**Exercice d’application 4 :**

create database etudiants;

use etudiants;

create table `etudiants`(

id\_etudiant int primary key AUTO\_INCREMENT not null,

nom\_etudiant varchar(50) not null,

prenom\_etudiant varchar(50) not null,

date\_naissance\_etudiant date,

niveau\_etudiant varchar(50)

)ENGINE=INNODB;

create table `cours`(

id\_cours int primary key AUTO\_INCREMENT not null,

nom\_cours varchar(50) not null,

enseignant\_cours varchar(50)

)ENGINE=INNODB;

create table `examen`(

id\_etudiant int not null,

id\_cours int not null,

primary key (id\_etudiant, id\_cours),

note\_examen float not null

)engine=INNODB;

alter table examen

add constraint foreign key (id\_etudiant) REFERENCES etudiants(id\_etudiant);

alter table examen

add constraint foreign key (id\_cours) REFERENCES cours(id\_cours);

insertion valeurs

insert into etudiants (nom\_etudiant, prenom\_etudiant, date\_naissance\_etudiant, niveau\_etudiant) VALUES

("Casagrande", "Léa", "2000-08-25","M2"),

("Grande", "Maria", "2002-01-13","M2"),

("Mourgues", "Fred", "2003-05-31","M1"),

("Sauviac", "Bernard", "2001-10-15","M2"),

("FourWheels", "Joulou", "2003-11-26","M1");

insert into cours (nom\_cours) VALUES

("Français"),

("Maths"),

("Anglais"),

("Biologie"),

("Sciences de l'ingénieur");

insert into examen (id\_etudiant, id\_cours, note\_examen) VALUES

(1,1,09.5),

(1,2,13),

(1,3,11.25),

(1,4,16.5),

(1,5,14.5),

(2,1,05.5),

(2,2,19),

(2,3,10.5),

(2,4,13.75),

(2,5,10),

(3,1,15),

(3,2,8.75),

(3,3,14.5),

(3,4,12),

(3,5,6.25),

(4,1,17),

(4,2,2.75),

(4,3,10.5),

(4,4,17.5),

(4,5,14),

(5,1,5.25),

(5,2,14.75),

(5,3,4.5),

(5,4,14),

(5,5,16.25);

Requêtes :

select \* from etudiants order by date\_naissance\_etudiant;

select \* from etudiants where niveau\_etudiant='M1' or niveau\_etudiant='M2';

select id\_etudiant from examen where id\_cours=2;

select id\_etudiant from examen where id\_cours=1 or id\_cours=2;

select id\_etudiant, id\_cours, note\_examen as "/20", note\_examen\*2 as "/40" from examen order by id\_etudiant, id\_cours;

select avg(note\_examen) from examen where id\_cours=2;

select count(id\_cours) from examen where id\_etudiant=1;

select count(id\_etudiant) from examen where id\_cours=2;

select avg(note\_examen) as "Moyenne générale" from examen where id\_etudiant=1;

Bonus :

select count(id\_cours) as "nb exams" from examen group by id\_etudiant;

select avg(note\_examen) as "moyenne note" from examen group by id\_etudiant;

select avg(note\_examen) as "moyenne" from examen group by id\_cours;

**Exercice d’application 5 :**

create database immobilier;

use immobilier;

create table `representants`(

id\_representant int primary key AUTO\_INCREMENT not null,

code\_representant varchar(50) not null,

nom\_representant varchar(50) not null,

prenom\_representant varchar(50)

)engine=INNODB;

create table `clients`(

id\_client int primary key AUTO\_INCREMENT not null,

code\_client varchar(50) not null,

nom\_client varchar(50) not null,

prenom\_client varchar(50),

adresse\_client varchar(50),

cp\_client int,

ville\_client varchar(50)

