

Licence Informatique - Sorbonne Université

LU2IN009 - Bases de Données

Support de TD 2023

Table des matières

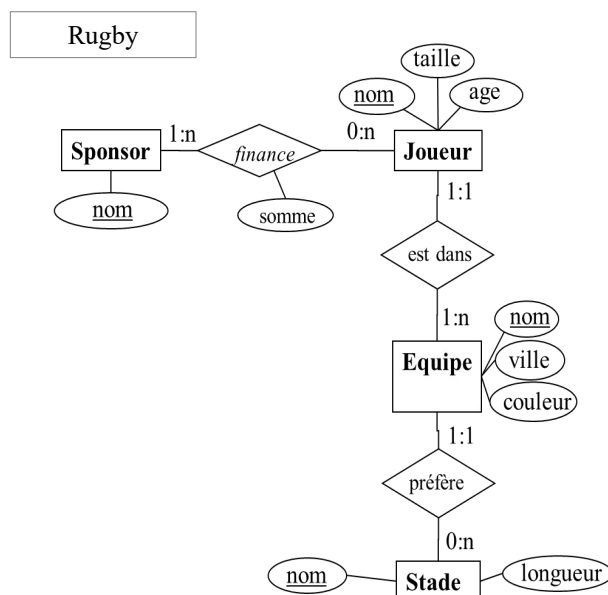
1	TD 1 et 2 - Modélisation Entité-Association	2
2	TD 3 - Calcul Relationnel	8
3	TD 4 - SQL Introduction	12
4	TD 5 - SQL Jointures	15
5	TD 6 - Requêtes imbriquées avec EXISTS, ALL et ANY	17
6	TD 7 - Requêtes d'agrégation et division	19
7	TD 8 - Création de schémas, contraintes d'intégrité	21
8	TD 9 - Création de schémas, modification de données	23
9	TD 10 - PL/SQL	26
10	TD 11 - Triggers	28

masquer:1

TD 1 ET 2 : MODÈLE ENTITÉS-ASSOCIATIONS – MODÈLE RELATIONNEL

1. Equipe de Rugby

Soit le schéma entité association ci-dessous.



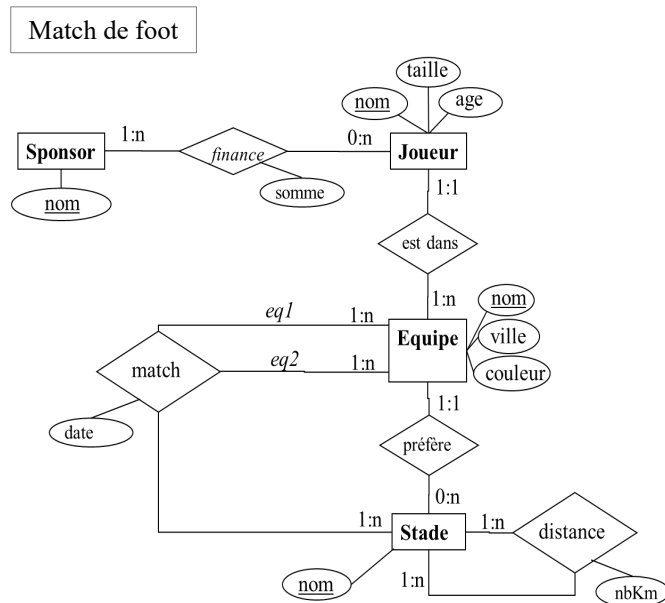
1) Questions simples de lecture :

- Combien y a-t-il d'entités ? Quel est l'identifiant d'une équipe ?
- Peut-on avoir deux stades nommés 'Stade municipal' ?
- Combien d'associations ont au moins un attribut ? L'attribut de l'association peut-il être ajouté comme attribut des entités reliées par cette association ?
- Pourquoi l'entité Equipe ne doit pas avoir un attribut « stade préféré » dont la valeur serait le nom du stade ?
- Est-ce qu'un Sponsor peut ne financer aucun Joueur ?
- Est-ce qu'un Joueur peut n'avoir aucun Sponsor ?
- A combien d'équipes (au moins et au plus) un joueur peut-il appartenir ?
- Est-ce que plusieurs équipes peuvent avoir le même stade préféré ?
- Est-ce qu'un stade est toujours préféré par au moins une équipe ?

2) Modifiez le schéma pour tenir compte des descriptions suivantes :

- On connaît la durée pendant laquelle un sponsor finance un joueur (en nombre d'années)
- On a 2 équipes nommées Racing dans les villes de Pau et à Aix.
- On connaît également la population de la ville.
- Une équipe peut avoir un stade d'entraînement qui n'est pas nécessairement celui qu'elle préfère. Certaines équipes n'ont pas de stade d'entraînement.

2. Match de foot



1) Questions simples de lecture :

- Combien existe-t-il d'associations réflexives ?
- Quelle association a la plus grande arité? Quelle est son arité ?
- Pourquoi n'y a-t-il pas de label sur les arcs de l'association *distance* ?

2) On veut adapter le schéma pour tenir compte des descriptions suivantes :

- Pour un match on connaît l'équipe qui invite et celle qui est invitée
- Une équipe peut-elle jouer contre elle-même?
- On connaît par la suite le numéro d'adhérent national d'une personne et on suppose également qu'il y a des joueurs avec le même nom.
- Une équipe a un entraîneur. On connaît le nom, l'âge et le n° d'adhérent national de l'entraîneur d'une équipe, mais pas sa taille. Un entraîneur peut entraîner plusieurs équipes, une équipe a un seul entraîneur.

3. Musée

Représenter le schéma E/A de l'énoncé suivant. Une ville (nom, pays) a des musées (nom, description). Une œuvre (titre, siècle) est exposée dans un musée pendant une certaine période (début, fin). Une œuvre peut ne pas être exposée. Elle peut aussi être exposée dans différents musées à différentes périodes. On connaît le nom et le prénom de l'artiste qui a réalisé une œuvre. Il y a un artiste par œuvre, les artistes réalisent de nombreuses œuvres.

4. Album de musique

- Représenter le schéma E/A de l'énoncé suivant. Un album (code, date), identifié par son code, est composé d'une série de plages. Les plages d'un album sont numérotées 1, 2, ... ; elles ont une durée. Un album contient au moins une plage. Chaque plage est l'enregistrement d'une seule œuvre, mais une œuvre peut s'étendre sur plusieurs plages (par exemple une symphonie en 4 mouvements). Une œuvre a un identifiant et un titre. Certaines œuvres ne sont pas enregistrées. On connaît les interprètes de l'œuvre pour une plage donnée. Un interprète a un identifiant et un nom et peut jouer

de nombreuses œuvres. Une œuvre peut être jouée par plusieurs interprètes.

- b) On suppose que chaque interprète utilise exactement un instrument (piano, guitare, etc.) sur une plage. Où placer l'attribut « instrument » dans le schéma précédent ?

5. Course Nautique

On considère une course nautique qui se déroule en plusieurs épreuves sanctionnées chacune par un classement des bateaux participants, chaque bateau participant à l'épreuve a toujours un classement même s'il n'arrive pas au bout de l'épreuve (dernière position). Il y a une seule épreuve par jour, chaque épreuve débute et se termine dans un port, le port d'arrivée pouvant être différent du port de départ. Chaque bateau a un numéro d'immatriculation, un nom et une longueur. Il a un skipper et un équipage et est financé par un ou plusieurs sponsors. Le skipper d'un bateau ne peut pas changer d'une épreuve à l'autre de la course, en revanche la composition des équipiers d'un bateau peut changer d'une épreuve à l'autre de la course.

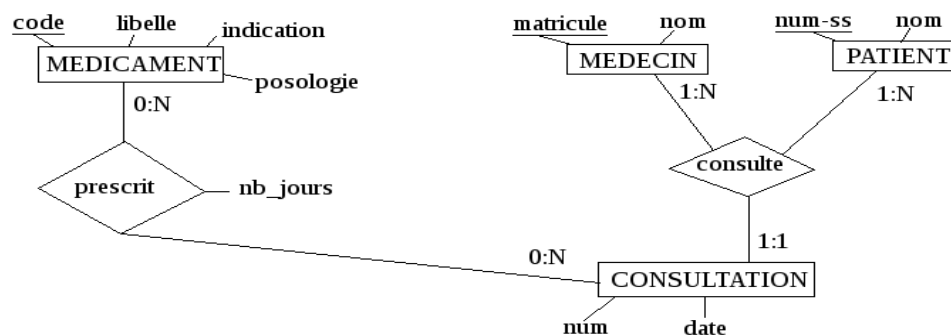
La base de données doit permettre de répondre, entre autres, aux questions suivantes :

- Quels sont les sponsors d'un bateau ?
- Quel est le montant de la subvention d'un sponsor particulier à un bateau particulier ?
- Quels bateaux sont engagés dans l'épreuve qui débute le 27 avril ?
- Quels sont les équipiers du bateau qui a gagné la première épreuve ?
- Sur quels bateaux de plus de 12 mètres un équipier médecin est-il engagé ?

