Membre d’équipe 1 : Hakim, Hamidreza, Amr, Oleksendr

**--1.Le script de création de tables**

drop table salle cascade constraints purge;

drop table typechirurgie cascade constraints purge;

drop table specialite cascade constraints purge;

drop table categories cascade constraints purge;

drop table specialisationsalle cascade constraints purge;

drop table medicament cascade constraints purge;

drop table ordonnancemedicaments cascade constraints purge;

drop table ordonnance cascade constraints purge;

drop table docteur cascade constraints purge;

drop table dossierpatient cascade constraints purge;

drop table consultation cascade constraints purge;

drop table chirurgie cascade constraints purge;

drop table ordonnancechirugie cascade constraints purge;

create table ordonnance(

numord integer,

recommandations varchar2(50),

typeo varchar2(20),

datec date default to\_char(sysdate,'yyyy-MM-dd') ,

constraint ord\_numord\_pk primary key (numord) ,

constraint ord\_typo\_ck check(typeo in ('Chirurgie','Medicaments'))

);

create table salle(

idSalle integer,

nom varchar2(20),

constraint salle\_id\_pk primary key(idSalle)

);

create table typechirurgie(

idtype integer,

nom varchar2(20),

description varchar2(50),

constraint typchir\_id\_pk primary key (idtype)

);

create table specialite(

code integer,

titre varchar2(20) not null,

description varchar2(50),

constraint spec\_code\_pk primary key(code)

);

create table categories(

idCategorie integer,

nom varchar2(20),

description varchar2(20),

constraint categ\_idCategorie\_pk primary key(idCategorie)

);

create table specialisationsalle(

idtype integer,

idsalle integer,

datec date,

constraint spec\_idtype\_fk foreign key (idtype) references typechirurgie(idtype),

constraint spec\_idsalle\_fk foreign key (idsalle) references salle(idsalle),

constraint specsalle\_pk primary key (idtype, idsalle)

);

create table medicament(

idmed integer,

nomMed varchar2(20) not null,

prix number (6,2) default 0,

categorie integer,

constraint med\_idmed\_pk primary key (idmed),

constraint prix\_check check(prix>=0),

constraint med\_nom\_cat\_u unique(nomMed,categorie),

constraint medic\_categorie\_fk foreign key(categorie) references categories(idCategorie)

);

create table docteur(

matricule integer,

nomM varchar2(20) not null,

prenomM varchar2(20) not null,

specialite integer ,

ville varchar2(10),

adresse varchar2(30),

niveau varchar2(20),

nbrpatients integer default 0,

constraint doc\_matricule\_pk primary key (matricule),

constraint doc\_specialite\_fk foreign key (specialite)references specialite(code),

constraint doc\_niveau\_ck check(niveau in('Etudiant','Interne','Docteur'))

);

create table dossierpatient(

numdos integer ,

nomP varchar2(20) not null,

prenomP varchar2(20) not null,

genre varchar2(1),

numas varchar2(20) ,

datenaiss date,

datec date default sysdate,

matricule integer,

constraint dp\_numdos\_pk primary key (numdos),

constraint dp\_matricule\_fk foreign key (matricule)references docteur(matricule),

constraint dp\_genre\_chk check(genre in('M','F')),

constraint dp\_numas\_unique unique(numas)

);

create table consultation(

codedocteur integer,

numdos integer,

datec date default sysdate,

diagnostic varchar2(50) not null,

numord integer,

constraint cons\_co\_num\_date\_pk primary key(codedocteur,numdos,datec),

constraint cons\_codedocteur\_fk foreign key (codedocteur)references docteur(matricule),

constraint cons\_numdos\_fk foreign key (numdos) references dossierpatient(numdos) ,

constraint cons\_numord\_fk foreign key (numord) references ordonnance(numord)

);

create table chirurgie (

idchir integer,

idtype integer,

idsalle integer,

datechirurgie date default sysdate,

heuredebut integer ,

heurefin integer,

constraint chirurgie\_idchir\_pk primary key(idchir),

constraint chirurgie\_idtype\_fk foreign key (idtype)references typechirurgie(idtype),

constraint chirurgie\_idsalle\_fk foreign key (idsalle)references salle(idsalle)