)engine=INNODB;

create table `appartements`(

id\_appartement int primary key AUTO\_INCREMENT not null,

ref\_appartement varchar(50) not null,

superficie\_appartement float,

prix\_appartement float,

secteur\_appartement varchar(50),

id\_representant int,

id\_client int

)engine=INNODB;

alter table appartements

add constraint foreign key (id\_representant) REFERENCES representants(id\_representant);

alter table appartements

add constraint foreign key (id\_client) REFERENCES clients(id\_client);

insert into clients(code\_client,nom\_client,prenom\_client,adresse\_client,cp\_client,ville\_client) VALUES

("code1", "Casagrande", "Léa", "12 rue de la victoire", 31000, "Toulouse"),

("code2", "Grande", "Maria", "3 rue berthelot", 65000, "Tarbes"),

("code3", "Mourgues", "Fred", "1 rue condorcet", 65310, "Odos"),

("code4", "Guidot", "Bernard", "9 avenue de la marne", 81000, "Albi");

insert into representants(code\_representant,nom\_representant,prenom\_representant) VALUES

("A01", "Gina", "Laurent"),

("C04", "Audran", "Corinne"),

("B15", "Berthome", "Jean-Philippe"),

("B15", "Galaud", "Céline");

insert into appartements(ref\_appartement,superficie\_appartement,prix\_appartement,secteur\_appartement,id\_representant,id\_client) VALUES

("356C1", 54, 68000, "Centre",2,4),

("147B8", 174, 225000, "Sud",1,1),

("957R2", 113, 140000, "Ouest",4,2),

("142N4", 250, 309000, "Sud",1,3);

select \* from clients order by nom\_client;

select secteur\_appartement,avg(prix\_appartement) from appartements group by secteur\_appartement;

select secteur\_appartement,count(id\_appartement) from appartements group by secteur\_appartement having count(id\_appartement)>1;

select secteur\_appartement,count(id\_appartement) from appartements where superficie\_appartement>80 group by secteur\_appartement;

select secteur\_appartement,max(prix\_appartement) from appartements group by secteur\_appartement having count(id\_appartement)>0 ;

select \* from clients inner JOIN appartements on clients.id\_client = appartements.id\_client;

select \* from appartements inner join representants on appartements.id\_representant = representants.id\_representant where appartements.secteur\_appartement='Sud' AND representants.nom\_representant='Gina';

select appartements.secteur\_appartement,count(clients.id\_client) from clients inner join appartements on clients.id\_client=appartements.id\_client where clients.ville\_client='Toulouse' group by appartements.secteur\_appartement;

**Exercice 1 Discord**

create database stagiaires;

use stagiaires;

create table `stagiaires`(

id\_stagiaire int not null primary key AUTO\_INCREMENT,

num\_stagiaire varchar(3) not null,

nom\_stagiaire varchar(50) not null,

prenom\_stagiaire varchar(50) not null,

tel\_stagiaire varchar(10) not null

)engine=INNODB;

create table `examens`(

id\_examen int not null primary key AUTO\_INCREMENT,

num\_examen varchar(3) not null,

salle\_examen varchar(3) not null,

date\_examen datetime not null,

type\_examen varchar(1)

)engine=innoDB;

create table `passer\_examen`(

id\_stagiaire int not null,

id\_examen int not null,

primary key(id\_stagiaire, id\_examen),

note\_examen float not null

)engine=INNODB;

alter table passer\_examen

add CONSTRAINT FOREIGN key (id\_stagiaire) REFERENCES stagiaires(id\_stagiaire);

alter table passer\_examen

add CONSTRAINT FOREIGN key (id\_examen) REFERENCES examens(id\_examen);

insert into stagiaires(num\_stagiaire,nom\_stagiaire,prenom\_stagiaire,tel\_stagiaire) VALUES

("S01","Casagrande","Léa","0647586978"),

("S02","Audran","Corinne","0647586912"),

("S03","Sauviac","Laurent","0636584156"),

("S04","Ruiz","Bryan","0647453678");

