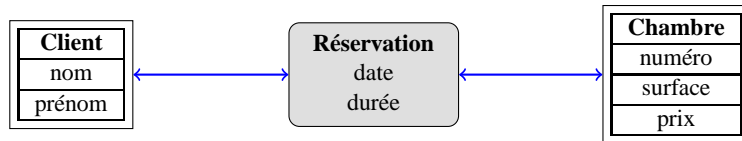


CORRECTION DU TD ITC² N° 5: BASES DE DONNÉES RELATIONNELLES

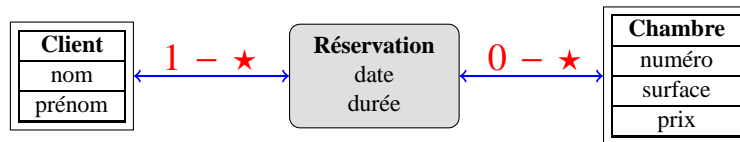
MODÈLE ENTITÉ-ASSOCIATION ET FORMULATION DES REQUÊTES SQL

EXERCICE N°1: Modèle entité-association

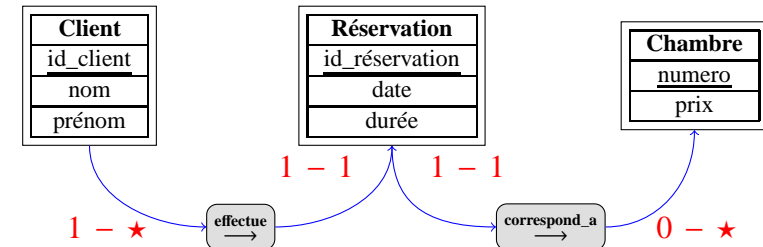


- Il peut y avoir des homonymes, on ne peut par conséquent pas utiliser l'attribut nom comme clé primaire de la table puisqu'il ne permettrait pas d'identifier de manière unique un enregistrement. Pour éviter toute ambiguïté, on ajoute un attribut supplémentaire identifiant de manière unique chaque client, par exemple un entier `id_client`.
- Chaque numéro étant attribué à une unique chambre, l'attribut `numéro` constitue une clé primaire.
- Cardinalité de $\{\text{client} \rightarrow \text{réserve}\}$: un client, s'il est répertorié dans la table a forcément procédé un jour à une réservation (date, durée), et peut également avoir réservé plusieurs fois une chambre; la cardinalité est donc $1 - *$
 - Cardinalité de $\{\text{réserve} \rightarrow \text{chambre}\}$: une chambre peut ne jamais avoir été réservée, et avoir été réservée plusieurs fois; la cardinalité est $0 - *$.

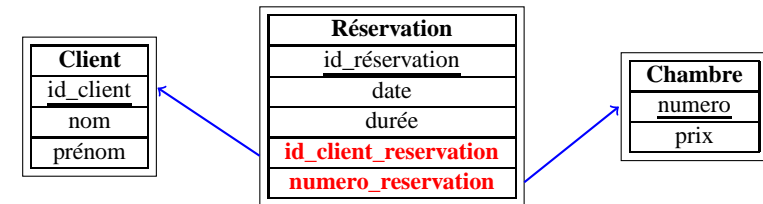
Le schéma élémentaire est donc:



- On propose pour la table `Réserve` suivante avec les cardinalités des associations qui en découlent:



Enfin, on peut bâtir la table `Réserve` avec les clés étrangères:



- Proposer une requête SQL renvoyant la liste des noms et prénoms des clients occupant les chambres dont les numéros sont compris entre 100 et 200:

RÉPONSE:

```
SELECT nom, prénom FROM (Client join Réserve
ON id_client=id_client_reservation) join Chambre ON nu-
mero_reservation=numero WHERE numero>=100 and numero<=200
```

Comme toutes les clés étrangères, la clé `numero_reservation` prend les mêmes valeurs que la clé `numero` dont elle dérive; par conséquent la seconde jointure n'est pas nécessaire, et on peut écrire:



```
SELECT nom, prénom FROM Client join Reservation ON
id_client=id_client_reservation WHERE numero_reservation>=100 and
numero_reservation<=200
```

