

Contrôle de Bases de Données  
Vendredi 06 mars 2023

Exercice 1 (12 pts)

Soit le schéma de la base de données ci-dessous modélisant une partie des informations concernant un tournoi de tennis.

On s'intéresse aux joueurs, aux matchs disputés, aux juges et aux arbitrages.

JOUEURS(IDJoueur, NomJoueur, Sexe, Pays)  
MATCHS(IDMatch, PoseMatch, Court)  
JUGES(IDJuge, NoMatch, NomJuge, Pays)  
ARBITRAGES(IDJuge, NoMatch, TypeJuge ) DISPUTÉS(IDJoueur,  
NoMatch)

Questions : écrire les requêtes ci-dessous en SQL.

1. Quels sont les noms et les pays des juges de ligne

SELECT NomJuge, Pays FROM JUGES WHERE TypeJuge='ligne'

2. Quels sont les noms des juges qui ont arbitré un match sur le court Suzanne-Lenglen

SELECT NomJuge FROM JUGES WHERE NoMatch IN (SELECT NoMatch FROM ARBITRAGES WHERE TypeJuge = 'principal' AND NoMatch IN (SELECT NoMatch FROM MATCHS WHERE Court = 'Suzanne-Lenglen'))

3. Pour chaque match, quels sont les joueurs (NomJoueur) qui ont disputé ce match

SELECT NomJoueur FROM JOUEURS WHERE IDJoueur IN (SELECT IDJoueur FROM DISPUTÉS WHERE NoMatch = [ID du match])

4. Pour chaque court, quel est le nombre des matchs disputés lors d'un tournoi

SELECT Court, COUNT(\*) AS NombreMatchs FROM MATCHS GROUP BY Court

5. Quels les joueurs qui n'ont participé à aucun match

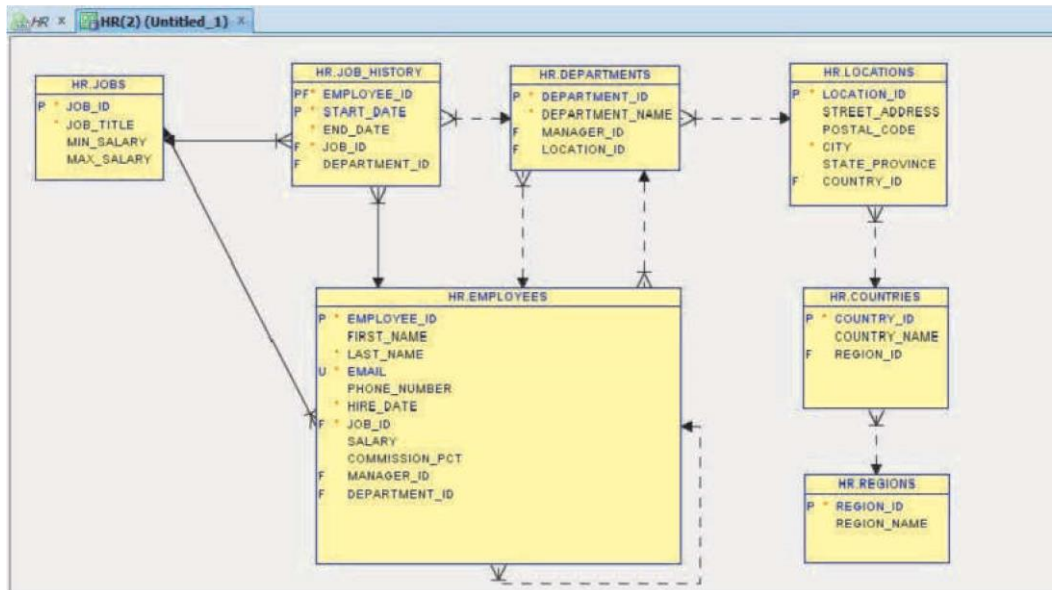
SELECT NomJoueur, Pays FROM JOUEURS WHERE IDJoueur NOT IN (SELECT IDJoueur FROM DISPUTÉS)

6. Quels sont les noms et pays des juges qui ont arbitré au moins un match

SELECT NomJuge, Pays FROM JUGES WHERE IDJuge IN (SELECT IDJuge FROM ARBITRAGES)

## Exercice ( 8 pts)

Considérons le modèle physique de données (MPD) du schéma HR d'oracle (voir image)



1. Décrire pour chaque table, les contraintes d'intégrité et relationnelle
2. Ecrire une requête qui retourne pour chaque employé, le nom, le salaire, l'emploi occupé (job\_title), la date d'embauche (hire\_date) ainsi que le nom du département.
3. Ecrire une requête qui retourne le nom du manager, le nom du département de l'employé 151

1/

Les contraintes d'intégrités de la table HR.JOBS sont :

- P\*JOB\_ID indique qu'elle doit contenir une valeur et qu'elle doit correspondre à la clé primaire de la table HR.JOBS.
- MIN\_SALARY" et " MAX\_SALARY " peuvent être définies avec une valeur par défaut. Si aucune valeur n'est spécifiée pour ces colonnes, la valeur par défaut sera utilisée

Les contraintes relationnelles de la HR.JOBS sont :

- Il y'a une contrainte relationnelle si cette table est utilisée comme une table de référence pour une autre table, une contrainte de clé étrangère pourrait être définie pour assurer la cohérence des données entre les deux tables.