);

create table ordonnancechirugie(

numord integer,

idchir integer,

rang integer,

constraint oc\_numord\_idchir\_pk primary key(numord,idchir),

constraint oc\_numord\_fk foreign key (numord)references ordonnance(numord),

constraint oc\_num\_rang\_u unique(numord,rang),

constraint oc\_idchir\_fk foreign key (idchir)references chirurgie(idchir)

);

create table ordonnancemedicaments(

numord integer,

idmed integer,

nbboites integer default 0,

constraint ordM\_ordmed\_pk primary key (numord,idmed),

constraint ordM\_numord\_fk foreign key (numord) references ordonnance(NUMORD),

constraint ordM\_idmed\_fk foreign key (idmed) references medicament(idmed),

constraint ordM\_nbboites\_check check(nbboites>=0)

);

**--2.Le script d’insertion de donnes**

insert into ordonnance values(1,'ne pas manger trop sale','Chirurgie',default);

insert into ordonnance values(2,'ne pas manger trop sucre','Chirurgie',to\_date('2018-05-09','yyyy-mm-dd'));

insert into ordonnance values(3,'dormir sffisament','Chirurgie',to\_date('2018-12-03','yyyy-mm-dd'));

insert into ordonnance values(4,'30 min de marche par jour','Medicaments',to\_date('2018-11-10','yyyy-mm-dd'));

insert into ordonnance values(5,'pas de viande','Medicaments','2017-11-10');

insert into SALLE values(100,'Operations');

insert into salle values(101,'Consultations');

insert into salle values(102,'Frants Fanon');

insert into salle values(103,'Pasteur');

insert into salle values(104,'Avesens');

insert into TYPECHIRURGIE values(10,'cardio','c''est pour le ceour ' );

insert into TYPECHIRURGIE values(11,'esthetique','c''est pour retrecir le nez ' );

insert into TYPECHIRURGIE values(12,'dentaire','c''est pour enlever une dent ' );

insert into TYPECHIRURGIE values(13,'bariatrique','c''est pour le gastro vertical' );

insert into SPECIALITE values(1000,'cariologie','maladie de cas cariologie');

insert into SPECIALITE values(1001,'endocrinologie','maladie de cas endocrinologie ');

insert into SPECIALITE values(1002,'dermatologie','maladie de cas dermatologie ');

insert into SPECIALITE values(1003,'andrologie','maladie de cas andrologie ');

insert into CATEGORIES values(200,'Analgesique','pour douleur');

insert into CATEGORIES values(201,'anti-inflammatoires','pour inflammation');

insert into CATEGORIES values(202,'antibiotique','pour mal a tete');

insert into CATEGORIES values(203,'antibacterie','pour bacterie ');

insert into SPECIALISATIONSALLE values(10,100,'2017-11-10');

insert into SPECIALISATIONSALLE values(11,101,'2017-12-10');

insert into SPECIALISATIONSALLE values(12,102,'2017-01-11');

insert into SPECIALISATIONSALLE values(13,103,'2018-11-10');

insert into medicament values (001, 'asperin', 150, 200);

insert into medicament values (002, 'coldstop', 176.43, 201);

insert into medicament values (003, 'tylenol', 140.82, 202);

insert into medicament values (004, 'amoxycilin', 217.89, 203);

insert into docteur values (0011, 'Oleksendr', 'Brown', 1000, 'Montreal', '2000 ave decarie', 'Etudiant', 12);

insert into docteur values (0012, 'Hami', 'Kave', 1001, 'Montreal', '2000 ave decarie', 'Docteur', 23);

insert into docteur values (0013, 'Amr', 'Soudy', 1002, 'Toronto', '2000 ave decarie', 'Interne', 45);

insert into docteur values (0014, 'Hakim', 'Mamar', 1003, 'Montreal', '2000 ave decarie', 'Docteur', 34);

insert into dossierpatient values (701, 'Young', 'Paul', 'M', 1265436,'2006-05-14', default, 0011);

