--Exercice 3

psql -h licinfo.univ-jfc.fr -5432 -U l3info08 -d l3info08

--Exercice 4

--création de la base de données

create database JOSyfney08 encoding ‘UTF8’;

--Exercice 5

--connexion à la base de données

\c josydney08;

--Exercice 6

--création des tables

create table sport(

code\_sport int not null,

intitule varchar(40),

constraint cle\_sport primary key (code\_sport)

);

create table discipline(

code\_discipline int not null,

intitule varchar(40),

code\_sport int,

constraint cle\_discipline primary key (code\_discipline),

foreign key (code\_sport) references sport(code\_sport)

);

create table sportif(

num\_licence int not null,

nom varchar(20) not null,

prenom varchar(20),

date\_naiss Date,

pays varchar(20),

code\_sport int,

constraint cle\_sportif primary key (num\_licence),

foreign key (code\_sport) references sport(code\_sport)

);

create table equipe(

num\_equipe int not null,

denomination varchar(128) not null,

constraint cle\_equipe primary key (num\_equipe)

);

create table appartenir\_equipe (

num\_equipe int,

num\_licence int,

foreign key (num\_equipe) references equipe(num\_equipe),

foreign key (num\_licence) references sportif(num\_licence)

);

create table gagner\_individuel(

num\_licence int,

code\_discipline int,

medaille varchar(10),

foreign key (num\_licence) references sportif(num\_licence),

foreign key (code\_discipline) references discipline(code\_discipline)

);

create table gagner\_collectif(

num\_equipe int,

code\_discipline int,

medaille varchar(10),

foreign key (num\_equipe) references equipe(num\_equipe),

foreign key(code\_discipline) references discipline(code\_discipline)

);

--Exercice 7

--medaille = soit argent soit or soit bronze

--faire attention à la casse du système

alter table gagner\_individuel add check (medaille = 'Or' or medaille = 'Argent' or medaille = 'Bronze');

--Exercice 8

--exécuter le scri^te BDJOSydney à l'aide de \i

--https://gist.github.com/elyeslamine/ba456f61460ce0bcca09#file-bdjosydney-sql

\i Downloads/L3\ JFC/S5/SGBD/BDJOSydney.sql

--Exercice 9

--a: nom sportifs présents dans la base

select nom from sportif;

--b: code et intitule du sport 19

select code\_sport, intitule from sport where code\_sport = 19;

--c: listes des pays d'origine des sportifs ; remarque ?

select nom, pays from sportif;

--Remarque : Tous les sportifs viennent de la France

--d: Intitule des sports dont l'intitule comporte la chaine 'te'

select intitule from sport where intitule like '%te%';

--e: nom des sportifs dont on ne connait pas de la date de naissance

select nom from sportif where date\_naiss is null;

--f: listes des sportifs (nom, prenom, date de naissance) par ordre l'alphabetique

select nom, prenom, date\_naiss from sportif order by nom, prenom asc;

--g: liste des sportifs (nom, prenom, date de naissance) par âge (décroissant)

select nom, prenom, age(date\_naiss) as age from sportif order by age desc;

--h: liste des discipline dont le code sport est compris entre 5 et 10

select distinct code\_sport from discipline where code\_sport between 5 and 10;

--i: listes des sportifs (nom, prenom, date\_naiss) qui avaient moins de 30 ans aux JO

-- date de JO 15/09/2000 - 1/10/2000

select nom, prenom, date\_naiss from sportif

where cast('2000-09-15' - date\_naiss as float)/365 <=30;

--j: corriger l'intitule de la discipline '-100kg' en '100 kg'

update discipline

set intitule = '100 kg'

where intitule = ' -100 kg';

--k: Tony Estanguet est né le 06/05/1978, rajouter cette info dans la base

update sportif

set date\_naiss = '1978-05-06'

where upper(nom) = upper('Estanguet') and upper(prenom) = upper('Tony');