- 1) Proposez un schéma E/A qui modélise la course nautique.
- 2) (*facultatif*) Dans votre schéma E/A, est-ce qu'un équipier peut être engagé sur plusieurs bateaux pendant la même épreuve ? Si oui, modifier le schéma E/A afin de préciser qu'un équipier ne peut pas changer de bateau pendant une épreuve.
- 3) (Séance 2) Déduisez le schéma relationnel de la base

6. Consultation médicale

Soit le schéma Entités/Associations (E/A) suivant, représentant les visites dans un centre médical.



1. Identifiez les différentes composantes d'un schéma E/A dans le schéma ci-dessus.
2. Répondez aux questions suivantes en fonction des caractéristiques de ce schéma.
 - a) Peut-on prescrire plusieurs médicaments dans une même consultation ?
 - b) Un médecin peut-il recevoir plusieurs patients dans la même consultation ?
 - c) Un patient peut-il être consulté plusieurs fois ?
 - d) Un médicament peut-il être prescrit plusieurs fois pour un même patient.
 - e) Un patient peut-il être consulté plusieurs fois le même jour ?
 - f) Un patient peut-il être consulté plusieurs fois le même jour par le même médecin ?
3. (Séance 2) Déduisez un schéma relationnel du schéma conceptuel E/A ci-dessus.
4. (*facultatif*) Donner un exemple de base de données qui suit le schéma relationnel précédent et qui

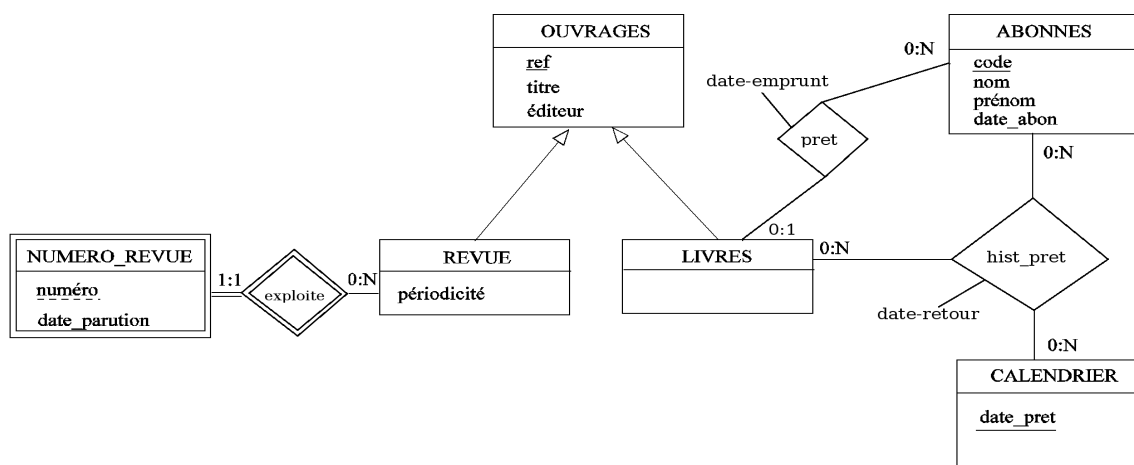
illustre les questions 2.a, 2.b et 2.c.

7. Cinéma

Une base de données relative à des films et des acteurs contient des informations sur des films et des acteurs. Les films sont identifiés par leur titre, ils sont décrits par leur réalisateur (nom et prénom) et par le nombre d'entrées. Les acteurs sont identifiés par leur nom et par leur prénom et décrits par leur âge et leur adresse. Les cachets reçus par chaque acteur et chaque réalisateur dans chaque film dans lequel il/elle a joué.

1. Proposer un schéma E/A pour cette base de données. On explicitera les hypothèses retenues et on justifiera la réponse.
2. (Séance 2) Déduisez-un schéma relationnel du schéma conceptuel E/A amélioré proposé. Quelles sont les clés primaires et les clés étrangères ?
3. (*facultatif*) Donner un exemple de base de données qui suit le schéma relationnel précédent avec au moins un acteur, un film où l'acteur a joué sans le réaliser, un film que l'acteur a réalisé sans y jouer, un film que l'acteur a réalisé et dans lequel il a joué et un film sans acteur.
4. Qu'est-ce que cela change pour le modèle conceptuel et pour le schéma relationnel si on considère dorénavant qu'un film peut avoir plusieurs réalisateurs ? *facultatif*: Donner un exemple de la nouvelle table avec le film 'Kung Fu Panda' réalisé par Mark Osborne et John Stevenson.

8. Bibliothèque



1. Déduire un schéma relationnel du schéma E/A en supposant que la bibliothèque ne contient que des revues et des livres
2. Comment modifier le schéma relationnel si la bibliothèque peut prêter des ouvrages qui ne sont ni des revues, ni des livres ?

9. Personnels Universitaires

On désire gérer les emplois du temps des différents personnels (*enseignants, enseignants-chercheurs et chercheurs*) de l'UFR, pour savoir à tout moment s'il est possible de les joindre, et où. Pour cela, on considère que, en dehors des périodes où ils peuvent être joints dans leur bureau, les personnels peuvent être en réunion, quel que soit leur statut. Une réunion est désignée par une date précise, une tranche horaire (début, fin) et une salle de réunion. On veut aussi connaître les autres personnes participant à la réunion.

Chaque personne est désignée par son nom, son prénom, le bureau où on peut la joindre.

Les *enseignants* peuvent, de plus, être en cours. Un cours est identifié par la matière enseignée à laquelle est affectée toujours la même salle de cours. Il est désigné par une période de début et de fin (ex. de février à mai), un jour de la semaine, une tranche horaire. Plusieurs enseignants peuvent enseigner la même matière dans l'année, à des jours et créneaux horaire différents. Un enseignant peut enseigner plusieurs matières et plusieurs fois la même matière dans l'année, à des périodes différentes.

Les *chercheurs* peuvent être à certaines périodes de l'année en mission en dehors de l'UFR. Une mission est désignée par une date de début et de fin et un lieu de mission. Les chercheurs appartiennent à un laboratoire dont on peut joindre le secrétariat en cas d'urgence.

Les *enseignants-chercheurs* sont à la fois enseignants et chercheurs, avec un pourcentage plus ou moins grand d'enseignement (par rapport à la recherche) à effectuer. Ils peuvent donc être soit en réunion, soit en mission, soit en cours.

Établir le schéma Entité-Association de cette application. En déduire un schéma de base de données relationnelle. Ne pas oublier de préciser les cardinalités des associations et les identificateurs des entités.

10. Fédération de cyclisme

La fédération internationale de cyclisme désire mettre au point une base de données. Celle-ci comporte des informations sur les différents coureurs, les équipes, les résultats obtenus aux différentes courses organisées ainsi que, pour des raisons d'actualité, sur le suivi médical des coureurs. Les coureurs sont identifiés par leur nom et leur prénom, on connaît leur taille, leur date de naissance et l'équipe à laquelle ils appartiennent. Une équipe est identifiée par son nom, elle possède un budget, un directeur sportif dont on connaît le nom, le prénom et la date de naissance. Elle est financée par des sponsors qui peuvent varier selon les années et dont on connaît le nom, l'adresse et le domaine d'activité.

Une course correspond à un nom de course (ex. «Tour de France»), on en connaît la distance totale à parcourir. Elle peut comporter une ou plusieurs étapes, dont on connaît le numéro d'ordre (ex. «3^e étape»), la date, le type (ex. «Contre la montre individuel»), la ville de départ et celle d'arrivée. Pour chaque coureur ayant participé à une étape d'une course, on connaît le classement qu'il a obtenu lors de cette étape. Pour chaque course, on connaît le vainqueur final et l'équipe à laquelle il appartient.

Pour chaque course, les équipes emploient des soigneurs, dont on connaît le nom, le prénom, la date de naissance et la nationalité. On note aussi, à chaque étape, quelle dose de quel(s) produit(s) a administré un soigneur à un coureur. Un produit est identifié par un numéro de produit, a un nom, une indication (ex. «douleur musculaire»), une contre-indication (ex. « ne pas administrer en dessous de 20 ans ») et une posologie (ex. «1 comprimé par jour»).

