

Sujet 1 – 3H – documents autorisés
Enseignant : Christine Plumejeaud

Modèle entité-association – 6pts

Le service de gestion du personnel d'une entreprise désire s'équiper d'un outil lui permettant de gérer informatiquement ses employés, leurs salaires et leurs congés. Les spécifications suivantes ont pu être produites à partir d'une analyse des besoins réalisée auprès des utilisateurs du service du personnel.

Généralités :

- Tout employé est identifié par un nom, un prénom et une date de naissance.
- Tout employé remplit une fonction et appartient à un service
- Pour chaque employé, on gère la date d'embauche et la quotité (c'est-à-dire le pourcentage de temps travaillé par rapport au temps plein, en cas de travail à temps partiel)

Gestion des salaires :

- Pour chaque employé, on gère l'historique de ses salaires dans l'entreprise, chaque salaire étant affecté à une période de temps (marquée par la date de début de la période).
- Un salaire est composé d'un salaire brut, de charges patronales et de charges salariales.

Gestion des congés :

- Pour chaque employé, on mémorise chaque congé pris (posant qu'un congé concerne toujours une ou plusieurs journées entières, dont on conserve la date et la durée).
- Pour chaque employé, on mémorise le nombre total de congés cumulés par année.
- Chaque employé a le droit aux jours de congés suivants, par année :
 - o 25 jours (pour une quotité de 1) et 25 x quotité sinon
 - o Chaque fonction ouvre les droits à un certain nombre de jours de RTT
 - o Chaque service ouvre les droits à un certain nombre de jours de RTT
 - o Chaque tranche de 5 ans passés dans l'entreprise donne droit à un jour supplémentaire
 - o Les employés de plus de 40 ans ont un jour supplémentaire ; et ceux de plus de 50 ans deux

Questions

- 1) Donner une représentation Entité-Association des données pour la gestion des employés, des salaires et des congés et noter les cardinalités entre les entités.
- 2) Définissez un modèle relationnel en intention à partir de ce schéma, en introduisant un identifiant unique pour les employés.
- 3) Soulignez les clés primaires, et indiquez les clés étrangères (en indiquant les attributs et tables référencés)
- 4) Proposer une vue « *salairesFiscaux* » sur cette base (donnez sa définition en intention, et l'expression SQL correspondant à sa création) qui renvoie pour chaque employé *le salaire chargé* (brut + charges patronales), et *le salaire net* (brut – charges salariales) pour le salaire en cours (celui que touche actuellement l'employé).

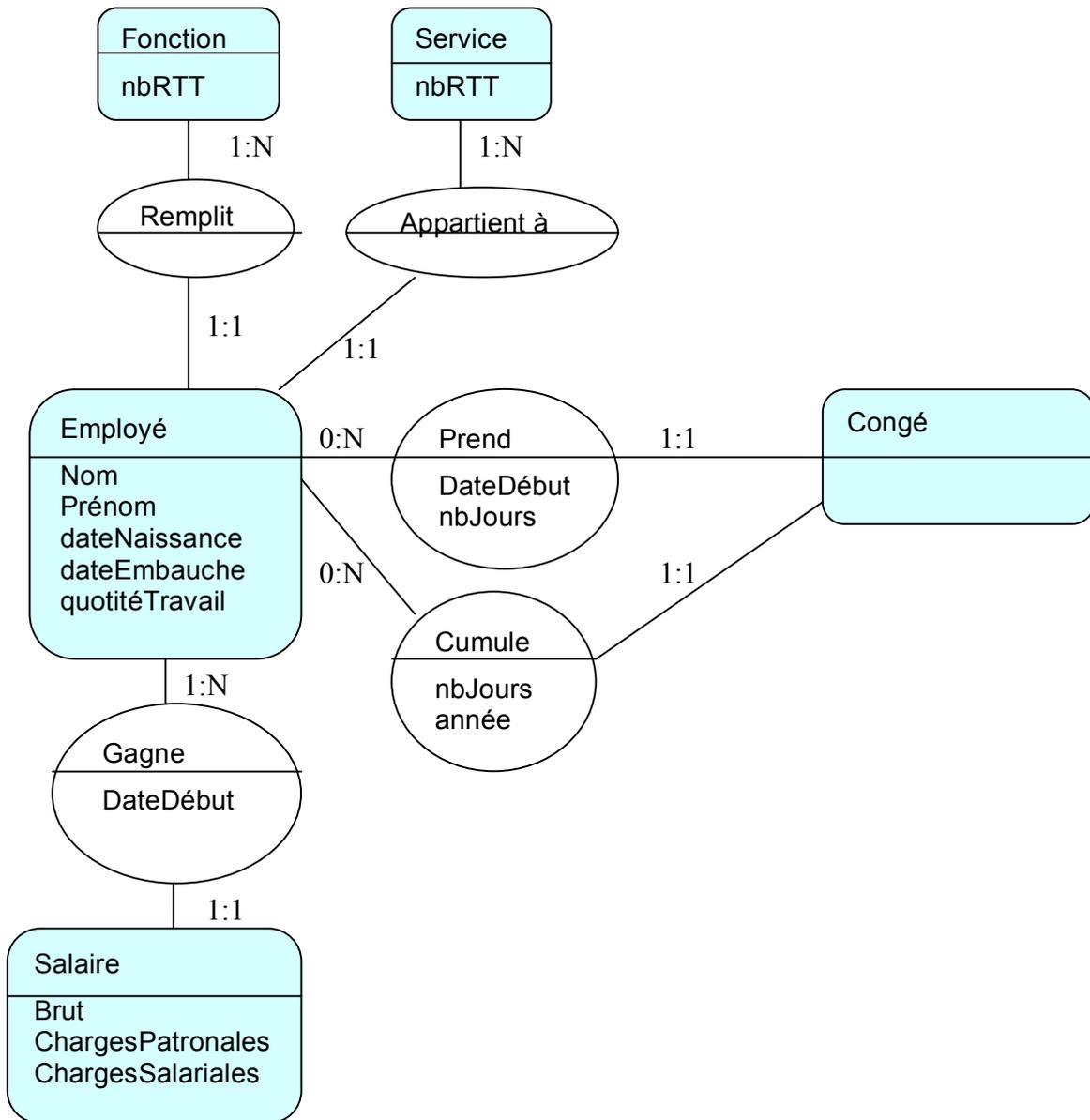
Questions (suite)

