**Requêtes de profilages ou exploratoires de données**

1. **Requêtes sur les données de la BDTOPO de l’IGN**

/\* Lecture du contenu de la table troncon41\*/

SELECT \* FROM ign.troncons41;

/\* Ajout des nouvelles colonnes et de leurs mises à jour ou changement du type de données de des colonnes date\_creation et date\_update de string en timestamp\*/

ALTER TABLE ign.troncons41 ADD COLUMN date\_creation timestamp;

UPDATE ign.troncons41 set date\_creation = TO\_TIMESTAMP(date\_creat,'YYYY-MM-DD HH24:MI:SS');

ALTER TABLE ign.troncons41 ADD COLUMN date\_update timestamp;

UPDATE ign.troncons41 set date\_update = TO\_TIMESTAMP(date\_maj,'YYYY-MM-DD HH24:MI:SS');

/\* Lecture du contenu des colonnes date\_maj, date\_update de la table troncon41\*/

SELECT date\_maj, date\_update FROM ign.troncons41;

/\* Ajout d'une nouvelle colonne et de sa mise à jour ou changement du type de données de date\_update\_a\_m\_j\*/

ALTER TABLE ign.troncons41 ADD COLUMN date\_update\_a\_m\_j date;

UPDATE ign.troncons41 set date\_update\_a\_m\_j = TO\_DATE(date\_maj,'YYYY-MM-DD');

/\* Lecture du contenu des colonnes date\_maj, date\_update, date\_update\_a\_m\_j de la table troncon41\*/

SELECT date\_update\_a\_m\_j FROM ign.troncons41;

SELECT date\_maj, date\_update, date\_update\_a\_m\_j FROM ign.troncons41;

/\* Ajout d'une nouvelle colonne et de sa mise à jour ou changement du type de données de date\_update\_a\_m\*/

ALTER TABLE ign.troncons41 ADD COLUMN date\_update\_a\_m date;

UPDATE ign.troncons41 set date\_update\_a\_m = TO\_TIMESTAMP(date\_maj,'YYYY-MM');

/\* Lecture du contenu des colonnes date\_maj, date\_update\_a\_m\_j, date\_update\_a\_m de la table troncon41\*/

SELECT date\_update\_a\_m FROM ign.troncons41;

SELECT date\_maj, date\_update\_a\_m\_j, date\_update\_a\_m FROM ign.troncons41;

/\* Ajout d'une nouvelle colonne et de sa mise à jour ou changement du type de données de date\_update\_a\*/

ALTER TABLE ign.troncons41 ADD COLUMN date\_update\_a date;

UPDATE ign.troncons41 set date\_update\_a = TO\_TIMESTAMP(date\_maj,'YYYY');

/\* Lecture du contenu des colonnes date\_maj, date\_update\_a\_m\_j, date\_update\_a\_m, date\_update\_a de la table troncon41\*/

SELECT date\_update\_a FROM ign.troncons41;

SELECT date\_maj, date\_update\_a\_m\_j, date\_update\_a\_m, date\_update\_a FROM ign.troncons41;

/\* Lecture du contenu des colonnes de la table troncon41\*/

SELECT \* FROM ign.troncons41;

/\* mise à jour ou changement du type de données de des colonnes date\_creation et date\_update\*/

ALTER TABLE ign.troncons41 ADD COLUMN date\_update timestamp;

UPDATE ign.troncons41 set date\_update = TO\_TIMESTAMP(date\_maj,'MM-DD');

/\* Ajout d'une nouvelle colonne et de sa mise à jour ou changement du type de données de date\_update\_m\*/

ALTER TABLE ign.troncons41 ADD COLUMN date\_update\_m timestamp;

UPDATE ign.troncons41 set date\_update\_m = TO\_TIMESTAMP(date\_maj,'MM');

/\* Ajout d'une nouvelle colonne et de sa mise à jour ou changement du type de données de date\_update\_d\*/

ALTER TABLE ign.troncons41 ADD COLUMN date\_update\_d timestamp;

UPDATE ign.troncons41 set date\_update\_d = TO\_TIMESTAMP(date\_maj,'DD');

/\* Lecture du contenu des colonnes de la table troncon41\*/

SELECT \* FROM ign.troncons41;