**Exercice 2 Discord**

create database voyages;

use voyages;

create table `circuits`(

id\_circuit int not null primary key AUTO\_INCREMENT,

destination\_circuit varchar(30) not null,

duree\_circuit float not null

)engine=INNODB;

create table `voyages`(

id\_voyage int not null primary key AUTO\_INCREMENT,

date\_voyage varchar(30) not null,

prix\_voyage float not null,

nb\_places\_voyage int not null,

id\_circuit int not null

)engine=INNODB;

alter table voyages

add CONSTRAINT FOREIGN key(id\_circuit) REFERENCES circuits(id\_circuit);

create table `clients`(

id\_client int not null primary key AUTO\_INCREMENT,

nom\_client varchar(30) not null,

prenom\_client varchar(30) not null,

tel\_client varchar(10),

adresse\_client varchar(50),

cp\_client varchar(5)

)engine=INNODB;

create table `inscriptions`(

id\_client int not null,

id\_voyage int not null,

primary key(id\_client, id\_voyage),

date\_inscription date DEFAULT CURRENT\_TIMESTAMP() not null

)engine=INNODB;

alter table inscriptions

add CONSTRAINT FOREIGN key(id\_client) REFERENCES clients(id\_client);

alter table inscriptions

add CONSTRAINT FOREIGN key(id\_voyage) REFERENCES voyages(id\_voyage);

insert into clients (nom\_client,prenom\_client,tel\_client,adresse\_client,cp\_client) values

("Rigal", "Mylène", "0618564413", "3 rue de la colombette", "31000"),

("Grande", "Maria", "0629564413", "avenue de strasbourg", "65000"),

("Audran", "Corinne", "0745964413", "11 rue de la gare", "81000"),

("Sauviac", "Bernard", "0699464413", "5 rue voltaire", "65310");

insert into voyages (date\_voyage,prix\_voyage,nb\_places\_voyage,id\_circuit) values

("2023-01-02", 1200, 250, 1),

("2023-01-05", 1000, 200, 2),

("2023-02-10", 980, 320, 3),

("2023-01-08", 2500, 400, 4);

insert into circuits (destination\_circuit,duree\_circuit) values

("Martinique", 10),

("New-York", 9),

("Milan", 1),

("Amsterdam", 2);

insert into inscriptions (id\_client,id\_voyage,date\_inscription) values

(1, 2, "2023-01-12"),

(2, 1, "2023-01-20"),

(3, 3, "2023-02-21");

4)

select clients.id\_client, clients.nom\_client, voyages.id\_voyage, inscriptions.date\_inscription, voyages.date\_voyage, voyages.prix\_voyage, circuits.destination\_circuit, circuits.duree\_circuit

from clients

inner join inscriptions on clients.id\_client=inscriptions.id\_client

inner join voyages on inscriptions.id\_voyage=voyages.id\_voyage

inner join circuits on voyages.id\_circuit=circuits.id\_circuit;

5)

select voyages.id\_voyage, voyages.nb\_places\_voyage - count(inscriptions.id\_client) as "Nombre de places restantes"

from inscriptions

inner join voyages on inscriptions.id\_voyage=voyages.id\_voyage

group by voyages.id\_voyage;

6)

select clients.id\_client, clients.nom\_client, sum(voyages.prix\_voyage) as "Total payé"

from clients

INNER join inscriptions on clients.id\_client=inscriptions.id\_client

inner join voyages on inscriptions.id\_voyage=voyages.id\_voyage

group by clients.id\_client;

7)

update voyages

set prix\_voyage=prix\_voyage - prix\_voyage\*0.1, nb\_places\_voyage=nb\_places\_voyage+nb\_places\_voyage\*0.2

where voyages.id\_voyage = (select voyages.id\_voyage

from voyages

inner join inscriptions on voyages.id\_voyage=inscriptions.id\_client

group by voyages.id\_voyage

having count(inscriptions.id\_client) = voyages.nb\_places\_voyage);

8)

delete from clients

where clients.id\_client in

(

select id\_client

from inscriptions

where datediff(curdate(), inscriptions.date\_inscription)>365\*3

);