On garde dans la relation congésCumulés les droits aux congés cumulés par employé chaque année, et détaillés par catégories (ancienneté, age, quotité travaillée, fonction, service), suivant cette relation : CongésCumulés(emp_id, année, TotalNbJours, ancienneté_jours, age-Jours, quotité_jours, fonction_jours, service_jours)

- 5) Donnez la requête SQL correspondant à l'insertion pour l'année 2009 d'une entrée dans cette table pour l'employé 19177 embauché le 1/03/1981 et né le 16/09/1954, qui appartient au service 'reseaux' et remplit la fonction de 'manager', et qui travaille à mi-temps.

Réponses

1)



2) Modèle relationnel en intention :

Employé (emp_id, nom, prenom, dateNaissance, dateEmbauche, quotitéTravaillée, fonction, service)

Salaire (emp_id, dateDébut, brut, chargesPatronales, chargesSalariales)

Fonction(fonction, nbRTT)

Service(service, nbRTT)

CongésPris(emp_id, dateDébut, nbJours,)

CongésCumulés(emp_id, année, TotalNbJours, ancienneté_jours, age-Jours, quotité_jours, fonction_jours, service_jours)

3) **emp_id** de **Salaire** est une clé étrangère qui référence l'attribut **emp_id** de la relation **Employé**
fonction de **Employé** est une clé étrangère qui référence l'attribut **fonction** de la relation **Fonction**

service de **Employé** est une clé étrangère qui référence l'attribut **service** de la relation **Service**
Emp_id de **CongésPris** est une clé étrangère qui référence l'attribut **emp_id** de la relation **Employé**
Emp_id de **CongésCumulés** est une clé étrangère qui référence l'attribut **emp_id** de la relation **Employé**

5) salairesFiscaux(emp_id, salaireChargé, salaireNet)

```
CREATE VIEW salairesFiscaux AS
SELECT emp_id, brut+chargesPatronales AS charge, brut-chargesSalariales AS net
FROM salaire
WHERE date IN (select max(date) FROM salaire group by emp_id)
```

```
5) INSERT INTO CongésCumulés (emp_id, année, totalNbJours, ancienneté_jours, age-Jours,
quotité_jours, fonction_jours, service_jours) VALUES (
19117,
2009,
(2009-1981 div 5) + (2009-1954) div 40 + (2009-1954) div 50 + 25 * 0.5 + select nbRTT from fonction
where fonction='manager' + select nbRTT from service where service='reseaux' ,
(2009-1981 div 5),
(2009-1954) div 40 + (2009-1954) div 50,
25 * 0.5,
select nbRTT from fonction where fonction='manager',
select nbRTT from service where service='reseaux') ;
```

Algèbre relationnelle et requêtes SQL – 6pts

Une maîtresse de maison s'est constituée une base de données sur les personnes (probables) qu'elle invite et les plats qu'elle leur sert. Cette base de données est composée de trois relations :

- REPAS (date, invité) donne la liste des invités qui ont été reçus et à quelle date
- MENU (date, plat) donne le plat servi à chaque date
- PREFERENCE (personne, plat) donne pour chaque personne ses plats préférés.