/\* Lecture le nombre d'attributs par nature de la table troncon41\*/

SELECT nature, COUNT(id) FROM ign.troncons41 GROUP BY nature;

/\* Lecture le nombre d'attributs de la conne date\_maj null (non remplit) par nature de la table troncon41\*/

SELECT nature, COUNT(id) FROM ign.troncons41 WHERE date\_maj IS NULL GROUP BY nature;

/\* Lecture le nombre d'attributs de la colonne date\_maj non null (bien remplit) par nature de la table troncon41\*/

SELECT nature, COUNT(id) FROM ign.troncons41 WHERE date\_maj IS NOT NULL GROUP BY nature;

/\* Lecture du contenu des colonnes de la table troncon41\*/

SELECT \* FROM ign.troncons41;

/\* Lecture le nombre d'éléments de piste cyclable de la table troncon41\*/

SELECT cyclable, COUNT(id) FROM ign.troncons41 WHERE nature='Piste cyclable' GROUP BY cyclable;

/\* Lecture le nombre d'éléments de piste cyclable de la colonne date\_maj null (non remplit) de la table troncon41\*/

SELECT cyclable, COUNT(id) FROM ign.troncons41 WHERE date\_maj IS NULL AND nature='Piste cyclable' GROUP BY cyclable;

/\* Lecture le nombre d'éléments de piste cyclable de la colonne date\_maj non null (bien remplit) de la table troncon41\*/

SELECT cyclable, COUNT(id) FROM ign.troncons41 WHERE date\_maj IS NOT NULL AND nature='Piste cyclable' GROUP BY cyclable;

/\* Lecture le nombre d'attributs ou éléments et la date de Création, grouper par date de Création de la table troncon41\*/

SELECT COUNT (id), date\_creat FROM ign.troncons41 GROUP BY date\_creat;

/\* Lecture du contenu des colonnes nature, date\_creat, date\_maj de la table troncon41 par ordre croissant de nature, date\_creat, date\_maj\*/

SELECT nature, date\_creat, date\_maj FROM ign.troncons41 ORDER BY nature, date\_creat, date\_maj;

/\* Lecture du contenu des colonnes nature, date Minimale Création d'objet, date Maximale Création d'objet de la table troncon41 grouper par nature\*/

SELECT nature, MIN(date\_creat), MAX(date\_creat) FROM ign.troncons41 GROUP BY nature;

/\* Lecture du contenu des colonnes nature, date Minimale de mise à jour d'objet, date Maximale de mise à jour d'objet de la table troncon41 grouper par nature\*/

SELECT nature, MIN(date\_maj), MAX(date\_maj) FROM ign.troncons41 GROUP BY nature;

/\* Lecture du nombre d'attributs par nature, date Minimale Création d'objet, date Maximale Création d'objet de la table troncon41 grouper par nature\*/

SELECT nature, COUNT(id), MIN(date\_creat), MAX(date\_creat) FROM ign.troncons41 WHERE date\_creat in (SELECT MIN(date\_creat) FROM ign.troncons41 GROUP BY nature) GROUP BY nature;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 GROUP BY date\_maj ORDER BY c DESC;

/\* Lecture du contenu chaque de type de nature, date Minimale maj d'objet de la table troncon41 grouper et ordonner par date mise à jour \*/

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Piste cyclable' GROUP BY date\_maj ORDER BY date\_maj ASC;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Chemin' GROUP BY date\_maj ORDER BY date\_maj ASC;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Bretelle' GROUP BY date\_maj ORDER BY date\_maj ASC;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Escalier' GROUP BY date\_maj ORDER BY date\_maj ASC;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Rond-point' GROUP BY date\_maj ORDER BY date\_maj ASC;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Route Ã  1 chaussÃ©e' GROUP BY date\_maj ORDER BY date\_maj ASC;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Route Ã  2 chaussÃ©es' GROUP BY date\_maj ORDER BY date\_maj ASC;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Route empierrÃ©e' GROUP BY date\_maj ORDER BY date\_maj ASC;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Sentier' GROUP BY date\_maj ORDER BY date\_maj ASC;

SELECT date\_maj, COUNT(date\_maj) AS c FROM ign.troncons41 WHERE nature='Type autoroutier' GROUP BY date\_maj ORDER BY date\_maj ASC;

