

- [L3 MIASHS/Ingémath/METIS](#)
- [Université Paris Cité](#)
- [Année 2024-2025](#)
- [Course Homepage](#)

- [Moodle](#)



Dans une métropole, un exploitant de vélos partagés cherche à archiver des informations concernant les vélos et les stations.

Chaque vélo est identifié par un numéro. Un vélo possède une date de mise en service, un type (électrique/mécanique). Un vélo possède éventuellement une date de retrait de service.

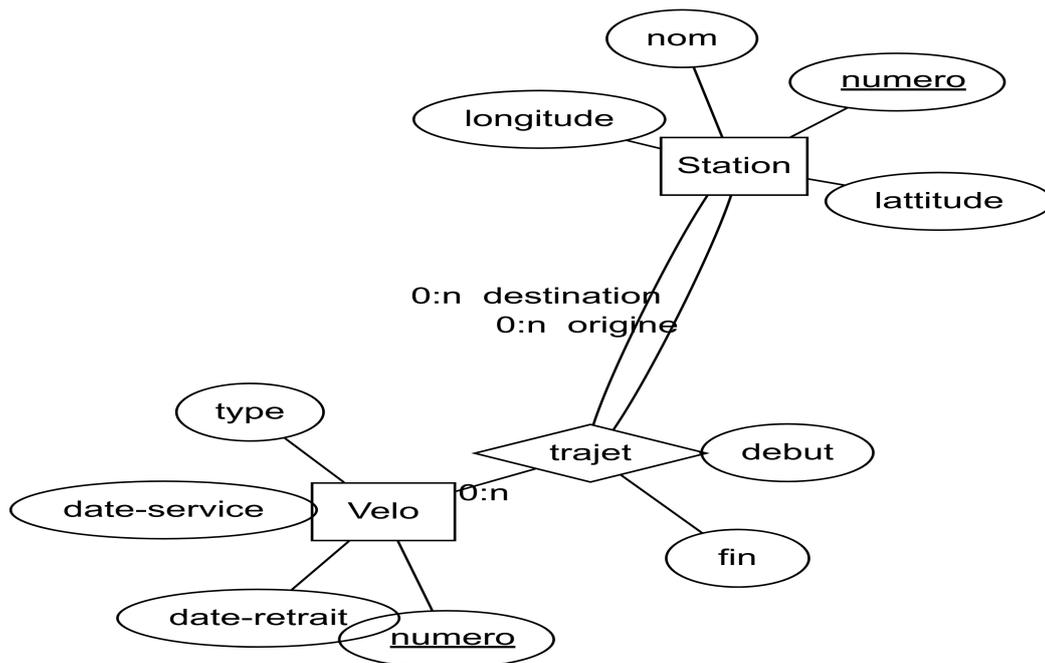
Une station est identifiée par un numéro. Une station possède un nom. Une station possède une position (latitude, longitude) et une altitude. Une station possède un nombre de fixe de bornes d'accrochages.

Un trajet est effectué par un vélo entre un instant (**timestamp**) de départ et un instant d'arrivée. Un trajet part d'une station et arrive à une station (pas toujours différente).

i Question

Proposer un diagramme entité-association correspondant à cette modélisation.

💡 Solution



trajet est une association *ternaire* avec des cardinalités $0:n$. C'est une association *plusieurs-plusieurs*.

i Remarque(s)

Ce schéma ressemble beaucoup à celui des vols dans *nycflights* voir [diapositives](#). Dans les deux schémas, une entité (aéroport, station) participe avec deux rôles distincts à une association