--l: nom des sportifs qui pratiquent l'escrime

select sf.nom

from sport s, sportif sf

where upper(s.intitule) = upper('escrime')

and s.code\_sport = sf.code\_sport;

--m: intitule des diverses disciplines de l'escrime

select d.intitule from discipline d, sport s

where upper(s.intitule) = upper('escrime')

and d.code\_sport = s.code\_sport;

--n: nom des sportifs qui ont gagné une médaille d'or en individuel

select distinct sf.nom

from sportif sf, gagner\_individuel g

where sf.num\_licence = g.num\_licence

and upper(g.medaille) = upper('or');

--o: intitule de la discipline et nom du sportif pour les médaille d'argent individuel

select distinct d.intitule, sf.nom

from sportif sf, discipline d, gagner\_individuel g

where d.code\_sport = sf.code\_sport

and g.num\_licence = sf.num\_licence

and upper(g.medaille) = upper('argent');

--p: nom et date de naissance des sportifs plus jeune que 'Jean-Niel Ferrari'

select nom, date\_naiss

from sportif

where cast(current\_date - date\_naiss as int) <

cast(current\_date -

(select date\_naiss from sportif

where upper(nom) = upper('Ferrari')) as int

);

--q: nom etm édaille obtenu en individuel pour tous les escrimeurs (même ceux qui n'ont pas eu de médaille) ???????

--des gens qui ont obtenu des médaille

select sf.nom, g1.medaille

from sportif sf

left join sport s on s.code\_sport = sf.code\_sport

and upper(s.intitule) = upper('escrime')

left join gagner\_individuel g1 on g1.num\_licence = sf.num\_licence

;

--r: composition d'equipe "Quatre sans barreur"

select sf.nom

from equipe e, sportif sf, appartenir\_equipe a

where a.num\_equipe = e.num\_equipe

and a.num\_licence = sf.num\_licence

and e.denomination = 'Quatre sans barreur';

--s: nom des sportifs qui ont eu une médaille d'or collective

select sf.nom, g.medaille

from sportif sf, gagner\_collectif g, discipline d, sport s

where d.code\_discipline = g.code\_discipline

and s.code\_sport = d.code\_sport

and s.code\_sport = sf.code\_sport

and upper(g.medaille) = upper('or');

--t: dénomination des équipes pour lesquelles on ne connait pas la composition ??????

select e.denomination

from equipe e, sportif sf, appartenir\_equipe a

where e.num\_equipe = a.num\_equipe

and sf.num\_licence = a.num\_licence

group by e.denomination

having count(sf.nom) = 0;

--u: donner le nb de sports aux JO de Sydney

select count(\*) from sport;

--v: obtenir le nb de médailles individuelles par type

select count(g.medaille) as nb\_medaille, s.intitule

from gagner\_individuel g, sport s, discipline d

where d.code\_discipline = g.code\_discipline

and s.code\_sport = d.code\_sport

group by s.intitule;

--w: liste de tous les sportifs (nom) avec pour chacun le nb de médailles individuelles

select sf.nom as nb\_medaille, count(g.medaille)

from sportif sf, gagner\_individuel g

where sf.num\_licence = g.num\_licence

group by sf.nom;

--y: nom des sportif qui ont gagné deux métailles en individuelles

select sf.nom, count(g.medaille) as nb\_medaille

from sportif sf, gagner\_individuel g

where sf.num\_licence = g.num\_licence

group by sf.nom

having count(g.medaille) = 2 ;

--z: pour chaque sport le nb de médaille collectives

select s.intitule, count(g.medaille) as nb

from sport s, gagner\_collectif g, discipline d

where d.code\_sport = s.code\_sport

and g.code\_discipline = d.code\_discipline

group by s.intitule;

--aa: nb total de metaille ?????????????????????????????????

select (count(g1.\*)+count(g2.\*)) as nb\_total

from gagner\_collectif g1, gagner\_individuel g2, discipline d

where g1.code\_discipline = d.code\_discipline

and d.code\_discipline = g2.code\_discipline;