insert into dossierpatient values (702, 'Marx', 'Richard', 'M', 1267637, '2012-05-14', default, 0012);

insert into dossierpatient values (703, 'Cary', 'Mary', 'F', 1534237, '2017-05-14', default, 0013);

insert into dossierpatient values (704, 'Denny', 'De Vito', 'M', 1263278, '2006-05-14', default, 0012);

insert into consultation values (0011, 701, default, 'Infection', 1);

insert into consultation values (0012, 702, default, 'Inflammation', 2);

insert into consultation values (0013, 703, default, 'Alcoolisme chronique', 3);

insert into consultation values (0012, 704, default, 'Gastro-entérite', 4);

insert into consultation values (0011, 704, '2006-05-14', 'Gastro', 5);

insert into chirurgie values (101, 10, 100, '2018-04-30', 830, 1100);

insert into chirurgie values (102, 11, 101, '2018-05-02', 1030, 1400);

insert into chirurgie values (103, 12, 100, '2018-06-15', 1000, 1500);

insert into chirurgie values (104, 13, 102, '2018-07-10', 830, 1100);

insert into ORDONNANCECHIRUGIE values (1, 101, 3500);

insert into ORDONNANCECHIRUGIE values (2, 102, 3600);

insert into ORDONNANCECHIRUGIE values (2, 103, 4500);

insert into ORDONNANCECHIRUGIE values (3, 104, 4888);

insert into ORDONNANCEMEDICAMENTS values (1, 001, default);

insert into ORDONNANCEMEDICAMENTS values (2,002, 2);

insert into ORDONNANCEMEDICAMENTS values (2,003, 34);

insert into ORDONNANCEMEDICAMENTS values (3,004, 56);

**--3. script de requête**

**--affiche le nombrede consulatation par docteurs**

select d.matricule, d.nomm, d.prenomm, count(c.codedocteur)

from docteur d, consultation c

where d.MATRICULE = c.CODEDOCTEUR

group by d.matricule, d.nomm, d.prenomm;

**--afficher le nombre de consultation par anne**

select to\_char(datec,'YYYY'), count(codedocteur)

from consultation

group by to\_char(datec,'YYYY');

--**afficher le nombre de chirurgies par docteur**

select matricule ,count(idchir) "number de chirurgie"

from CHIRURGIE join ORDONNANCECHIRUGIE using(idchir)

join CONSULTATION using (numord)

join docteur on (codedocteur=matricule)

group by matricule;

**--afficher le nombre de medicaments par decteur**

select MATRICULE ,count(idmed) from MEDICAMENT

join ORDONNANCEMEDICAMENTS using (idmed)

join CONSULTATION using (NUMORD)

join docteur on (CODEDOCTEUR=MATRICULE) group by MATRICULE;

**--afficher le nombre de medicament par docteur et par anne**

select MATRICULE ,to\_char(datec,'yyyy') Anne,count(idmed) NoMedicament from MEDICAMENT

join ORDONNANCEMEDICAMENTS using (idmed)

join CONSULTATION using (NUMORD)

join docteur on (CODEDOCTEUR=MATRICULE)

group by MATRICULE ,to\_char(datec,'yyyy');

**--4. Code Java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package Control;

import java.sql.\*;

/\*\*

\*

\* @author team1

\*/

public class Aptctr {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws ClassNotFoundException, SQLException {

Connection con = null;

Class.forName("oracle.jdbc.OracleDriver");

con = DriverManager.getConnection("jdbc:oracle:thin:@144.217.163.57:1521:XE", "team8211", "anypw1");

System.out.println("connected");

//les inserts

insertOrdonnance(con, 6, "ne pas manger trop saler", "Chirurgie", Date.valueOf("2018-02-12"));

insertSalle(con, 110, "soldat anonyme");

insertTypeChirurgie(con, 14, "neurochirurgie", " chirurgie des nerf et cerveau");

insertSpecialite(con, 1004, "neuro", " maladie des nerfs");

insertCategories(con, 204, "categorie", "pour maladie");

insertSpecSalle(con, 11, 102, Date.valueOf("2018-08-07"));

insertMedicament(con, 5, "advil", 7.99, 200);

insertDocteur(con, 15, "House", "Robert", 1001, "Laval", "2015 rue delavie ", "Docteur", 110);

insertDossierPat(con, 710, "Malade", "Grave", "F", "MG1111111", Date.valueOf("1990-02-28"),