EXERCICE N°2: Classification périodique des éléments: requêtes SQL

On reprend dans cet exercice la base de données explorée en cours et comportant les tables **tableau**, **découvertes**, et **grandeurs** dont on rappelle la structure:

nom	numéro_atomique	symbole	colonne	ligne	bloc
Hydrogène	1	H	1	1	s
Hélium	2	He	18	1	s
Lithium	3	Li	1	2	s
Béryllium	4	Be	2 (IIA)	2	s

Table 1: La table **tableau**

nom	pays	symbole	date
Henry Cavendish	Grande-Bretagne	H	1766
Jules Janssen	Grande-Bretagne	He	1895
Joseph Norman Lockyer	Grande-Bretagne	He	1895
Johan August Arfwedson	Suède	Li	1817

Table 2: Table découvertes

symbole	remplissage	masse_atomique	temp_fusion	temp_ébullition
H	1	1,00794		
He	2	4,002602		
Li	2 1	6,94100	180,5	1342
Be	2 2	9,012182	1287	2471

Table 3: Table grandeurs

- Créer une table contenant la liste des symboles des éléments, leur numéro atomique et le remplissage de leurs couches:

RÉPONSE:

```
SELECT tableau.symbole,numéro_atomique,remplissage FROM tableau JOIN
grandeurs ON tableau.symbole=grandeurs.symbole
```

- Créer une table contenant le nombre d'éléments, le plus grand et le plus petit numéro atomique, la moyenne des masses atomiques de chaque ligne:

RÉPONSE:

```
SELECT COUNT(*),MAX(numéro_atomique),MIN(numéro_atomique),
AVG(masse_atomique) FROM tableau JOIN grandeurs ON
tableau.symbole=grandeurs.symbole GROUP BY ligne
```

- Déterminer l'élément chimique dont la découverte est attribuée à deux chercheurs français et de masse atomique comprise entre 20 et 220 ($g.mol^{-1}$):

RÉPONSE:

```
SELECT tableau.nom FROM (découverte JOIN grandeurs
ON découverte.symbole=grandeurs.symbole) JOIN tableau ON
tableau.symbole=grandeurs.symbole WHERE découverte.pays="France"
AND (grandeurs.masse_atomique>20 AND grandeurs.masse_atomique<220)
GROUP BY découverte.symbole HAVING COUNT(découverte.nom)=2
```

- Déterminer le pays où l'on a découvert au XIX^{ème} siècle un élément chimique du bloc f et dont le nom commence par A; quel est de plus cet élément?:

RÉPONSE:

```
SELECT découverte.pays,tableau.nom FROM découverte JOIN tableau ON dé-
couverte.symbole=tableau.symbole WHERE date>1799 AND date<1900 AND
bloc="f" AND tableau.nom<"B"
```

- Créer une table contenant tous les attributs des gaz rares (colonne 18) ; attention, les numéros de colonne ne sont pas des entiers!:

RÉPONSE:

Cette requête est la plus compliquée de l'exercice. On la reprend ici par étapes.

On peut proposer pour commencer ceci:

```
SELECT * FROM (découverte JOIN tableau ON découverte.symbole=tableau.symbole) JOIN grandeurs ON tableau.symbole=grandeurs.symbole WHERE colonne>="18" AND colonne<"2" ORDER BY numéro_atomique
```

Cependant, vous allez constater que cette requête conduit à l'affichage de "pseudo-doublons" si l'on considère le nom des éléments car certains éléments ont été découverts par plusieurs chimistes; en effet, chaque enregistrement extrait est unique puisqu'il est au moins différencié par la valeur d'un des attributs: le nom du découvreur de l'élément chimique en question, il y aura des multiplets si l'on considère le nom des éléments! Ce n'est donc pas un véritable doublon au sens strict. Evidemment, cela ne change rien d'ajouter "**distinct**" après "**select** puisque les "doublons" constatés n'en sont pas en réalité! (essayez donc!)