Dans cette base de donnée de production, seules les informations courantes (concernant l'édition en cours) de la course, des coureurs, des équipes, etc. sont stockées.

- Faire le schéma entité-association correspondant aux besoins de la fédération internationale de cyclisme. Ne pas oublier les cardinalités et les identificateurs.
- (*facultatif*) En déduire le schéma relationnel de la base de données correspondante, sans oublier de préciser les clés de chaque relation.
- Aspect temporel. Chaque année, les informations évoluent. Les résultats bien sûr, mais aussi la composition des équipes, le directeur technique, le budget des équipes, les sponsors, etc. changent chaque année. Comment modifier la base de données précédente pour stocker l'historique des informations selon les différentes éditions ?

11. Tennis : du schéma relationnel vers le schéma E/A

Retrouver le schéma E/A du schéma relationnel suivant (les clés sont soulignées et les attributs de clés étrangères sont signalés avec une * sans le nom de la table référencée):

TOURNOI (lieu, année)

JOUEUR (nujoueur, nom, prénom, annaiss, nationalité)

RENCONTRE (nugagnant*, nuperdant*, lieu*, année*, score)

GAIN (nujoueur*, lieu*, année*, sponsor*, prime,)

SPONSOR (nom_sponsor, chiffre_d_affaires, adresse)

12. JO : relationnel vers E/A

Retrouver le schéma E/A du schéma relationnel suivant

Pays (code, nom)

Ex. ('FRA', 'France')

Sport (sid, nom)

Ex. (1, 'Biathlon')

Epreuve (epId, sid*, nom, categorie, dateDebut, dateFin)

Ex. (10, 1, '15 km individuel', 'Hommes', '13-02-2014', '13-02-2014')

Athlete (aid, nom, prenom, dateNaissance, pays*)

Ex. (100, 'Aaa', 'Bbbb', '05-08-1987', 'FRA')

Equipe (eqId, pays*)

Ex. (1000, 'AUS')

AthletesEquipe (eqId*, aid*)

Ex. (1000, 200)

RangIndividuel (epId*, aid*, rang)

Ex. (10, 100, 23) : L'athlète 100 était 23e dans l'épreuve (individuelle) 10

RangEquipe (epId*, eqId*, rang)

Ex. (13, 1000, 2) : L'équipe 1000 était 2e dans l'épreuve (en équipe) 13

masquer: 1

TD 3: Calcul Relationnel

1. RAPPEL : CALCUL RELATIONNEL À VARIABLE N-UPLET

Dans *calcul relationnel à variable n-uplet* (ou *calcul n-uplet*), une requête est représentée par une expression de la forme $\{ \mathbf{t.A_0}, \mathbf{t.A_1}, \dots \mid \mathcal{F}(\mathbf{t}) \}$, qui désigne une projection sur les attributs A_0, A_1, \dots de l'ensemble des n-uplets \mathbf{t} satisfaisant le prédicat $\mathcal{F}(\mathbf{t})$ de la logique du premier ordre. Dans cette expression, la variable \mathbf{t} représente une n-uplet d'une seule relation ou une concaténation de n-uplets appartenant à des relations différentes (variables non quantifiées ou *libres* dans \mathcal{F}).

Rappel transformation de $\forall x (p \Rightarrow q)$:

$$\forall x (p \Rightarrow q) \equiv \neg (\exists x \neg (p \Rightarrow q)) \equiv \neg (\exists x \neg (\neg p \vee q)) \equiv \neg (\exists x (p \wedge \neg q))$$

2. BASE «FOOFLE»

La société «Foofle» est chargée de l'organisation informatique d'un championnat de football. Parmi les relations de la base de données, on trouve les relations suivantes:

SPONSORISE (NOMSPONSOR, NOMJOUEUR, SOMME)

JOUEUR (NOMJOUEUR, EQUIPE, TAILLE, AGE)

EQUIPE (NOMEQUIPE, VILLE, COULEUR, STADEPRÉFÉRÉ)

MATCH (EQUIPE1, EQUIPE2, DATE, STADE)

DISTANCE (STADE1, STADE2, NBKM)

Le n-uplet SPONSORISE(ROBEEK, KOU FRANCK, 10000) indique que le sponsor ROBEEK a donné 10000 euros au joueur FRANCK KOU pour la coupe du monde. Le n-uplet JOUEUR(KOU FRANCK, DIREKT, 180, 25) indique que le joueur FRANCK KOU fait partie de l'équipe des DIREKT, mesure 1m80 et a 25 ans.

Le n-uplet MATCH(DIREKT, PIÉPLA, 12/05/98, GRANDARENA) indique qu'un match entre l'équipe des DIREKT et celle des PIÉPLA a lieu le 12 mai 1998 au stade GRANDARENA. La relation MATCH étant symétrique, on y trouvera également le n-uplet MATCH(PIÉPLA, DIREKT, 12/05/98, GRANDARENA).

Le n-uplet DISTANCE(GRANDARENA, BOULODROME, 120) indique que la distance entre le stade GRANDARENA et le stade BOULODROME est de 120 km. La relation DISTANCE étant symétrique, on y trouvera également le n-uplet DISTANCE(BOULODROME, GRANDARENA, 120). La relation n'étant pas réflexive, on ne trouve pas, par exemple, un n-uplet DISTANCE(BOULODROME, BOULODROME, 0).

Afin de simplifier l'expression des requêtes, la base est abrégée de la manière suivante:

SPONSORISE (NSP, NJO, SOMME)

JOUEUR (NJO, EQ, TAILLE, AGE)

EQUIPE (NEQ, VILLE, COULEUR, STP)

MATCH (EQ1, EQ2, DATE, ST)

DISTANCE (ST1, ST2, NBKM)

Exprimez les requêtes suivantes en calcul relationnel:

SÉLECTIONS

1. Quels sont les joueurs (nom, prénom) sponsorisés par 'Adadis' ?
2. Quels sont les stades qui se trouvent à une distance comprise entre 100 et 200 km de 'GrandArena' ? Donnez le nom du stade ainsi que sa distance de 'GrandArena' .
3. Quelles sont les équipes (avec tous leurs attributs) de 'Paris' et de 'Lyon' ?

JOINTURES

4. Quelles équipes ont déjà joué au stade préféré de l'équipe des Piépla ?
5. Quels sont les stades où a déjà joué Manon Messi ?

AUTO-JOINTURES

6. Donnez pour chaque ville les couples de différentes équipes de cette ville.
7. Quels sont les joueurs qui ont au moins trois sponsors différents?
8. A quelle date a eu lieu un match entre deux équipes sponsorisées par le même sponsor ? On dit qu'une équipe est sponsorisée par un sponsor si au moins un de ses joueurs est sponsorisé par ce sponsor
9. Quels sont les couples des joueurs différents ayant eu un match le même jour et dans des stades différents mais proches (moins de 50 km) ?

DIFFÉRENCE

10. Quels sont les joueurs qui ne sont pas sponsorisés par 'Adadis' ?
11. Quel est le(s) plus grand(s) joueur(s) sponsorisé(s) par 'Adadis' ?
12. Quelle équipe n'a jamais joué dans son stade préféré ?
13. Quels sont les joueurs qui ont exactement deux sponsors différents?
14. A quelle date a lieu un match opposant le plus grand des joueurs au plus petit ?

15. Quelles sont les équipes rencontrées par l'équipe des Direkt, avec, pour chaque équipe rencontrée, celle que les Direkt ont rencontré ou rencontreront juste après (d'après la date du match). Par exemple, si le nuplet (Piépla, Tètokaré) figure dans le résultat de la requête, cela veut dire que les Tètokaré sont la prochaine équipe rencontrée par les Direkt après les Piépla.
16. Quelles équipes se sont déplacées le plus loin de leur stade préféré pour un match? Pour chaque équipe donner son nom et la longueur du déplacement.

DIVISION

17. Quelle équipe a joué dans tous les stades ?
18. Quelle équipe a joué dans tous les stades autres que son stade préféré ?
19. Quel joueur de plus de 50 ans a été sponsorisé par tous les sponsors ?
20. Quel sponsor a sponsorisé au moins un joueur pour chaque équipe?

2. BASE «TENNIS»

On considère la base TENNIS de schéma :

JOUEUR (NUJOUEUR, NOM, PRENOM, ANNAISS, NATIONALITE)

RENCONTRE (NUGAGNANT, NUPERDANT, LIEUTOURNOI, ANNEE, SCORE)

GAIN (NUJOUEUR, LIEUTOURNOI, ANNEE, PRIME, SPONSOR).