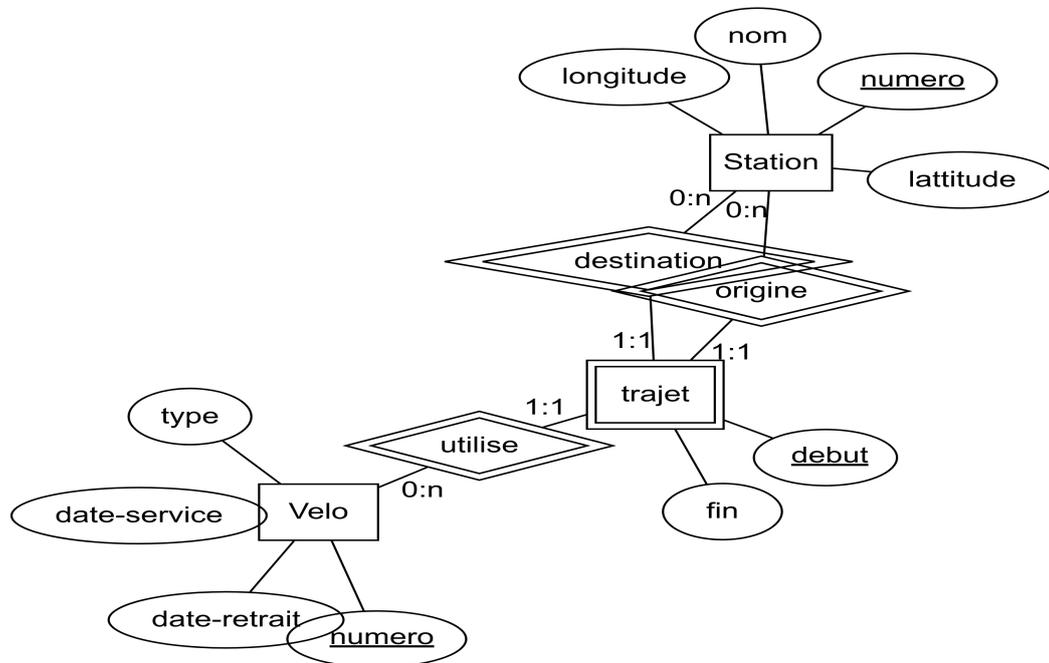
En science des données on parle de schéma en étoile (*star schéma*) : un vol, une mission, un trajet est un *événement* (*event*) auquel participent des objets (vélo, avion, aéroport, station de vélib, etc). L'événement est –à première vue– une instance d'association. Cette association est dans les deux cas une association plusieurs-plusieurs et son *arité* est supérieure ou égale à trois.

Il n'est pas utile de définir une entité position munie deux attributs latitude et longitude. Pour lier position et station on est conduit à définir une association avec des cardinalités 1 :1 (côté station) et 0 :1 (côté position). Ce genre d'association gagne à être représenté par des attributs (rester simple, invoquer le rasoir d'Occam).

i Question

Si besoin, transformez votre schéma de façon à ce qu'il ne comporte plus d'associations d'arité supérieure ou égale à 3, ou d'associations plusieurs-plusieurs (0 :n pour plusieurs entités participantes).

