1. Afficher la liste des utilisateurs triés par ordre alphabétique (order)

SELECT name FROM utilisateur ORDER BY nom ASC

2. Afficher les trajets créés par un utilisateur (where)

SELECT id\_trajet FROM trajet WHERE id\_user = XXXX

3. Afficher les différents types de trajets (distinct)

SELECT DISTINCT type\_trajet FROM trajet

4. Lister les utilisateurs d’un trajet

SELECT nom FROM utilisateurs JOIN trajet ON trajet.id\_user = utilisateur.id\_user WHERE trajet.id\_trajet =  « 321658497849651 »;

5. Lister les trajets auxquels un utilisateur peut s’inscrire aujourd’hui (…)

SELECT id\_trajet FROM trajet WHERE heure\_depart > NOW() AND nombre\_places > 0;

6. Compter le nombre d’étapes d’un trajet (group)

SELECT COUNT (id\_trajet) FROM trajet

JOIN etapes ON trajet.id\_trajet = etape.id\_trajet

WHERE trajet.id\_trajet = “1234567890” GROUP BY trajet.id\_trajet;

7. Lister les trajets d’un utilisateur avec le nombre d’invités associés (left join)

SELECT trajet.id\_trajet, COUNT(s\_inscrit.id\_utilisateur) FROM trajet LEFT JOIN s\_inscrit ON trajet.id\_trajet = s\_inscrit.id\_trajet GROUP BY trajet.id\_trajet

1. Lister des trajets qui démarrent de Lille, qui passent par Douai ou par Amiens, qui arrivent à Paris le 21/03/2018 ou le 23/03/2018 et qui ne concernent pas un personnel de l’IMT Lille Douai.

SELECT id\_trajet FROM trajet

JOIN etape ON etape.id\_trajet = trajet.id\_trajet

JOIN utilisateur ON trajet.id\_user = utilisateur.id\_user

WHERE lieu\_depart = “Lille” AND lieu\_arrivee = “Paris” AND (etape.ville = « Douai » OR etape.ville = « Amiens »)

AND ( (DAYOFMONTH(heure\_depart) = 21 AND MONTH(heure\_depart) = 3 AND YEAR(heure\_depart) = 2018)

OR DAYOFMONTH(heure\_depart) = 23 AND MONTH(heure\_depart) = 3 AND YEAR(heure\_depart) = 2018) )

AND utilisateur.service = « Etudiant » ;

2. On veut afficher 3 listes pour un mois : les grands trajets (>100km), les trajets moyens (entre 50 et 100 km) et les petits trajets (<50km).

SELECT id\_trajet FROM trajet WHERE MONTH(heure\_depart) = “02” AND distance > 100;

SELECT id\_trajet FROM trajet WHERE MONTH(heure \_depart) = “02” AND distance < 50;

SELECT id\_trajet FROM trajet WHERE MONTH(heure \_depart) = “02” AND distance < 100 AND distance > 50;

3. On cherche les personnes dont le mail contient ‘imt-lille-douai’ dans la liste des utilisateurs.

SELECT nom, prenom FROM utilisateur WHERE email LIKE “% imt-lille-douai%”;

4. On veut afficher les trajets avec leur durée. Ajouter une date/heure de départ et date/heure d’arrivée si besoin.

SELECT id\_trajet, duree FROM trajet

5. Pour que le site soit complet, on veut afficher quelques statistiques :

* Le nombre de trajets par jour, par mois et par année

SELECT COUNT(id\_trajet), heure\_depart FROM trajet ORDER BY DAYOFMONTH(heure\_depart) ASC;

SELECT COUNT(id\_trajet), heure\_depart FROM trajet ORDER BY MONTH(heure\_depart) ASC;

SELECT COUNT(id\_trajet), heure\_depart FROM trajet ORDER BY YEAR(heure\_depart) ASC;

* La distance totale parcourue par jour, par mois et par année

SELECT SUM(distance) FROM trajet

GROUP BY ( DAYOFMONTH(trajet.heure\_depart) )

* Le top 10 des personnes créant des trajets

SELECT COUNT(id\_trajet) AS nbDeTrajet, prenom, nom

FROM utilisateur

JOIN trajet ON utilisateur.id\_user = trajet.id\_user

GROUP BY id\_user

ORDER BY nbDeTrajet DESC LIMIT 10 ;

* Le top 10 des personnes participant à des trajets

SELECT prenom, nom, nombre\_trajets\_realises

FROM utilisateur

ORDER BY nombre\_trajets\_realises DESC LIMIT 10 ;

* Le top 10 des villes (ou site) départs de trajets

SELECT COUNT(id\_trajet) AS nbTrajets, nom\_ville

FROM ville

JOIN trajet ON trajet.lieu\_depart = ville.nom\_ville

GROUP BY nom\_ville

ORDER BY nbTrajets DESC LIMIT 10 ;

* Le top 10 des villes (ou site) destinations de trajets

SELECT COUNT(id\_trajet) AS nbTrajets, nom\_ville

FROM ville

JOIN trajet ON trajet.lieu\_arrivee = ville.nom\_ville

GROUP BY nom\_ville

ORDER BY nbTrajets DESC LIMIT 10 ;

* Le nombre moyen de personnes par trajet

SELECT AVG( SELECT COUNT(participe.id\_user) FROM participe) AS avgPersonne, prenom, nom

FROM utilisateur

JOIN trajet ON participe.id\_trajet = trajet.id\_trajet

GROUP BY id\_trajet

ORDER BY avgPersonne DESC LIMIT 10 ;

* Le nombre moyen d’étapes par trajet

SELECT AVG( SELECT COUNT(etape.nom\_ville) FROM etape) AS avgEtape

FROM etape

JOIN trajet ON etape.id\_trajet = trajet.id\_trajet

GROUP BY id\_trajet

ORDER BY avgEtape DESC LIMIT 10 ;