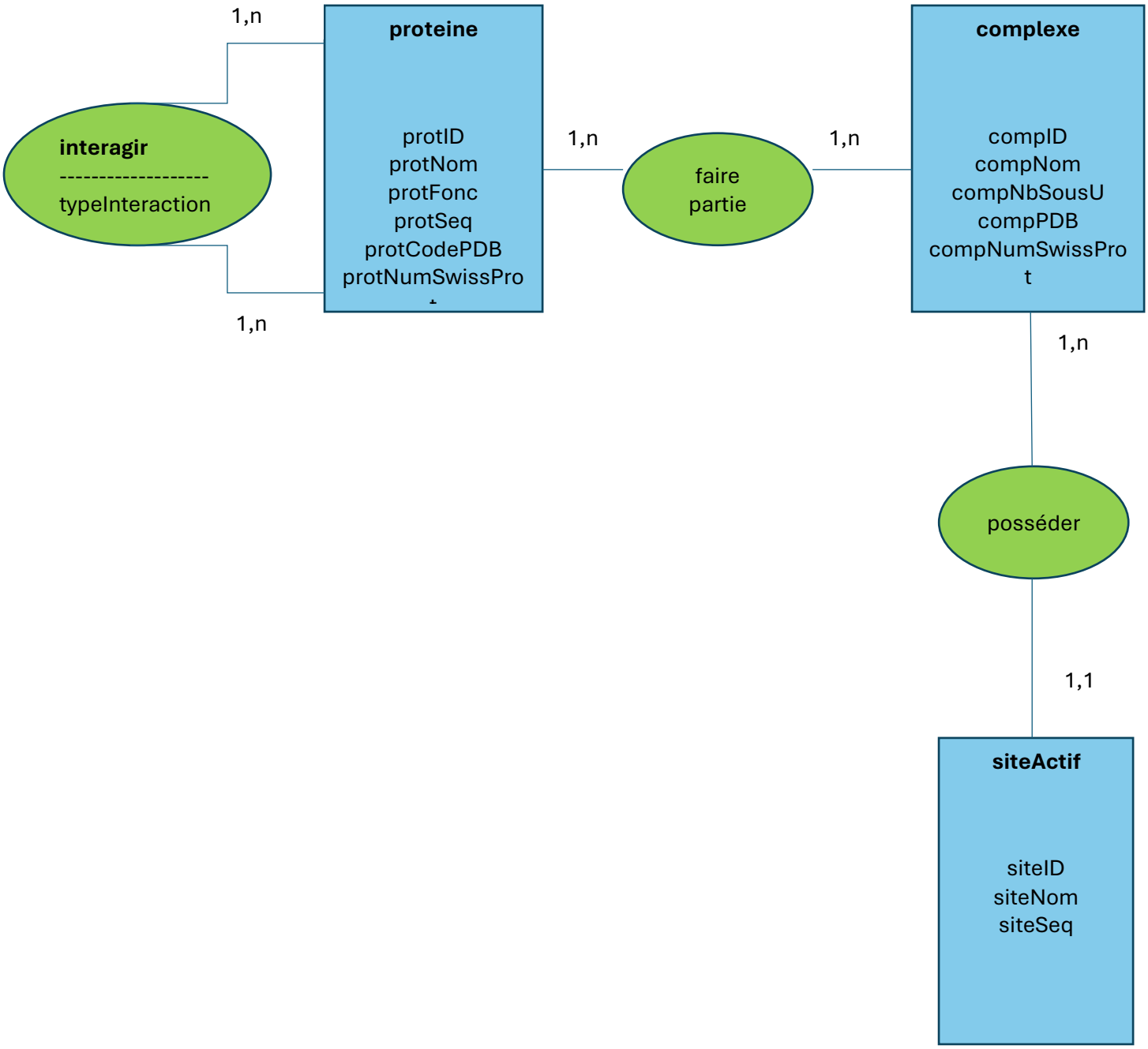
BNF101 - Exercice interactions protéine-protéine

Corrigé :

1/ MCD : Modèle Conceptuel de Données de la base de données «Interactions protéines/protéines.

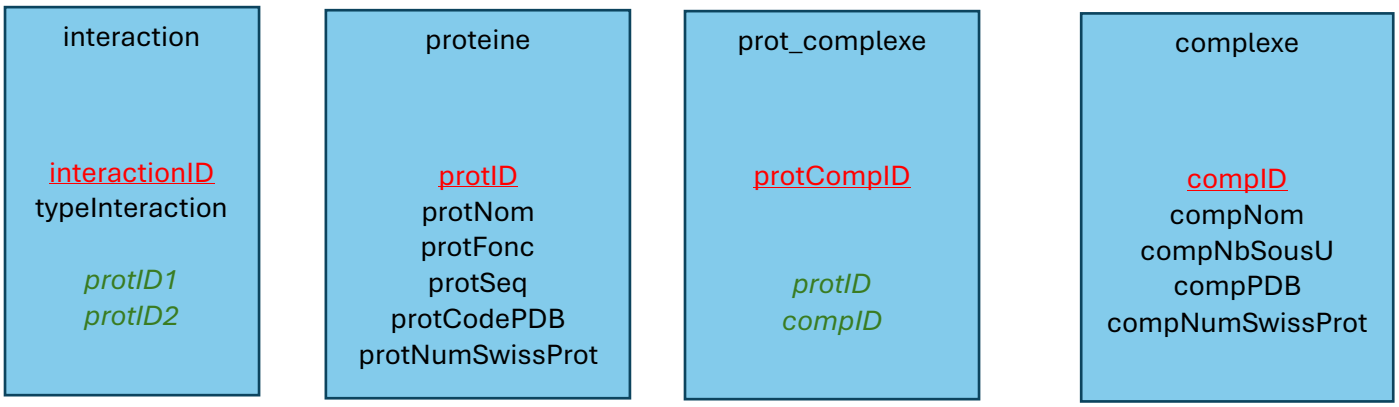


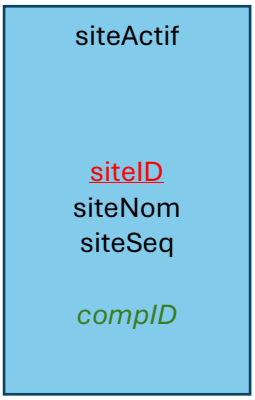
Remarque :

Dans cette exercice, on suppose qu'une protéine interagit au moins avec une protéine et qu'une protéine fait partie d'au moins un complexe.