21. Numéro et tournoi d'engagement (défini par le lieu et l'année) des joueurs sponsorisés par Peugeot entre 1990 et 1994 ;
22. Nom et année de naissance des joueurs ayant participé à Roland Garros en 1994 ;
23. Nom et nationalité des joueurs ayant participé à la fois au tournoi de Roland Garros et à celui de Wimbledon, en 1992 ;
24. Nom et nationalité des joueurs ayant été sponsorisés par Peugeot et ayant gagné à Roland Garros au moins un match (avec un sponsor quelconque);
25. Nom et nationalité des joueurs qui n'ont jamais perdu une rencontre à Roland Garros;
26. Nations qui n'ont jamais participé à Roland Garros;
27. Nom des joueurs ayant toutes leur primes à Roland Garros supérieures à 100000;
28. Numéros des joueurs qui ont toujours perdu à Wimbledon et toujours gagné à Roland Garros;
29. Liste des vainqueurs de tournoi, mentionnant le nom du joueur avec le lieu et l'année du tournoi qu'il a gagné;
30. Noms des joueurs qui ont gagné un tournoi ou qui ont gagné plus de 100000 à un tournoi;
31. Noms des joueurs ayant participé à tous les tournois disputés en 1994;
32. Nombre de joueurs ayant participé au tournoi de Wimbledon en 1994.

3. BASE DE DONNÉES « GÉNÉALOGIE »

Soit la base composée des deux relations suivantes :

PERSONNE (nuPersonne, nom, prénom, nuPère, nuMère, sexe)

UNION (nuMari, nuFemme)

dont les clés sont soulignées, et dont les attributs nuPère, nuMère, nuMari et nuFemme sont définis sur le même domaine que nuPersonne.

Exprimer les requêtes suivantes, dans le formalisme du calcul relationnel à variables n-uplet :

33. Prénoms des enfants du Docteur 'John March' ;
34. Nombre de filles du Docteur 'March' ;
35. Liste des couples sans enfants communs;
36. Liste des personnes qui ont eu plusieurs conjoints.
37. Liste des personnes qui n'ont pas de conjoint.

masquer: 1

TD 4 : PREMIERS PAS EN SQL

1. BASE DE DONNÉES « JEUX OLYMPIQUES D'HIVER 2014 »

On considère le schéma de la base JEUXOLYMPIQUE2014 qui décrit les athlètes et leurs résultats aux épreuves des Jeux Olympiques d'Hiver Sotchi 2014 :

ATHLETE (NOM,PRENOM,DATE_{NAISSANCE},PAYS)

RANGÉPREUVE(NOMATH*,PRENOMATH*,ÉPREUVE,SPORT,CATEGORIE,DATE_{DEBUT},DATE_{FIN},RANG)

La relation **ATHLETE** contient des informations concernant les athlètes ayant participé aux Jeux Olympiques. Plus précisément : leur nom, leur prénom, leur date de naissance si elle est renseignée (ou NULL sinon) ainsi que le pays qu'ils représentent.

La relation **RANGÉPREUVE** renvoie pour chaque athlète identifié par NOMATH ET PRENOMATH et chaque épreuve identifiée par SPORT, ÉPREUVE ET CATEGORIE (HOMMES OU FEMMES) à laquelle il a participé, le RANG qu'il a obtenu. On pour chaque épreuve les dates de début et de fin (DATE_{DEBUT}, DATE_{FIN}).

Lorsqu'un athlète participe à une épreuve puis est disqualifié, son rang n'est pas renseigné (NULL).

Exemple de données

ATHLETE :

<u>NOM</u>	<u>PRENOM</u>	DATE _{NAISSANCE}	PAYS
AHN	Victor	'20-10-1985'	'Russie'
ANESI	Matteo	'16-08-1984'	'Italie'
AYMONIER	Celia	NULL	'France'
...

RANGÉPREUVE :

<u>SPORT</u>	<u>ÉPREUVE</u>	CATEGORIE	DATE _{DEBUT}	DATE _{FIN}	<u>NOMATH</u>	<u>PRENOMATH</u>	RANG
'Patinage de vitesse'	'1500m'	'Hommes'	'15-02-14'	'15-02-2014'	ANESI	Matteo	39
'Saut à ski'	'Grand tremplin individuel'	'Hommes'	'14-02-2014'	'15-02-2014'	PREVC	Peter	3
'Surf des neiges'	'Slopestyle'	'Femmes'	'06-02-2014'	'09-02-2014'	ENNE	Merika	NULL
...

Exprimer les requêtes suivantes en SQL :

- 1) Les athlètes français.

Résultat : (92 lignes)

- 2) La date de naissance de Denis TEN.

Résultat: 1993-06-13

- 3) Les épreuves (sport, épreuve) dans lesquelles Therese JOHAUG a gagné une médaille, ainsi que le rang qu'elle a obtenu.

Indice : pour gagner une médaille, il faut avoir été classé dans le rang 1, 2 ou 3.

Résultat : ('Ski de fond', '10km', 3) et ('Ski de fond', '30km', 2)

- 4) La nationalité des athlètes Kamil STOCH et Suk-Hee SHIM.

Résultat : ('SHIM', 'Suk-Hee', 'Corée du Sud') et ('STOCH', 'Kamil', 'Pologne')

- 5) Tous les sports des Jeux Olympiques.

Résultat : (12 lignes)

- 6) Le nom des athlètes originaire de Scandinavie (pays : Danemark, Finlande, Norvège, Suède, Islande) triés par le nom, puis par leur prénom.

Résultat : (186 lignes)

- 7) Le nom de sports contenant la chaîne de caractères 'ski' (par exemple, 'Ski de fond', 'Saut à ski', ...).

Résultat : Ski de fond, Ski alpin, Ski acrobatique, Saut à ski

- 8) Les athlètes (nom, prénom) dont la date de naissance n'est pas renseignée.

Résultat : (1516 lignes)

- 9) Les athlètes qui ont participé à l'épreuve '10km' de 'Ski de fond' sans être disqualifié (rang inconnu), triés par rang.

Résultat : (75 lignes)

- 10) Les épreuves (sport, épreuve et catégorie) qui ont eu lieu après le 21 février 2014.

Résultat : (4 lignes)

- 11) Les épreuves (sport, épreuve et catégorie) qui ont duré entre 2 et 4 jours, ainsi que leur durée exacte.

Résultat : (16 lignes)

- 12) Les épreuves (sport, épreuve et catégorie) dont le final a eu lieu pendant un weekend.

Résultat : (11 lignes)

- 13) Les athlètes (nom, prénom) femmes de nationalité française.

Résultat : (26 lignes)

- 14) Les athlètes (nom, prénom) qui ont aujourd'hui moins de 25 ans et qui ont gagné une médaille.

Résultat : dépend de la date d'exécution : Le 17/12/2021 : 4 lignes pour la première solution, 2 lignes pour la deuxième

- 15) Les athlètes (nom, prénom) qui avaient leur anniversaire à la date de fin de l'épreuve dans laquelle ils participaient.

Résultat : ('GARANICHEV', 'Evgeniy', 13, 2), ('SKOBREV', 'Ivan', 8, 2) et ('WEIBRECHT', 'Andrew', 10, 2)

- 16) Les athlètes français (nom, prénom) qui ont gagné une médaille.

Résultat : (12 lignes)

- 17) Les sports qui n'ont pas d'épreuves féminines.

Résultat : Combiné nordique

RAPPELS : EQUIVALENCE ENTRE LES OPÉRATEURS EXISTS ET IN

Soit le schéma :

COMMANDE(NUM, NOMCLIENT, NOMPRODUIT, QUANTITE)

PRODUIT(NOMPRODUIT, FOURNISSEUR, PRIX)

et les requêtes

R1 : « *Nom, prix et fournisseur des produits commandés par Jean* »

R2: « *Fournisseurs d'un produit ayant un coût supérieur au coût de tous les produits fournis par Pierre*»

1. Exprimer la requête R1 en SQL des trois façons suivantes :

- en utilisant un produit cartésien avec condition de jointure
- en utilisant l'opérateur **exists**
- en utilisant l'opérateur **in** si possible.