/\* Lecture du contenu chaque de type de nature, date Minimale Création d'objet de la table troncon41 grouper et ordonner par date mise à jour \*/

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='Piste cyclable' GROUP BY date\_creat ORDER BY date\_creat ASC;

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='CheMIN' GROUP BY date\_creat ORDER BY date\_creat ASC;

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='Bretelle' GROUP BY date\_creat ORDER BY date\_creat ASC;

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='Escalier' GROUP BY date\_creat ORDER BY date\_creat ASC;

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='Rond-point' GROUP BY date\_creat ORDER BY date\_creat ASC;

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='Route Ã  1 chaussÃ©e' GROUP BY date\_creat ORDER BY date\_creat ASC;

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='Route Ã  2 chaussÃ©es' GROUP BY date\_creat ORDER BY date\_creat ASC;

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='Route empierrÃ©e' GROUP BY date\_creat ORDER BY date\_creat ASC;

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='Sentier' GROUP BY date\_creat ORDER BY date\_creat ASC;

SELECT date\_creat, COUNT(date\_creat) AS c FROM ign.troncons41 WHERE nature='Type autoroutier' GROUP BY date\_creat ORDER BY date\_creat ASC;

/\* Lecture du nombre d'attributs par nature, date Minimale mise à jour d'objet, date Maximale mise à jour d'objet de la table troncon41 grouper par nature suivant aussi par date de mise à jour\*/

SELECT nature, COUNT(nature), MIN(date\_maj), MAX(date\_maj) FROM ign.troncons41 WHERE date\_maj in (SELECT MIN(date\_maj) FROM ign.troncons41 GROUP BY nature) GROUP BY nature;

SELECT nature, COUNT(nature), MIN(date\_maj), MAX(date\_maj) FROM ign.troncons41 WHERE date\_maj in (SELECT MIN(date\_maj) FROM ign.troncons41 GROUP BY nature) GROUP BY nature;

/\* Création d'une fonction de calcul de la différence des dates\*/

CREATE OR REPLACE FUNCTION datediff(timestamp, timestamp)

RETURNS int

LANGUAGE sql

AS

$$

SELECT CAST($1 AS date) - CAST($2 AS date) AS DateDifference

$$;

/\* Lecture du contenu de la table troncon41\*/

SELECT \* FROM ign.troncons41;

/\* Lecture le nombre d'attributs suivant la date de Création et la date de mise à jour grouper et ordonner par la date de Création et la date de mise à jour de la table troncon41\*/

SELECT COUNT (distinct id), date\_creat, date\_maj FROM ign.troncons41 GROUP BY date\_creat, date\_maj ORDER BY date\_creat, date\_maj;

SELECT COUNT (distinct id), date\_creat, date\_maj FROM ign.troncons41 GROUP BY date\_creat, date\_maj ORDER BY date\_creat, date\_maj;

1. **Requêtes sur les données d’OSM**

/\* Lecture du contenu des colonnes de la table lastversionway\*/

SELECT \* FROM osm.lastversionway;

/\* Lecture du contenu des colonnes de la table lastversionway dont le tag contient la clé highway\*/

SELECT tags ? 'highway' FROM osm.lastversionway;

/\* Lecture du nombre de version de modification par ordre croissant pour chaque identifiant de la table lastversionway dont le tag contient la clé highway\*/

SELECT nbmodif, COUNT(\*) FROM (SELECT id, SUM(vway-1) AS nbmodif FROM osm.lastversionway WHERE tags ? 'highway' GROUP BY id) modif GROUP BY nbmodif ORDER BY nbmodif ASC;

/\* Lecture du nombre de version de modification par ordre croissant pour chaque highway de la table lastversionway dont le tag contient la clé highway\*/

SELECT nbmodif, COUNT(\*) FROM (SELECT tags ? 'highway', SUM(vway-1) AS nbmodif FROM osm.lastversionway GROUP BY tags ? 'highway') modif GROUP BY nbmodif ORDER BY nbmodif ASC;

/\* Lecture du nombre de version de modification par identifiant de la table lastversionway dont le tag contient la clé highway\*/

SELECT id, SUM(vway-1) AS nbmodif FROM osm.lastversionway WHERE tags ? 'highway' GROUP BY id;

/\* Lecture du nombre de version de modification par identifiant de chaque valeur de highway de la table lastversionway dont le tag contient la clé highway\*/