C'est pourquoi on indique des cardinalités (1,n).

2 et 3/ MPD modèle physique de données:
Nous allons maintenant appliquer les règles de la méthode Merise pour traduire le MCD en MPD (Modèle Physique de Données).
Les clés primaires sont colorées en rouge et soulignées.
Les clés étrangères sont colorées en vert et *en italique*.





4/ Création des tables siteActif et complexe :

```
CREATE TABLE complexe
(
compID NUMBER(5),
compNom VARCHAR2(30),
compNbSousU NUMBER(2),
compPDB VARCHAR2(8),
compNumSwissProt NUMBER(5),
CONSTRAINT c_pk_compID PRIMARY KEY (compID)
);

CREATE TABLE siteActif
(
siteID NUMBER(5),
siteNom VARCHAR2 (30),
siteSeq VARCHAR2 (200),
compID NUMBER(5),
CONSTRAINT c_pk_siteID PRIMARY KEY (siteID),
CONSTRAINT c_fk_compID FOREIGN KEY (compID) REFERENCES
complexe(compID)
);
```

Il faut d'abord créer la table "**complexe**" dans la mesure où la table "**siteActif**" possède une clé étrangère **compID** qui référence la clé primaire **compID** de la table "**complexe**".

5/ Requêtes SQL:

- obtenir la liste complète des protéines:

```
SELECT * FROM proteine;
```

- obtenir la liste complète des complexes:

```
SELECT * FROM complexe;
```

- obtenir le nom des protéines, le nom et la séquence des sites actifs de ces protéines:

forme historique :
SELECT a.protNom, d.siteNom, d.siteSeq FROM protein a, protein_complexe b, complexe c, siteActif d WHERE
a.protID=b.protID AND
b.compID=c.compID AND
c.compID=d.compID;

forme INNER JOIN :
SELECT a.protNom, d.siteNom, d.siteSeq FROM protein a
INNER JOIN protein_complexe b ON a.protID=b.protID
INNER JOIN complexe c ON b.compID=c.compID
INNER JOIN siteActif d ON c.compID=d.compID;

- obtenir le nom des protéines et le nom des complexes pour les complexes qui ont 3 sous-unités:

forme historique :
SELECT a.protNom, c.compNom FROM protein a, protein_complexe b, complexe c WHERE
a.protID=b.protID AND
b.compID=c.compID AND compNbSousU=3;

forme INNER JOIN :

SELECT a.protNom, c.compNom FROM protein a
INNER JOIN protein_complexe b ON a.protID=b.protID
INNER JOIN complexe c ON b.compID=c.compID
WHERE compNbSousU=3;

- obtenir les noms des protéines dont l'interaction est de type «coiled-coil»:

forme HISTORIQUE :

SELECT a.protNom as **proteineNom1**, b.protNom as **proteineNom2** FROM protein **a**, protein **b**, interaction **c** WHERE
a.protID = **c**.protID1 **AND b**.protID = **c**.protID2
AND
c.typeInteraction = 'coiled-coil';

forme INNER JOIN :

SELECT **a**.protNom as **proteineNom1**, **b**.protNom as **proteineNom2** FROM **interaction c** INNER JOIN **protein a** ON **a**.protID = **c**.protID1 INNER JOIN **protein b** **b**.protID = **b**.protID2)
WHERE **c**.typeInteraction = 'coiled-coil';

remarque :

Dans cette requête, pour extraire les interactions entre les protéines, il faut penser à **matérialiser les 2 tables « protéine »** dans la requête : cf **proteine a et proteine b.**

Pour bien comprendre la structure de cette requête, vous pouvez comparer le résultat de cette requête avec 2 jointures et la matérialisation de 2 tables protéines avec une autre requête qui réalise 1 jointure 1 seule table PROTEINE.

>> exemples de requête avec 1 seule table protéine :

SELECT a.protNom FROM protein a
INNER JOIN interaction b ON a.protID = b.protID1 **AND** a.protID = b.protID2
WHERE b.typeInteraction = 'coiled-coil';

Ou bien

SELECT a.protNom FROM protein a
INNER JOIN interaction b ON a.protID = b.protID1 **OR** a.protID = b.protID2
WHERE b.typeInteraction = 'coiled-coil';

Vous pouvez simuler les requêtes sur les tables ci-dessous :

PROTEINE

protID	protNom
1	WWWW
2	XXXX
3	YYYY
4	ZZZZ

INTERACTION

ProtID1	ProtID2	typeInteraction
2	3	
4	1	coiled-coil
2	2	coiled-coil
3	3	

Réponse à la question subsidiaire :

On peut on peut optimiser l'enregistrement des types d'interaction pour qu'ils soient en 1ère forme normale.

On peut ainsi créer une table TYPE_INTERACTION avec 2 colonnes :

- TYPE_INTERACTION_ID
- TYPE_INTERACTION_NOM

TYPE_INTERACTION_ID serait positionné comme une clé étrangère dans la table INTERACTION.