2. Écrire la requête R2 en SQL en utilisant l'opérateur **not exists**, **all** et **not in** si possible.

masquer: 1

TD 5 : JOINTURES ET IMBRICATION AVEC IN ET EXISTS

BASE DE DONNÉES « JEUX OLYMPIQUES D'HIVER »

On considère le schéma de la base JEUXOLYMPIQUE2014 qui décrit les athlètes et leurs résultats aux épreuves des sports des Jeux Olympiques d'Hiver Sotchi 2014 :

PAYS (CODEPAYS, NOMP)

Ex. ('FRA', 'France')

SPORT (SID, NOMSp)

Ex. (1, 'Biathlon')

EPREUVE (EPID, SID*, NOMÉp, CATÉGORIE, DATEDEBUT, DATEFIN)

Ex. (10, 1, 'relais 4x7,5km', 'Hommes', 22/02/2014, 22/02/2014)

ATHLETE (AID, NOMATH, PRENOMATH, DATENAISSANCE, CODEPAYS*)

Ex. (1000, 'SOBOLEV', 'Alexey', NULL, 'RUS')

EQUIPE (EQID, CODEPAYS*)

Ex. (30, 'SUI')

ATHLETES EQUIPE (EQID*, AID*)

Ex. (30, 796) : L'athlète (aid=796) a participé à l'équipe (eqid=30)

RANGINDIVIDUEL (EPID*, AID*, RANG)

Ex. (15, 61, 1) : L'athlète (aid=61) a gagné la médaille d'or (rang=1) de l'épreuve (epid=15)

RANGEQUIPE (EPID*, EQID*, RANG)

Ex. (10, 30, 14) : L'équipe (eqid=30) a été classée 14e à l'épreuve (epid=10)

Les attributs qui forment la clé primaire de chaque relation sont soulignés. Les clés étrangères sont signalées avec une *. Les attributs aid, epid, eqid et sid correspondent aux identifiants des athlètes, épreuves, équipes et sports et sont utilisés à la fois comme clé primaire ou comme référence (clé étrangère) vers la relation correspondante.

La relation **PAYS** contient le code et le nom de tous les pays, même si ils n'ont pas participé aux Jeux Olympiques. Les sports (n-uplets de la relation **SPORT**) sont un ensemble d'épreuves (n-uplets de la relation **EPREUVE**). Pour chaque épreuve on connaît son nom et les date de début et fin de l'épreuve. Les épreuves peuvent être individuelles ou par équipe. Dans le premier cas, la participation des athlètes (n-uplets de la relation **ATHLETE**) est stocké dans la table **RANGINDIVIDUEL** qui contient en plus le rang qu'ils ont obtenu (1 pour la médaille d'or). Pour les épreuves par équipe les résultats sont stockés dans la relation **RANGEQUIPE**, alors que l'information sur le pays de chaque équipe et ses participants est stocké dans les relations **EQUIPE** et **ATHLETES EQUIPE**. Dans les relations **RANGINDIVIDUEL** et **RANGEQUIPE** l'attribut rang est égal à null si l'athlète ou l'équipe a été disqualifié.

Exprimer les requêtes suivantes en SQL :

1. Les athlètes français (nom pays = 'France') .

Résultat : (104 lignes)

2. Les épreuves (sport, nom d'épreuve, catégorie) triées par nom de sport, puis par nom d'épreuve dans l'ordre inverse du dictionnaire.

Résultat : (98 lignes)

3. Les athlètes ayant participé à une équipe de France.

Résultat : (30 lignes)

4. Les épreuves (sport, épreuve, catégorie) auxquelles participent des équipes.

Résultat : (25 lignes)

5. Le pays qui a gagné, en équipe, la médaille d'or dans l'épreuve de la catégorie 'Femmes' intitulée 'relais 4x6km' du sport 'Biathlon'.

Résultat : Ukraine

6. Les athlètes ayant participé à au moins une épreuve individuelle et au moins une par équipe. Exprimer la requête de trois façons différentes.

Résultat : (372 lignes)

7. Les homonymes (les nom de familles portés par deux athlètes ou plus).

Résultat : (141 lignes)

8. Les athlètes ayant participé à (au moins) deux épreuves individuelles.

Résultat : (706 lignes)

9. Les athlètes ayant gagné une médaille dans une épreuve individuelle, mais ayant été disqualifiés dans une autre.

Résultat : (14 lignes)

10. Les athlètes ayant gagné (au moins) une médaille dans une épreuve individuelle.

Résultat : (182 lignes)

11. Les noms de pays qui n'ont qu'une seule équipe.

Résultat : (9lignes)

12. Les athlètes ayant participé à exactement 2 épreuves individuelles.

Résultat : (402 lignes)

masquer=1

TD 6 : REQUÊTES IMBRIQUÉES AVEC EXISTS, ALL ET ANY

BASE DE DONNÉES « JEUX OLYMPIQUES D'HIVER »

On considère le schéma de la base JEUXOLYMPIQUE2014 qui décrit les athlètes et leurs résultats aux épreuves des sports des Jeux Olympiques d'Hiver Sotchi 2014 :

PAYS (CODEPAYS, NOMP)

Ex. ('FRA', 'France')

SPORT (SID, NOMSp)

Ex. (1, 'Biathlon')

EPREUVE (EPID, SID*, NOMÉp, CATÉGORIE, DATEDEBUT, DATEFIN)

Ex. (10, 1, 'relais 4x7,5km', 'Hommes', 22/02/2014, 22/02/2014)

ATHLETE (AID, NOMATH, PRENOMATH, DATENAISSANCE, CODEPAYS*)

Ex. (1000, 'SOBOLEV', 'Alexey', NULL, 'RUS')

EQUIPE (EQID, CODEPAYS*)

Ex. (30, 'SUI')

ATHLETES EQUIPE (EQID*, AID*)

Ex. (30, 796) : L'athlète (aid=796) a participé à l'équipe (eqid=30)

RANGINDIVIDUEL (EPID*, AID*, RANG)

Ex. (15, 61, 1) : L'athlète (aid=61) a gagné la médaille d'or (rang=1) de l'épreuve (epid=15)

RANGEQUIPE (EPID*, EQID*, RANG)

Ex. (10, 30, 14) : L'équipe (eqid=30) a été classée 14e à l'épreuve (epid=10)

Les attributs qui forment la clé primaire de chaque relation sont soulignés. Les clés étrangères sont signalées avec une *. Les attributs aid, epid, eqid et sid correspondent aux identifiants des athlètes, épreuves, équipes et sports et sont utilisés à la fois comme clé primaire ou comme référence (clé étrangère) vers la relation correspondante.

La relation **PAYS** contient le code et le nom de tous les pays, même si ils n'ont pas participé aux Jeux Olympiques. Les sports (n-uplets de la relation **SPORT**) sont un ensemble d'épreuves (n-uplets de la relation **EPREUVE**). Pour chaque épreuve on connaît son nom et les date de début et fin de l'épreuve. Les épreuves peuvent être individuelles ou par équipe. Dans le premier cas, la participation des athlètes (n-uplets de la relation **ATHLETE**) est stocké dans la table **RANGINDIVIDUEL** qui contient en plus le rang qu'ils ont obtenu (1 pour la médaille d'or). Pour les épreuves par équipe les résultats sont stockés dans la relation **RANGEQUIPE**, alors que l'information sur le pays de chaque équipe et ses participants et stocké dans les relations **EQUIPE** et **ATHLETES EQUIPE**. Dans les relations **RANGINDIVIDUEL** et **RANGEQUIPE** l'attribut rang est égal à null si l'athlète ou l'équipe a été disqualifié.

Exprimez les requêtes suivantes en SQL :

1. Les athlètes ayant participé à (au moins) une épreuve individuelle et (au moins) une épreuve par équipe. Exprimer la requête en utilisant ANY (au lieu de IN; voir requête 2.6 dans TD5).

Résultat : (372 lignes)

2. a) Les pays ayant au moins un athlète (requête équivalente: les pays ayant participé aux JO).

Résultat : (88 lignes)

- b) Les pays ayant exactement un seul athlète

Résultat : (18 lignes)

3. a) Les athlètes qui n'ont jamais été disqualifiés aux épreuves individuelles.

Résultat : (2194 lignes)

- b) Les pays qui n'ont pas eu d'athlète disqualifié aux épreuves individuelles.

Résultat : (143 lignes)

- c) Les pays qui ont participé aux JO et qui n'ont pas eu d'athlètes disqualifiés aux épreuves individuelles.

Résultat : (25 lignes)

- d) Les pays qui ont participé aux JO et qui n'ont pas eu des athlètes disqualifiés *ni en individuel ni par équipe*

Résultat : (25 lignes)

4. a) Les athlètes n'ayant pas gagné de médaille *ni en individuel ni en équipe*

Résultat : (1921 lignes)

- b) Les pays ayant participé aux JO et n'ayant pas gagné de médaille *ni en individuel ni en équipe*.