SELECT id, tags -> 'highway' AS highway, SUM(vway-1) AS nbmodif FROM osm.lastversionway WHERE tags ? 'highway' GROUP BY id, tags -> 'highway';

/\* Lecture du nombre de version et du nombre de version de modification pour la valeur cycleway de la table lastversionway\*/

SELECT svals('highway=>cycleway'), SUM(vway), SUM(vway-1) FROM osm.lastversionway;

/\* Lecture du nombre de version et du nombre de version de modification pour la valeur cycleway de la table lastversionway\*/

SELECT svals('highway=>cycleway'), SUM(vway) AS creation\_vway, SUM(vway-1) AS vway FROM osm.lastversionway;

/\* Lecture du nombre de version et du nombre de version de modification pour la valeur cycleway de la table lastversionway par groupe et d’ordre croissant de idway\*/

SELECT svals('highway=>cycleway'), SUM(vway), SUM(vway-1) FROM osm.lastversionway GROUP BY idway, id ORDER BY idway;

/\* Lecture des identifiant way, des identifiant correspondants, de chaque version, sa version antérieure pour la valeur cycleway de la table lastversionway par ordre croissant de idway\*/

SELECT idway, id, vway, vway-1, svals('highway=>cycleway') FROM osm.lastversionway ORDER BY idway;

/\* Lecture des identifiant way, des identifiant correspondants, de chaque version pour la valeur cycleway de la table lastversionway par ordre croissant de idway\*/

SELECT idway, id, vway, svals('highway=>cycleway') FROM osm.lastversionway ORDER BY idway;

/\* Lecture des identifiant way, des identifiant correspondants, de chaque version, sa version antérieure pour la valeur cycleway de la table lastversionway\*/

SELECT idway, id, vway, vway-1, svals('highway=>cycleway') FROM osm.lastversionway;

/\* Lecture des identifiant way, des identifiant correspondants, de chaque version pour la valeur cycleway de la table lastversionway\*/

SELECT idway, id, vway, svals('highway=>cycleway') FROM osm.lastversionway;

/\* Lecture par groupe d’ordre de clé tags, du nombre d’éléments ou type de la clé du champ tag de la table lastversionway\*/

SELECT key, COUNT(\*) FROM

(SELECT (each(tags)).key FROM osm.lastversionway) AS stat

GROUP BY key

ORDER BY COUNT DESC, key;

/\* Lecture du nombre d’éléments ou type de la valeur cycleway de la table lastversionway par groupe d’ordre de la valeur cycleway\*/

SELECT value, COUNT(\*) FROM

(SELECT (each('highway=>cycleway')).value FROM osm.lastversionway) AS stat

GROUP BY value

ORDER BY COUNT DESC, value;

/\* Création de la vue du nombre de modification en récupérant (identifiants, nombre de modifications et les valeurs d’highway) à partir de la table lastversionway pour toute clé highway\*/

create view vue\_nombre\_modification AS SELECT id, vway-1 AS nombre\_modif, tags -> 'highway' AS highway FROM osm.lastversionway WHERE tags ? 'highway';

/\* Lecture du contenu des colonnes de la vue vue\_nombre\_modification \*/

SELECT \* FROM osm.vue\_nombre\_modification ;

SELECT highway, nombre\_modif FROM osm.vue\_nombre\_modification ;

/\* Lecture du contenu des colonnes de la vue vue\_nombre\_modification pour la nature bridleway \*/

SELECT \* FROM osm.vue\_nombre\_modification WHERE highway='bridleway';

/\* Lecture du nombre de modification par highway de la vue vue\_nombre\_modification \*/

SELECT highway, SUM(nombre\_modif) FROM osm.vue\_nombre\_modification GROUP BY highway;

/\* Lecture du nombre d’entité modifié, du nombre Minimal, Maximal et la moyenne de modification par highway de la vue vue\_nombre\_modification \*/

SELECT highway, MIN(nombre\_modif) AS Minimum, MAX(nombre\_modif) AS Maximum, COUNT(id) AS nombre\_entité, AVG(nombre\_modif) AS moyenne FROM osm.vue\_nombre\_modification GROUP BY highway ;

/\* Lecture du nombre d’entité modifié, du nombre Minimal, Maximal et la moyenne de modification par highway de la vue vue\_nombre\_modification \*/