💡 Solution



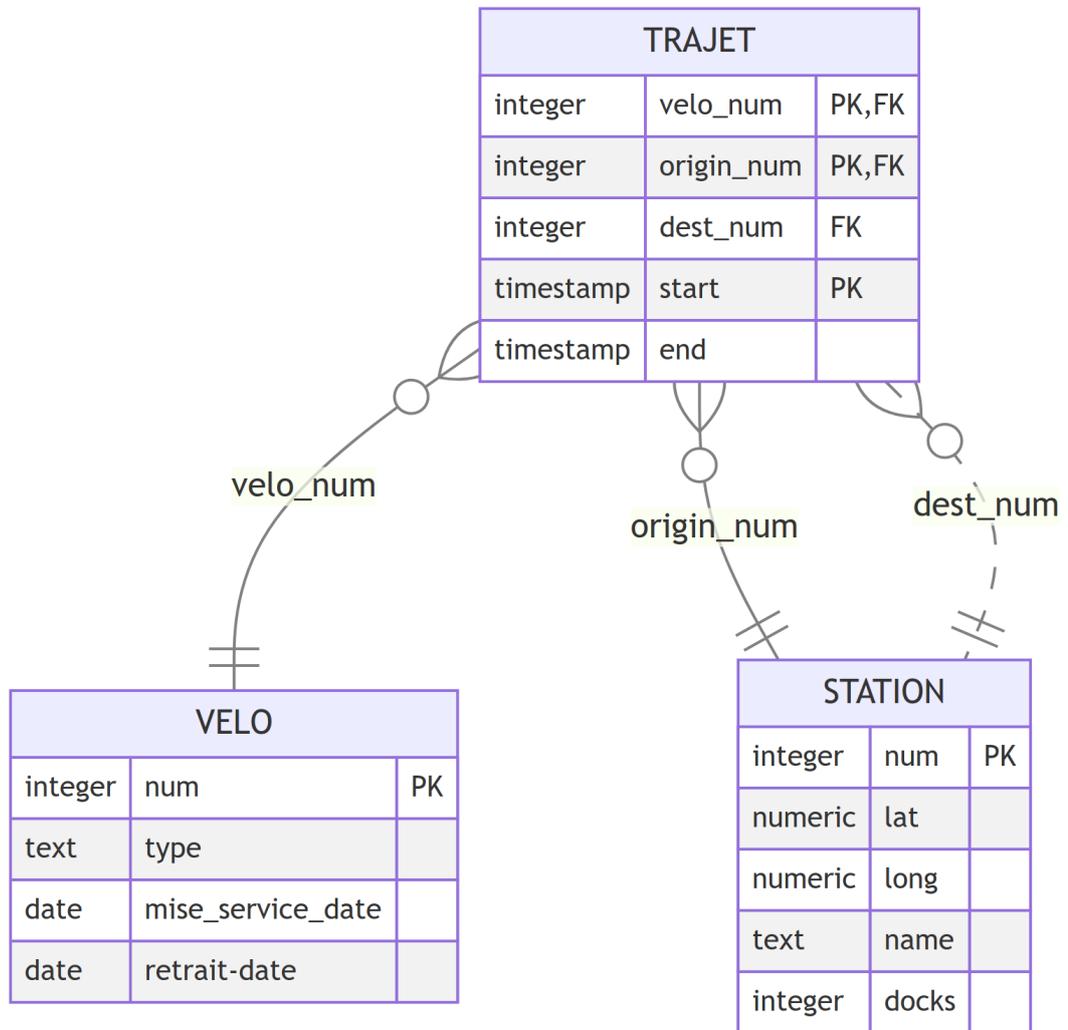
i Remarque

Considérer **Trajet** comme une entité est assez tentant. Si on le fait d'emblée, il faut se demander ce qui identifie les instances de cette entité. Beaucoup sont tentés d'introduire un identifiant artificiel `trajet_id` (de fait ce genre d'identifiant artificiel est facile à introduire lors de l'implémentation en PostGres, en MySQL, ou autre). On perd de vue que ce qui identifie un trajet, c'est un vélo, une station de départ et une heure de départ. Vu comme cela, **Trajet** apparaît comme une association ternaire plusieurs-plusieurs, qu'on se dépêche de transformer en entité faible. L'heure de départ est un identifiant relatif.

i Question

Proposer une traduction en pattes de corbeau du diagramme EA proposé en réponse à la première question.

💡 Solution



i Remarque

Le lien qui correspond à la contrainte référentielle entre TRAJET.dest_num et STATION.num n'est pas identifiant.

i Question

Explicitez d'éventuelles contraintes externes (si vous l'avez déjà fait en marge du diagramme entité-association, contentez-vous d'un renvoi).

Solution

1. La date de retrait de service d'un vélo ne peut pas précéder sa mise en service (**CHECK**)
2. Un vélo ne peut pas être utilisé pour un trajet s'il n'est pas en service (contrainte *multitable*, réalisable avec une fonction qui renvoie l'intervalle pendant lequel un vélo est en service et une contrainte **CHECK** qui vérifie que (**start,end**) est inclus dans l'intervalle de service)
3. Le début d'un trajet doit précéder sa fin (**CHECK**)
4. Un même vélo ne peut pas effectuer deux trajets simultanément (**EXCLUDE**)

Remarque

Les contraintes référentielles, les contraintes de clé sont déjà décrites par le diagramme. Il est inutile de les répéter. Ce ne sont pas des contraintes externes.