Pour remédier au problème, il suffit de regrouper entre eux les éléments de même masse atomique, ou symbole, ou numéro atomique, avec la commande "**group by**" en remplacement de "**order by**"; je vous propose ici de regrouper par numéro_atomique ce qui permettra un affichage dans l'ordre de la classification (le regroupement par masse atomique conduit évidemment au même résultat, mais pas celui par symbole):

```
select * from (tableau join grandeurs on tableau.symbole=grandeurs.symbole) join découverte on grandeurs.symbole=découverte.symbole where colonne>="18" and colonne<"2" group by numéro_atomique
```

6. Créer une table contenant le symbole, le pays de découverte et les lignes des éléments des trois premières lignes n'ayant pas été découverts ni par des anglais, ni par des allemands:

RÉPONSE:

On peut proposer ici plusieurs variantes:

```
SELECT DISTINCT tableau.symbole,découverte.pays,tableau.ligne FROM tableau JOIN découverte ON tableau.symbole=découverte.symbole WHERE tableau.ligne<=3 AND pays NOT IN (SELECT pays FROM découverte WHERE pays="Grande-Bretagne" OR pays="Allemagne" )
```

Autres possibilités de formulation:

```
SELECT DISTINCT tableau.symbole, découverte.pays, tableau.ligne FROM tableau JOIN découverte ON tableau.symbole=découverte.symbole WHERE tableau.ligne<="3" AND NOT (pays="Grande-Bretagne" OR pays="Allemagne")
```

ou encore le plus simple:

```
SELECT DISTINCT t.symbole, d.pays, t.ligne FROM (tableau t JOIN découverte d on t.symbole=d.symbole) WHERE t.ligne<=3 AND d.pays<>"Allemagne" AND d.pays<>"Grande-Bretagne"
```

7. Trouver le nom de l'élément du bloc d qui a le plus grand rapport entre sa masse atomique et son numéro atomique:

RÉPONSE:

```
SELECT nom,(masse_atomique/numéro_atomique) FROM tableau JOIN grandeurs on grandeurs.symbole=tableau.symbole WHERE bloc="d" GROUP BY (numéro_atomique/masse_atomique) LIMIT 1
```

Une autre solution, un peu plus longue mais explicite, est de dégager le plus grand rapport de masse atomique sur numéro atomique par une sélection:

```
SELECT nom FROM (tableau t JOIN grandeurs g ON t.symbole=g.symbole) WHERE bloc="d" AND masse_atomique/numéro_atomique=(SELECT MAX(masse_atomique/numéro_atomique) FROM (tableau t JOIN grandeurs g ON t.symbole=g.symbole) WHERE bloc="d")
```

ou encore (fonctionnelle uniquement si l'on s'autorise à afficher ,MAX(masse_atomique/numéro_

```
SELECT nom,MAX(masse_atomique/numéro_atomique) FROM tableau t JOIN grandeurs g ON g.symbole=t.symbole WHERE bloc="d"
```

8. Trouver le chercheur qui a découvert le plus grand nombre d'éléments:

RÉPONSE:

```
SELECT nom,count(nom) FROM découverte GROUP BY nom ORDER BY 1.0/(1.0+COUNT(nom)) LIMIT 1
```

9. Trouver le chercheur français qui a trouvé exactement deux éléments différents du radium Ra:

RÉPONSE:

```
SELECT nom,count(nom) FROM decouverte WHERE symbole<>"Ra" AND  
pays= "France" GROUP BY nom HAVING COUNT(symbole)=2
```

10. Trouver les métaux du bloc d qui sont liquides en-dessous de 20°C:

RÉPONSE:

On s'attend à obtenir le mercure, seul métal liquide à température ambiante:

```
SELECT t.nom FROM tableau t JOIN grandeurs g ON t.symbole=g.symbole  
WHERE t.bloc="d" AND g.temp_fusion<20.0
```

11. Déterminer la valeur moyenne du nombre de neutrons par protons par ligne de la classification qui compte strictement plus de deux éléments:

RÉPONSE:

```
SELECT AVG(round(masse_atomique)-numéro_atomique)/(numéro_atomique))  
FROM tableau t JOIN grandeurs g ON t.symbole=g.symbole group BY t.ligne  
HAVING COUNT(*)>2
```