Résultat : (63 lignes)

5. L(es) épreuve(s) avec la plus grande durée.

Résultat : Hockey sur glace, Hockey sur glace, Hommes, 16 jours

6. Les athlètes ayant gagné une médaille à toutes les épreuves individuelles auxquelles ils ont participé. *Résultat : (109 lignes)*

BASE « FOOFLE »

On reprend le schéma « Foofle ». Exprimez les requêtes suivantes de plusieurs manières, en utilisant des jointures, ainsi que les opérateurs EXISTS, ANY, ALL:

Sponsorise(NSp, NJo, Somme),

Joueur(NJo, Eq, Taille, Age),

Equipe(NEq, Ville, Couleur, StP)

Match(Eq1, Eq2, DateM, St),

Distance(St1, St2, NbKm)

7. Quelles équipes ont déjà joué au stade préféré de l'équipe des Piepla ?

8. Quels sont les stades où a déjà joué Manon Messi ?

9. A quelle date a eu lieu un match entre deux équipes sponsorisées par le même sponsor ?

10. Quel sponsor a financé deux joueurs différents ayant eu un match le même jour et dans des stades différents mais proches (moins de 50 km) ?

masquer=1

TD 7 : REQUÊTES D'AGRÉGATION ET DIVISION

DONNÉES ASTRONOMIQUES

On veut réaliser une application pour des données astronomiques. Les observations, les dates et les valeurs observées sont inventées pour les besoins de l'exercice. Le rayon des astres est en km. On suppose que les attributs ne peuvent pas être null. On considère le schéma suivant :

Categorie (idC, nom)

Astre (idA, nom, rayon, idC*) avec idC référence **Categorie** (idC)

TourneAutour (idA1*, idA2*, position) avec idA1 et idA2 référence **Astre** (idA)

Observation (idO, idA*, dateObs, valObs) avec idA référence **Astre** (idA)

Exemples de données :

- La Terre est la troisième planète qui tourne autour du Soleil.
- Le 10 mai 2010, on a réalisé deux observations du Soleil avec les valeurs 12001 et 12003.

Categorie		Astre				TourneAutour			Observation			
idC	nom	idA	nom	rayon	idC	idA1	idA2	pos.	idO	idA	dateObs	valObs
10	étoile	100	Soleil	696342	10	101	100	3	90	100	10-05-2010	12001
11	planète	101	Terre	6371	11	102	101	1	91	100	10-05-2010	12003
12	satellite	102	Lune	1737	12	103	100	4	92	101	18-12-2013	8005
		103	Mars	3390	11				93	102	27-08-2014	300

A - Questions préliminaires pour étudier le schéma et les données

1. La première ligne de la table TourneAutour (101,100,3) peut se traduire en français par :
« La Terre est la troisième planète qui tourne autour du Soleil ».
Traduire en français la deuxième ligne de la table TourneAutour : (102,101,1).
2. Pourquoi dans la relation TourneAutour seul l'attribut idA1 est souligné ?
3. Pourrait-on regrouper en une seule relation les relations Astre et TourneAutour ?
4. Peut-on avoir deux observations du même astre le même jour et à la même date ?

B- Fonctions d'agrégation « COUNT, SUM, AVG, MIN, MAX »

Écrivez les expressions SQL pour répondre aux requêtes suivantes.

5. Le nombre de catégories. Résultat : 3
6. Le nombre de catégories pour lesquelles on connaît au moins un astre. Résultat : 3
7. Le rayon minimal et le rayon maximal de tous les astres. Résultat : (1737, 696342)
8. Le rayon moyen en milliers de km (et non pas en km), arrondi à 2 chiffres après la virgule.
Résultat : '176,96 milliers km'. Aide : utilisez :
a) round(valeur,nb) pour garder seulement nb décimales à valeur
b) l'opérateur de concaténation || pour concaténer des chaînes, attributs... :
select 'abc' || attribut || 'def' from UneTable ;
9. Rayon maximal

- a) Le rayon de l'astre de rayon maximal. Résultat : 696342
- b) Le nom et le rayon de l'astre de rayon maximal. Résultat : (Soleil, 696342)

C- Partitionnement « group by »

10. Pour chaque catégorie, l'identifiant de la catégorie et le nombre d'astres de cette catégorie (une seule table suffit), ordonner par nombre d'astres décroissant.
Résultats (3 lignes) : (11, 2) ; (10,1) ; (12,1)
11. Nombre d'astres qui tournent autour d'un astre
 - a) Pour chaque astre, l'idA de l'astre et le nombre d'astres qui lui tournent autour. Pour simplifier, prendre que les astres qui ont au moins un autre astre qui leur tourne autour.
Résultats (2 lignes) : (100, 2) ; (101,1).
 - b) Même question, mais on veut le nom de l'astre au lieu de l'idA.
Résultats (2 lignes) : (Soleil, 2) ; (Terre, 1)
12. Pour chaque couple (astre,date), le nom de l'astre, la date de l'observation et la moyenne des valeurs des observations réalisées.
Résultats : (Soleil,2010-05-10,12002) ; (Terre,2013-12-18,8005) ; (Lune,2014-08-27)
13. Rayon maximal (suite)
 - a) Pour chaque catégorie, l'idC de la catégorie et le rayon maximal des astres de cette catégorie, ordonner par rayon croissant. Aide : une seule table suffit.
Résultats (3 lignes) : (12,1737) ; (11,6371) ; (10,696342)
 - b) On veut maintenant afficher non pas l'idC, mais le nom de la catégorie.
Pour chaque catégorie, le nom de la catégorie et le rayon maximal des astres de cette catégorie, ordonner par nombre d'astres dans chaque catégorie décroissant.
Résultats (3 lignes) : (planète, 6371) ; (étoile,696342) ; (satellite, 1737)
 - c) On veut maintenant afficher non pas le nom de la catégorie, mais le nom de l'astre.
Pour chaque catégorie, l'idC de la catégorie, le(s) nom(s) de(s) astre(s) de rayon maximal et le rayon maximal des astres de cette catégorie. Aide : deux tables suffisent.
Résultats (3 lignes) : (10, Soleil,696342) ; (11, Terre,6371) ; (12, Lune, 1737)

D- Partitionnement avec sélection de partition « group by / having »

14. L'idC des catégories pour lesquelles il existe au moins deux astres dans cette catégorie.
Résultat : 11
15. Pour chaque date (où il y a eu au moins une observation) et où il y a eu une valeur maximale observée supérieure à 8000, la date, le nombre d'observations effectuées ce jour là et la valeur maximale observée. Résultats(2 lignes):(2010-05-10,2, 12003);(2013-12-18, 1, 8005)
16. Le nom des astres pour lesquels on a fait exactement deux observations. Résultat : Soleil.
17. Le nom des astres qui ont au moins deux astres qui leur tournent autour. Résultat : Soleil

E- Réécriture double négation avec Group By et Having

18. Les dates d'observations pour lesquelles il y a au moins une observation pour chaque astre :
 - a) existant dans la bases de données.
 - b) observés au moins une fois.
 Résultat : aucun sur cet exemple
19. Catégories pour lesquelles tous les astres sont observés
 - a) L'idC des catégories pour lesquelles il existe au moins une observation pour chaque astre de cette catégorie. Résultats (2 lignes) : 10 ; 12
 - b) Même question mais on veut afficher non pas l'idC, mais le nom de la catégorie.
Résultats (2 lignes) : étoile ; satellite

masquer=1

TD 8 : CREATION DE SCHEMAS- CONTRAINTES D'INTEGRITE

Rappels Les contraintes d'intégrité permettent à l'utilisateur de définir des règles que doivent respecter les données de la base. Le plus souvent, les contraintes sont définies lors de la création des tables (CREATE TABLE). Elles peuvent néanmoins être rajoutées sur des tables existant sous certaines conditions (ALTER TABLE). Types de contraintes :

- contrainte de clé (PRIMARY KEY et UNIQUE),
- contrainte référentielle (FOREIGN KEY)
- contrainte de domaine (valeurs possibles des attributs, NOT NULL).
- contrainte de tuple (condition liant les valeurs des différents attributs pour tous les n-uplets de la table).
- contraintes globales (sur plusieurs tables) en utilisant la clause CHECK ou les assertions au moyen de CREATE ASSERTION. A noter que certains systèmes, comme Oracle, ne supportent pas les assertions.

CRÉATION D'UN SCHÉMA DE BD AVEC CONTRAINTES D'INTÉGRITÉ

On considère le schéma Entreprise décrit ci-dessous.