SELECT highway, MIN(nombre\_modif) AS Minimum, MAX(nombre\_modif) AS Maximum, COUNT(id) AS nombre\_entité, ROUND(AVG(nombre\_modif),2) AS moyenne FROM osm.vue\_nombre\_modification GROUP BY highway ;

/\* Lecture du mode du nombre de modification par highway de la vue vue\_nombre\_modification\*/

SELECT highway, mode() WITHIN GROUP(ORDER BY nombre\_modif) AS mode\_modif FROM osm.vue\_nombre\_modification GROUP BY highway;

/\* Lecture du nombre d’entité modifié, du nombre Minimal, Maximal, la moyenne de modification et le mode du nombre de modification par highway de la vue vue\_nombre\_modification \*/

SELECT highway, MIN(nombre\_modif) AS Minimum, MAX(nombre\_modif) AS Maximum, COUNT(id) AS nombre\_entité, ROUND(AVG(nombre\_modif)) AS moyenne, MODE() WITHIN GROUP(ORDER BY nombre\_modif) AS MODE\_modif FROM osm.vue\_nombre\_modification GROUP BY highway;

/\* Lecture du nombre d’entité modifiée, par le nombre croissant de modification du type d’autoroutier (motorway) par highway de la vue vue\_nombre\_modification\*/

SELECT COUNT(\*), nombre\_modif FROM osm.vue\_nombre\_modification WHERE highway='motorway' GROUP BY nombre\_modif;

/\* Lecture De chaque version de modification, sa version supérieure et sa période de modification par identifiant et valeur d’highway la table way\*/

SELECT v1.id AS identifiant, v1.tags -> 'highway' AS highway, v1.vway AS version\_ant, v2.vway AS version\_act, (v2.datemodif-v1.datemodif) AS datedif FROM osm.way v1 join osm.way v2 ON v1.id=v2.id WHERE v2.vway=v1.vway+1 AND v1.vway<v2.vway AND v1.tags ? 'highway' ORDER BY v1.id, v1.tags -> 'highway', v1.vway ASC;

/\* Lecture du contenu des colonnes de la table way\*/

SELECT \* FROM osm.way;

SELECT \* FROM osm.way AS w WHERE idway=49861374 AND w.tags ? 'highway';

/\* Lecture contenu des colonnes du highway de l’élément dont idway=49861374 de la table way\*/

SELECT id AS identifiant, tags -> 'highway' AS highway, vway AS version\_ant, datemodif AS datedif FROM osm.way AS w WHERE idway=49861374 AND w.tags ? 'highway';

/\* Lecture du nombre d’entité par highway dont la date de modification n’est pas remplis ou vide de la table way\*/

SELECT tags -> 'highway' AS highway, COUNT(datemodif) AS champs FROM osm.way WHERE datemodif is null GROUP BY tags -> 'highway';

/\* Lecture du nombre d’entité par highway dont la date de modification est bien remplis ou non vide de la table way\*/

SELECT tags -> 'highway' AS highway, COUNT(datemodif) AS champs FROM osm.way WHERE datemodif is not null GROUP BY tags -> 'highway';

/\* Lecture du contenu des colonnes de la table way\*/

SELECT \* FROM osm.way;

/\* Création de la vue des périodes de création des objets en récupérant (identifiants de la version, identifiants, numéros de la version, dates de modification de la version et les valeurs d’highway) à partir de la table way pour toute clé highway dont le numéro de version est 1\*/

create view osm.vue\_periodes\_creation AS SELECT idway AS identifiant\_version, id AS identifiant, vway AS num\_version, datemodif AS date\_modif, tags -> 'highway' AS highway FROM osm.way WHERE tags ? 'highway' AND vway=1;

/\* Lecture du contenu des colonnes de la vue vue\_periodes\_creation \*/

SELECT COUNT(\*) FROM osm.vue\_periodes\_creation;

/\* Lecture du contenu des colonnes de la table lastversionway\*/

SELECT highway, COUNT(identifiant), MIN(date\_modif), MAX(date\_modif) FROM osm.vue\_periodes\_creation WHERE date\_modif IN (SELECT MIN(date\_modif) FROM osm.vue\_periodes\_creation GROUP BY highway) GROUP BY highway ORDER BY COUNT desc;