Attention

Dans la suite, vous formulerez les requêtes dans le schéma relationnel défini par votre schéma en pattes de corbeau.

Question

Pour chaque station, chaque heure de la journée, comptez le nombre de trajets partis de la station.

 PostgreSQL propose un type **timestamp** (qui permet de représenter les instants à la microseconde près). PostgreSQL propose une fonction **date_part(text, timestamp)** qui permet d'extraire les différents composants d'un objet de type **timestamp**. **date_part('month', timestamp)** extrait le mois grégorien (un entier de 1 à 12), **date_part('hour', timestamp)** extrait l'heure du jour (un entier de 0 à 23).

Solution

```
SELECT
  tr.origin_num, date_part('hour', start) AS heure,
  count(*) AS n_trajet
FROM
  trajet tr
GROUP BY
  tr.origin_num, date_part('hour', start) ;
```

Question

Pour chaque vélo, comptez le nombre de trajets de plus d'une heure effectués.

 En PostgreSQL, la différence de deux objets de type **timestamp** est un objet de type **interval** (durée). Pour définir une constante correspondant à une durée d'un jour, d'un mois, d'une heure, vous pouvez utiliser la syntaxe **'1 day'::interval**, **'1 month'::interval**, **'1 hour'::interval**.

💡 Solution

```
SELECT
  tr.velo_num,
  count(*) AS nb_trajets
FROM
  trajet tr
WHERE
  (tr.end IS NOT NULL AND (tr.start - tr.end) > '1 hour'::interval) OR
  (tr.end IS NULL AND (current_timestamp - tr.start) > '1 hour'::interval)
GROUP BY
  tr.velo_num ;
```

i Question

Pour chaque jour de la semaine, listez les quatre couples (station de départ, station d'arrivée) utilisés par le plus grand nombre de trajets ce jour de la semaine là.

💡 En PostgreSQL, pour extraire le jour de la semaine d'un objet `ts` de type `timestamp`, vous pouvez utiliser `EXTRACT(DOW FROM ts)`. Le résultat est un entier entre 0 et 6, 0 pour dimanche, 1 pour lundi, ..., 7 pour samedi.

```
postgres=# SELECT
  current_timestamp as maintenant ,
  EXTRACT(DOW FROM current_timestamp + '1 day'::interval) as dow_demain ;
```

maintenant		dow_demain
2024-12-08 10:55:05.672936+01		1

💡 Solution

```
WITH R AS
(
  SELECT
    EXTRACT(DOW FROM tr.start) AS jds,
    tr.origin_num, tr.dest_num,
    COUNT(*) AS nb_trajets
  FROM
    trajet tr
  GROUP BY
    EXTRACT(DOW FROM tr.start),
    tr.origin_num,
    tr.dest_num
), S AS
(
  SELECT
    R.*,
    RANK() OVER (PARTITION BY R.jds ORDER BY R.nb_trajets DESC) AS rnk
  FROM
    R
)

SELECT
  S.*
FROM
  S
WHERE S.rnk <= 4 ;
```

i Question

Soit $\mathcal{A} = \{A, B, C, D, E, H, S, T\}$ un schéma.

Soit $\Sigma = \{A, E\} \rightarrow \{H\}, \{S, T\} \rightarrow \{C, E\}, \{S\} \rightarrow \{A\}, \{B\} \rightarrow \{T\}$

- Quelle est la fermeture de $\{B, S, E\}$?
- Quelles sont les clés de \mathcal{A}, Σ ?
- \mathcal{A}, Σ est elle forme normale de Boyce-Codd ?

💡 Solution

- $[B, S, E]_{\Sigma}^{+} = \{B, S, E, T, C, A, H\} = \{A, B, C, E, H, S, T\}$
- Toute clé contient B, D, S qui est une super-clé $[B, D, S]_{\Sigma}^{+} = \{B, D, S, A, T, C, E, H\}$. C'est la seule clé.
- Le schéma n'est pas en FNBC, toutes les DF de Σ ont un déterminant qui n'est pas une super-clé.