Date.valueOf("2018-05-09"), 15);

insertConsultation(con, 15, 710, Date.valueOf("2016-03-31"), "Alergie", 5);

insertChirurgie(con, 105, 10, 104, Date.valueOf("2015-03-29"), 830, 1000);

insertOrdChir(con, 4, 103, 4000);

insererOrdonnanceMedicamants(con, 3, 2, 67);

//les script

nbChirurgiesParDocteur(con);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

afficherNbConsultationParAnne(con);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

afficherNbConsulatationParDocteurs(con);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

afficherNbMedicamentsParDecteur(con);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

afficherNbMedicamentParDocteurParAnne(con);

con.close();

}

private static void nbChirurgiesParDocteur(Connection con) throws SQLException {

String sql = "select matricule ,count(idchir)\n"

+ "from CHIRURGIE join ORDONNANCECHIRUGIE using(idchir) \n"

+ "join CONSULTATION using (numord)\n"

+ "join docteur on (codedocteur=matricule)\n"

+ "group by matricule";

Statement stm = con.createStatement();

ResultSet rs = stm.executeQuery(sql);

while (rs.next()) {

System.out.println("Matricule: " + rs.getInt(1) + " Nombre de chirurgie: " + rs.getInt(2));

}

stm.close();

rs.close();

}

private static void afficherNbConsultationParAnne(Connection con) throws SQLException {

String sql = "select to\_char(datec,'yyyy') MoisAnne,count(\*)\n"

+ "from consultation\n"

+ "group by to\_char(datec,'yyyy')";

Statement stm = con.createStatement();

ResultSet rs = stm.executeQuery(sql);

while (rs.next()) {

System.out.println("Anne: " + rs.getInt(1) + " Nombre de cosultation: " + rs.getInt(2));

}

stm.close();

rs.close();

}

private static void afficherNbConsulatationParDocteurs(Connection con) throws SQLException {

String sql = "select d.matricule, d.nomm, d.prenomm, count(c.codedocteur)\n"

+ "from docteur d, consultation c\n"

+ "where d.MATRICULE = c.CODEDOCTEUR\n"

+ "group by d.matricule, d.nomm, d.prenomm";

Statement stm = con.createStatement();

ResultSet rs = stm.executeQuery(sql);

while (rs.next()) {

System.out.println("Matricule: " + rs.getInt(1) + " Nom: " + rs.getString(2)

+ " Prenom: " + rs.getString(3) + " Nombre de consultation: " + rs.getInt(4));

}

stm.close();

rs.close();

}

private static void afficherNbMedicamentsParDecteur(Connection con) throws SQLException {

String sql = "select MATRICULE ,count(idmed) from MEDICAMENT \n"

+ "join ORDONNANCEMEDICAMENTS using (idmed)\n"

+ "join CONSULTATION using (NUMORD)\n"

+ "join docteur on (CODEDOCTEUR=MATRICULE) group by MATRICULE";

Statement stm = con.createStatement();

ResultSet rs = stm.executeQuery(sql);

while (rs.next()) {

System.out.println("Matricule: " + rs.getInt(1) + " Nomber de Medicament : " + rs.getInt(2));

}

stm.close();

rs.close();

}

private static void afficherNbMedicamentParDocteurParAnne(Connection con) throws SQLException {

String sql = "select MATRICULE ,to\_char(datec,'yyyy'),count(idmed) from MEDICAMENT \n"

+ "join ORDONNANCEMEDICAMENTS using (idmed)\n"

+ "join CONSULTATION using (NUMORD)\n"

+ "join docteur on (CODEDOCTEUR=MATRICULE) \n"

+ "group by MATRICULE ,to\_char(datec,'yyyy')";

