de trajet et moyenne par from_station_name, par usertype
SELECT
  from_station_name,
  usertype,
  AVG(Trip_duration) AS moy_trip_duration,
  COUNT(*) AS n
FROM
  `caduran2025.Projet_certif_google.Trajet3`
GROUP BY
  from_station_name, usertype
ORDER BY
  from_station_name, usertype;

--Pour affiner les résultats et analyser cette base, on va continuer sur le
logiciel R.
```