EMPLOYE (NumSS, NomE, PrenomE, NumChef*, VilleE, DateNaiss)

PROJET(NumProj, NomProj, RespProj*, VilleP, Budget)

EMBAUCHE (NumSS*, NumProj*, DateEmb, Profil*)

GRILLE_SAL (Profil, salaire)

La clé primaire de chaque relation est soulignée et les attributs des clés étrangères sont suivis d'un astérisque. Cette base contient des informations sur des employés et sur les projets dans lesquels ils sont impliqués. DateEnreg dans la table employe donne la date à laquelle l'employé a été enregistré dans la BD. Ces employés sont embauchés dans un projet sur un profil donné et perçoivent un salaire en fonction de ce profil. Le chef d'un employé dans la table Employé et le chef d'un projet dans la table Projet sont des employés. En plus des contraintes de clé et contraintes référentielles indiquées dans le schéma, on voudrait intégrer les contraintes suivantes :

On rajoute les contraintes suivantes :

Domaines

- Le numéro de sécurité sociale possède exactement 5 chiffres.
- Les attributs textuels (NomE, PrenomE, NomProj, Profil) ne dépassent pas 20 caractères (ils peuvent en avoir moins).
- La ville d'un employé (VilleE) ou d'un projet (VilleP) se limite à 'Paris', 'Lyon' et 'Marseille' et sa longueur ne dépasse pas 9 caractères.
- Le numéro d'un projet varie entre 5 et 7 chiffres.
- Le salaire peut avoir deux chiffres après la virgule et ne dépasse pas 90 000.
- Le budget est un entier sur 6 chiffres (sans virgule)
- Aucun employé ne peut avoir plus de 70 ans au moment où il est enregistré dans la table Employé.

Clés

- Il n'y a pas deux employés avec le même nom et le même prénom.

Not-null

- Chaque projet doit avoir un responsable.
- Dans la table Embauche, NumSS et NumProj représentent un employé et un projet existants. De même, profil est un profil existant.

EXERCICE

1. Donner en SQL les instructions de création des contraintes de domaines indiquées dans l'énoncé.
2. Donner en SQL les instructions de création du schéma de la base en leur associant les contraintes de clés, référentielles et de domaines indiquées dans l'énoncé.
3. Un responsable de projet doit habiter la ville du projet dont il est responsable. Donner en SQL l'instruction permettant d'exprimer cette contrainte de deux manière différentes :
 1. En considérant que la table projet n'a pas été créée
 2. En considérant que la table projet a été créée, dans ce cas utiliser la commande *alter table <table> add constraint <nom> <expression>* pour associer la contrainte à Projet

masquer=1

TD 9 : CREATION DES SCHEMAS - MODIFICATION DES DONNEES.

Rappels Les contraintes d'intégrité permettent à l'utilisateur de définir des conditions que doivent respecter les données. Le plus souvent, les contraintes sont définies lors de la création des tables (CREATE TABLE). Elles peuvent néanmoins être rajoutées sur des tables existant sous certaines conditions. Il existe principalement deux types de contraintes : contrainte de clé (PRIMARY KEY et UNIQUE), contrainte référentielle (FOREIGN KEY) et contrainte de domaine (clause CHECK. Toutes ces contraintes sont locales : elles sont définies pour une seule table. Il est aussi possible de définir des contraintes globales en utilisant la clause CREATE ASSERTION. A noter que certains systèmes ne supportent pas les assertions.

VILLES ET PAYS

On veut créer un schéma relationnel pour stocker des informations sur des villes et des pays.

1. Traduisez le schéma relationnel suivant en instructions SQL:

Ville(nom, population, pays*)

Pays(nom, capitale*)

où **pays** est une référence vers un pays dans la table **Pays** et **capitale** est une référence vers une ville dans la table **Ville**.

2. Insérez la France avec sa capitale Paris (3 millions d'habitants) dans la base de données.
3. Modifiez le schéma de telle manière que la suppression d'un pays déclenche automatiquement la suppression de toutes les villes du pays.
4. Effacez les deux tables **Ville** et **Pays** du schéma.

ARBRES ORDONNÉS

Voici une table **Arbre** qui permet de stocker des arbres ordonnés dans une base de données relationnelle:

```
create table Arbre(  
    id number(10),  
    par number(10),  
    pos number(2),  
    constraint pk primary key (id),  
    constraint fk foreign key (par) references Arbre (id));
```

- **id** est l'identifiant du nœud,
- **par** est l'identifiant du parent,
- **pos** est la position parmi les enfants ;

1. Insérez l'arbre binaire (1(2(4,5),3(6,7(8,9)))) dans la base de données.
2. Est-ce qu'on peut insérer des données incohérentes ?

3. Comment faut-il modifier le schéma pour empêcher cette incohérence.
4. Effacez le sous-arbre (2(4,5)) de la base de données.
5. Effacez le sous-arbre 3(6,7(8,9)) de la base de données.
6. Est-ce que l'instruction suivante est possible sur l'arbre initial (avant l'effacement) ?
delete from Arbre where id=3;
7. Comment peut-on modifier le schéma pour effacer le sous-arbre avec l'instruction précédente ?
8. Est-ce qu'il est possible de modifier l'identifiant d'un nœud ? Par exemple : *update Arbre set id=10 where id = 3;*
9. Comment peut-on modifier le schéma pour permettre l'instruction précédente ?
10. Quel est le résultat des instructions suivantes (sur l'arbre initial):

```
alter table Arbre drop constraint fk ;
alter table Arbre add constraint fk foreign key (par) references Arbre(id)
on delete set null;
delete from Arbre where id=3;
```

MISE À JOUR DE TABLES

Considérer le schéma Entreprise du TD précédent.

Le but de cet exercice est d'exprimer des instructions permettant d'insérer, de modifier et de supprimer des nuplets.

Dire à chaque fois si l'instruction exprimée est acceptée ou rejetée par le système en justifiant.

INSERTIONS

1. Insérer l'employé identifié par '12456' qui se prénomme 'Alain'.
2. Insérer l'employée identifiée par '21456' qui s'appelle 'LARS Anna', qui habite 'Paris' et qui est née le 25-08-1975.
3. Insérer le projet numéro '78143' dénommé 'ORCA' qui s'opère sous la responsabilité de 'Lars Anna' à Paris et qui a pour budget 250 000 euros.
4. Renseigner dans la base les salaires correspondants aux profils suivants :
 1. 'Responsable' → 80 000
 2. 'Développeur' → 45 000
 3. 'Technicien' → 35 000
5. Renseigner dans la base le fait que 'Alain' a été embauché dans le projet 'ORCA' en tant que testeur en date du 01-04-2014.
6. Renseigner dans la base le fait que 'LARS Anna' fut embauchée dans le projet 'MEDUSA' en date du 28-02-2012 en tant que 'Développeur'.

SUPPRESSIONS.

Note. On considère que les questions suivantes sont indépendantes.

7. Supprimer les employés de plus de 67 ans
8. Supprimer les employés prénommés 'Alain'
9. Supprimer l'employée 'LARS Anna'

10. Supprimer les employés embauchés lors de la dernière année.
11. Supprimer les projets dont la proportion des salaires dépasse la moitié du budget.

MISES À JOUR

12. Désormais on connaît que l'employé Alain porte le nom BERNARD. Répercuter cette information.
13. L'employée LARS Anna est promue à la tête du projet identifié par 78143. Elle doit d'abord déménager avant de prendre ses responsabilités. Donner l'instruction qui modifie la ville de cette employée.
On suppose qu'il existe un nuplet dans la table Projet avec les données suivantes :
(78143,'ORCA',null,'paris',250000)
14. Le budget de chaque projet équivaut au double des salaires de ses employés. Donner une instruction qui modifie les budgets des projets en conséquence.

masquer=1

TD 10 : PL/SQL

PROGRAMMES PL/SQL

Nous travaillerons dans ce TD sur les trois tables suivantes qui permettent de décrire des employés, des projets, l'affectation des employés aux projets et les grilles de salaire pour différents profils de postes :

EMPLOYE (NumSS, NomE, PrenomE, VilleE, DateNaiss)
PROJET (NumProj, NomProj, RespProj*, VilleP, Budget)
EMBAUCHE (NumSS*, NumProj*, DateEmb, Profil*)
GRILLE_SAL (Profil, Salaire)

La clé primaire de chaque table est soulignée, les clés étrangères sont signalées par un astérisque.