Statement stm = con.createStatement();

ResultSet rs = stm.executeQuery(sql);

while (rs.next()) {

System.out.println("Matricule: " + rs.getInt(1) + " Anne : " + rs.getInt(2) + " Nomber de Matricule : " + rs.getInt(3));

}

stm.close();

rs.close();

}

public static int insertOrdonnance(Connection con, int num, String recommandations, String typ, Date datec) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into ordonnance values(?,?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, num);

stm.setString(2, recommandations);

stm.setString(3, typ);

stm.setDate(4, datec);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertSalle(Connection con, int no, String nom) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into salle values(?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, no);

stm.setString(2, nom);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertTypeChirurgie(Connection con, int no, String nom, String descr) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into TYPECHIRURGIE values(?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, no);

stm.setString(2, nom);

stm.setString(3, descr);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertSpecialite(Connection con, int code, String titre, String descr) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into SPECIALITE values(?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, code);

stm.setString(2, titre);

stm.setString(3, descr);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertCategories(Connection con, int num, String nom, String descr) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into CATEGORIES values(?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, num);

stm.setString(2, nom);

stm.setString(3, descr);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertSpecSalle(Connection con, int idtype, int idsalle, Date datec) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into SPECIALISATIONSALLE values(?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, idtype);

stm.setInt(2, idsalle);

stm.setDate(3, datec);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertMedicament(Connection con, int idmed, String nommed, double prix, int idcat) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into MEDICAMENT values(?,?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, idmed);

stm.setString(2, nommed);

stm.setDouble(3, prix);

stm.setInt(4, idcat);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertDocteur(Connection con, int matricule, String nom, String prenom,

int spec, String ville, String adresse, String niveau, int nbPatient) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into Docteur values(?,?,?,?,?,?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, matricule);

stm.setString(2, nom);

stm.setString(3, prenom);

stm.setInt(4, spec);

stm.setString(5, ville);

stm.setString(6, adresse);

stm.setString(7, niveau);

stm.setInt(8, nbPatient);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertDossierPat(Connection con, int nomDos, String nom, String prenom,

String genre, String numeroAs, Date dateNais, Date dateC, int matricule) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into DOSSIERPATIENT values(?,?,?,?,?,?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, nomDos);

stm.setString(2, nom);

stm.setString(3, prenom);

stm.setString(4, genre);

stm.setString(5, numeroAs);

stm.setDate(6, dateNais);

stm.setDate(7, dateC);

stm.setInt(8, matricule);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertConsultation(Connection con, int codeDoc, int numDos,

Date dateC, String diagnos, int numOrd) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into CONSULTATION values(?,?,?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, codeDoc);

stm.setInt(2, numDos);

stm.setDate(3, dateC);

stm.setString(4, diagnos);

stm.setInt(5, numOrd);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertChirurgie(Connection con, int idChir, int idType,

int idSalle, Date dateChir, int heureDebut, int heureFin) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into CHIRURGIE values(?,?,?,?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, idChir);

stm.setInt(2, idType);

stm.setInt(3, idSalle);

stm.setDate(4, dateChir);

stm.setInt(5, heureDebut);

stm.setInt(6, heureFin);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

public static int insertOrdChir(Connection con, int numOrd, int idChir, int rang) throws SQLException {

PreparedStatement stm;

String sql1 = "insert into ORDONNANCECHIRUGIE values(?,?,?)";

stm = con.prepareStatement(sql1);

stm.setInt(1, numOrd);

stm.setInt(2, idChir);

stm.setInt(3, rang);

int rs = stm.executeUpdate();

stm.close();

return rs;

}

private static int insererOrdonnanceMedicamants(Connection con, int numord, int idmed, int nbboites) throws SQLException {

String sql = "insert into ORDONNANCEMEDICAMENTS values (?, ?, ?)";

PreparedStatement pstm = con.prepareStatement(sql);

pstm.setInt(1, numord);

pstm.setInt(2, idmed);

pstm.setInt(3, nbboites);

int rs = pstm.executeUpdate();

pstm.close();

return rs;

}

}