Les contraintes d'intégrités de la table HR.JOB\_HISTORY sont :

- PF\*EMPLOYEE\_ID indique que la colonne EMPLOYEE\_ID doit contenir une valeur et que cette valeur doit correspondre à la clé primaire de la table HR.EMPLOYEES.
- P\*START\_DATE indique que la colonne START\_DATE doit contenir une et que cette valeur doit être unique
- \*END\_DATE indique que la colonne END\_DATE peut contenir une valeur nulle, mais si elle contient une valeur, elle doit être supérieure ou égale à la valeur de la colonne START\_DATE pour le même enregistrement.

Les contraintes relationnelles de la table HR.JOB\_HISTORY sont :

- F\*JOB\_ID indique la colonne JOB\_ID doit contenir une et que cette valeur doit correspondre à la clé primaire de la table HR.JOBS.
- F DEPARTEMENT\_ID indique que la colonne DEPARTMENT\_ID peut contenir une valeur nulle, mais si elle contient une valeur, elle doit correspondre à la clé primaire de la table HR.DEPARTMENTS.

Les contraintes d'intégrités de la table HR.DEPARTMENTS sont :

- P\*DEPARTMENT\_ID indique que la colonne DEPARTMENT\_ID doit contenir une valeur unique et qu'elle doit correspondre à la clé primaire de la table HR.DEPARTMENTS.
- \*DEPARTMENT\_NAME indique que la colonne DEPARTMENT\_NAME doit contenir une valeur et que cette valeur doit être unique.

Les contraintes relationnelles de la table HR.DEPARTMENTS sont :

- F MANAGER\_ID indique que la colonne MANAGER\_ID peut contenir une valeur nulle, mais si elle contient une valeur, elle doit correspondre à la clé primaire de la table HR.EMPLOYEES.
- F LOCATION\_ID indique que la colonne LOCATION\_ID peut contenir une valeur nulle, mais si elle contient une valeur, elle doit correspondre à la clé primaire de la table HR.LOCATIONS.

Les contraintes d'intégrités de la table HR.LOCATIONS sont :

- P\*LOCATION\_ID indique que la colonne LOCATION\_ID doit contenir valeur unique et qu'elle doit correspondre à la clé primaire de la table HR.LOCATIONS.
- \*CITY indique que la colonne CITY doit contenir une valeur et que cette valeur doit être unique pour chaque enregistrement de la table.

Les contraintes relationnelles de la table HR.LOCATIONS sont :

- COUNTRY\_ID indique que la colonne COUNTRY\_ID doit contenir une valeur et que cette valeur doit correspondre à la clé primaire de la table HR.COUNTRIES.

Les contraintes d'intégrités de la table HR.COUNTRIES sont :

- P\*COUNTRY\_ID indique que la colonne COUNTRY\_ID doit contenir une valeur unique et qu'elle doit correspondre à la clé primaire de la table HR.COUNTRIES.
- COUNTRY\_NAME indique que la colonne COUNTRY\_NAME doit contenir une valeur et que cette valeur doit être unique .

Les contraintes relationnelles de la table HR.COUNTRIES sont :

- F\* REGION\_ID indique que la colonne REGION\_ID peut contenir une valeur nulle, mais si elle contient une valeur, elle doit correspondre à la clé primaire de la table HR.REGIONS.

Les contraintes d'intégrités de la table HR.REGIONS sont :

- P\*REGION\_ID : indique que la colonne REGION\_ID doit contenir une valeur unique et qu'elle doit correspondre à la clé primaire de la table HR.REGIONS.
- REGION\_NAME indique que la colonne REGION\_NAME doit contenir une valeur) et que cette valeur doit être unique .

Les contraintes relationnelles de la table HR.REGIONS sont :

- Il n'y en a pas

2/ SELECT e.last\_name AS "Nom", e.salary AS "Salaire", j.job\_title AS "Poste occupé", e.hire\_date AS "Date d'embauche", d.department\_name AS "Nom du département" FROM employees e JOIN departments d ON e.department\_id = d.department\_id JOIN jobs j ON e.job\_id = j.job\_id;

3 /SELECT m.last\_name AS "Nom du manager", d.department\_name AS "Nom du département" FROM employees m JOIN departments d ON m.department\_id = d.department\_id WHERE m.employee\_id = (SELECT manager\_id FROM employees WHERE employee\_id = 151);