Exercices

1. Expliquez le fonctionnement du programme PL/SQL suivant:

```
DECLARE
    trouve BOOLEAN;
BEGIN
    FOR r IN (Select numproj, nomproj, villep from Projet ORDER BY nomproj) LOOP
        dbms_output.put_line('Projet: '||r.nomproj||' dans la ville '||r.villep);
        trouve := FALSE;
        for r2 IN (Select nome, prenome from employe e, embauche b
                    Where e.numss=b.numss and b.numproj=r.numproj )
            LOOP
                dbms_output.put_line('Employé: '|| r2.nome||', '|| r2.prenome);
                trouve := TRUE;
            END LOOP;
        IF(trouve = FALSE) THEN
            dbms_output.put_line('Pas d"employé');
        END IF;
    END LOOP;
END;
```

2. (*Curseur*). Modifiez le programme précédent afin de remplacer la première boucle FOR par un curseur.

3. (*Curseur avec des paramètres*). Modifiez le programme de la question 2 afin de remplacer la boucle FOR par un curseur avec des paramètres. Utilisez les attributs des curseurs au lieu de la variable trouve pour vérifier l'employé.

4. (*Curseur Implicite*). Écrivez un bloc PL/SQL anonyme qui supprime tous les employés de la table EMPLOYE qui ont 50 ans ou plus et qui affiche le nombre de lignes qui ont été supprimées ou le message 'Aucun employé supprimé' si aucun employé n'a plus de 50 ans. Utilisez les attributs

d'un curseur implicite.

5. Optionnel - (Curseur et CASE). Écrire un bloc anonyme qui augmente tous les salaires dans la table Grille_Sal. Les salaires inférieurs à 40000 sont augmentés de 30%, les salaires compris entre 40000 et 60000 de 20% et les salaires supérieurs à 60000 sont augmentés de 10%. Utilisez un CASE pour tester les différentes valeurs possibles du salaire.

6. (*Exception prédéfinie*). Écrire une procédure PL/SQL qui a 3 paramètres: le *numSS*, le *nomE* et *prenomE* d'un nouvel employé à insérer dans la table Employé. S'il existe déjà un autre employé avec le même numSS dans la table Employé elle doit afficher un message d'erreur sans insérer l'utilisateur. Dans le cas contraire la procédure doit insérer le nouvel employé et afficher un message de confirmation. Utilisez l'exception prédéfinie **NO_DATA_FOUND**.

7. (*Exception utilisateur*). Écrire une procédure qui prend comme paramètre le numéro et le budget d'un projet et qui l'ajoute à la table projet seulement si la somme tous les budgets après l'ajout ne dépasse pas 400000. Dans le cas contraire le projet n'est pas ajouté et un message d'erreur est affiché à l'utilisateur. Utilisez une exception utilisateur.

masquer=1

TD 11 : TRIGGERS

1. RAPPELS TRIGGERS

La syntaxe d'une expression de création de trigger en SQL3 est la suivante :

1. CREATE TRIGGER <nom-trigger>
2. BEFORE | AFTER | INSTEAD OF
3. INSERT | DELETE | UPDATE OF <liste_attributs>
4. ON <nom-table>
5. [ORDER <valeur de priorité>]
6. [REFERENCING NEW | OLD AS <nom-variable>]
7. FOR EACH ROW | STATEMENT
8. [WHEN (<conditionSQL>)]
9. [DECLARE]
10. <variables locales>
11. BEGIN <actionSQL> END ;

Les expressions entre [...] sont optionnelles. Le symbol '|' sépare les options :

- Ligne 1: <nom-trigger> indique le nom du trigger.
- Ligne 2: indique si le trigger est déclenché avant (BEFORE), après (AFTER) ou à la place (INSTEAD OF) d'un événement
- Ligne 3: indique le type de l'événement
- Ligne 4: indique la table concernée par l'événement
- Ligne 5: la clause ORDER est optionnelle et sert à gérer les priorités entre des triggers en conflit
- Ligne 6: la clause REFERENCING sert à nommer une ou plusieurs variables temporaires utilisables dans la partie condition et dans la partie action. NEW et OLD désignent respectivement,
1. la valeur du dernier n-uplet modifié par l'événement si la granularité de déclenchement est ROW (ligne 7)
 2. l'ensemble des nuplets touchés par l'événement si la granularité est STATEMENT (ligne 7).
- Le nom de la variable est :nom_variable (avec un ':' ajouté au début dans le reste du trigger. Si la clause REFERENCING est omise, alors les variables s'appellent par défaut :new et :old dans le reste du trigger.
- Ligne 7: la clause FOR EACH détermine la granularité de déclenchement:
- ROW : la règle est déclenchée à chaque n-uplet touché par l'événement.
 - STATEMENT : la règle est déclenchée qu'une seule fois pour l'événement (**Oracle** : cette ligne est vide pour FOR EACH STATEMENT).
- Ligne 8: permet d'indiquer une condition SQL qui doit être vraie pour déclencher le trigger.
- Ligne 9: lause optionnelles pour la déclaration de variables locales
- Ligne 10: déclaration de variables locales (optionnelles)
- Ligne 11: indique les actions à effectuer (en PLSQL) – voir TME7 pour plus d'explications.

Il est évident qu'à un événement de type INSERT (resp. DELETE) ne peut pas correspondre une delta-structure OLD (resp. NEW). Une règle AFTER ne devrait en principe pas modifier la valeur de la variable temporaire déclarée par NEW (cette valeur est déjà écrite dans la base), une règle BEFORE ne devrait en principe pas modifier la base par une commande INSERT, UPDATE ou

DELETE sur les relations de la base (puisque l'événement n'a pas encore eu lieu réellement), elle ne peut modifier que les variables temporaires.

2. BASE DE DONNÉES « ENTREPRISE »

La base de données d'une entreprise contient les trois tables suivantes :

```
CREATE TABLE EMPLOYE (  
    ID_EMP NUMBER(8) PRIMARY KEY,  
    NOM VARCHAR(32) NOT NULL,  
    PRENOM VARCHAR(32) NOT NULL,  
    FONCTION VARCHAR(32),  
    SALAIRE NUMBER(7,2) NOT NULL);  
  
CREATE TABLE PROJET (  
    ID_PROJ NUMBER(8) PRIMARY KEY,  
    NOM VARCHAR(32) NOT NULL,  
    ID_CHEF_PROJET NUMBER(8) REFERENCES EMPLOYE  
        ON DELETE SET NULL  
        ON UPDATE CASCADE);  
  
CREATE TABLE PARTICIPE (  
    ID_EMP NUMBER(8) REFERENCES EMPLOYE  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    ID_PROJET NUMBER(8) REFERENCES PROJET  
        ON DELETE RESTRICT  
        ON UPDATE RESTRICT,  
    PRIMARY KEY (ID_EMP, ID_PROJET));
```

La table EMPLOYE contient pour chaque employé son identifiant, son nom et prénom, sa fonction (optionnelle), et son salaire. La table PROJET contient les identifiants et noms des projets et une référence vers l'employé qui dirige le projet. La table PARTICIPE stocke les employés avec les projets auxquels ils participent.

Rappel **Oracle** :

1. ON UPDATE CASCADE n'est pas implanté.
2. ON DELETE RESTRICT est utilisé par défaut.

2.1 Écrire un trigger BEFORE qui évite qu'un salaire ne puisse diminuer (l'ancien salaire est maintenu en cas de diminution par une mise-à-jour).

2.2 Écrire un trigger AFTER qui évite qu'un salaire ne puisse diminuer (l'ancien salaire est maintenu en cas de diminution par une mise-à-jour).

2.3 Écrire deux triggers nb_empl_avant et nb_empl_après qui affichent pour chaque opération de suppression dans la table EMPLOYE, le nombre d'employés avant et après la suppression. Vous pouvez utiliser la fonction DBMS_OUTPUT.PUT_LINE.

2.4 Écrire un trigger qui annule les transaction avec des opérations qui suppriment plus de 50 n-uplets dans la relation EMPLOYE.

2.5 Écrire un trigger AFTER qui *simule* les actions ON DELETE | UPDATE associées aux clés étrangères de la table PROJET (on enlève les actions définies dans le schéma pour les remplacer par un trigger) :

- quand un EMPLOYE est effacé, l'attribut ID_CHEF_PROJET de la table PROJET est mis à NULL pour les projets concernés.
- quand l'identifiant d'un EMPLOYE est mis-à-jour, l'attribut ID_CHEF_PROJET de la table PROJET est mis à jour pour les projets concernés.

2.6 Écrire un trigger AFTER qui *simule* les actions ON DELETE | UPDATE associées à la clé étrangère ID_EMP de la table PARTICIPE :