N.B. : les attributs personne et invité ont le même domaine.

À l'aide du langage SQL, exprimer les requêtes suivantes :

Questions

- 1) Quels sont les invités du repas du 01/05/97 ?
- 2) Quels sont les plats qui ont été servis à Alice
- 3) Quels sont les invités qui lors d'un repas ont eu au moins un de leur plat préféré ?
- 4) Quelles sont les personnes qui n'ont jamais été invitées ?
- 5) Quels sont les invités venus à tous les repas ?

Réponses

- 1) SELECT invité FROM repas WHERE date=01/05/97 ;
- 2) SELECT plat FROM menu WHERE date IN (
SELECT date FROM repas WHERE invité='Alice') ;
- 3) SELECT invité FROM repas WHERE date IN (
SELECT date FROM menu WHERE plat IN (
SELECT plat FROM preference WHERE personne=invité));
- 4) SELECT personne FROM preference WHERE personne NOT IN (
SELECT invité FROM repas) ;
- 5) SELECT invité FROM REPAS GROUP BY invité HAVING COUNT(*) = (SELECT COUNT
(DISTINCT (date)) FROM REPAS)

Organisation Physique – 2pts

Dessinez le résultat des opérations suivantes sur l'organisation par hachage linéaire de la figure suivante (un dessin par opération) en supposant que la fonction de hachage est $h(\text{clé}) = \text{clé} \text{ MOD } 32$. Donnez la valeur de P à la fin de chaque insertion.

Début deuxième expansion : $d = 2$

Donc 2 digits de la clé servent à identifier le bloc.

$P = 0$

10100	11101 00001	11010	01111 11011
-------	----------------	-------	----------------

00	01	10	11
----	----	----	----

- a) insérer 22 ($22 = 16 + 4 + 2$)
 b) insérer 51 ($51 = 32 + 16 + 2 + 1$)

Réponses :

- a) $22 \text{ MOD } 32 = 22$
 $22_{10} = 10110_2$

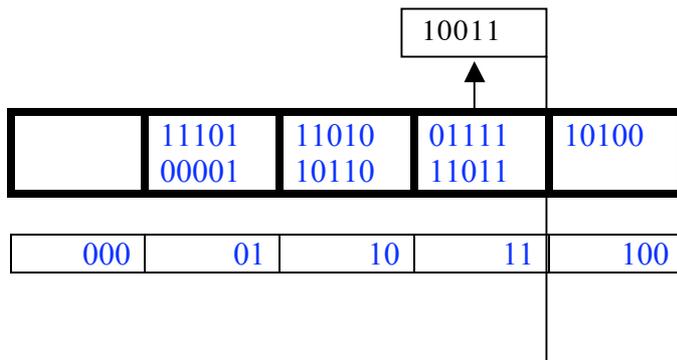
10100	11101 00001	11010 10110	01111 11011
-------	----------------	----------------	----------------

00	01	10	11
----	----	----	----

$P = 0$

- b) $51 \text{ MOD } 32 = 19$
 $19_{10} = 10011_2$

Il y a débordement, 00 est éclaté
 $P = 1$



Optimisation – 6pts

Soit le schéma relationnel :

Journaliste (JID, Nom, Prenom)

Journal (Titre, Redaction, ReID)

La table *Journaliste* stocke les informations (*Nom*, *Prenom*) sur les journalistes (*JID* est le numéro d'identification du journaliste). La table *Journal* stocke pour chaque rédaction d'un

journal le titre du journal (*Titre*), le nom de la rédaction (*Redaction*) et le *JID* de son rédacteur (*ReID*).

Soit la requête :

```
SELECT Nom
FROM Journal, Journaliste
WHERE Titre='Le Monde' and JID=ReID and Prenom='Jean'
```

1. (1 point) Que calcule cette requête?

2. (2 points) Voici deux requêtes algébriques (la jointure est notée *) :

(1) $\pi_{\text{Nom}} (\sigma_{\text{Titre} = \text{'Le Monde'} \wedge \text{Prenom} = \text{'Jean'}} (\text{JOURNALISTE} * (\text{JID} = \text{ReID}) \text{ JOURNAL}))$

(2) $\pi_{\text{Nom}} ((\sigma_{\text{Prenom} = \text{'Jean'}} (\text{JOURNALISTE}) * (\text{JID} = \text{ReID})) (\sigma_{\text{Titre} = \text{'Le Monde'}} \text{ JOURNAL}))$

a) Est-ce que les deux requêtes retournent le même résultat (sont équivalentes) et citer la/les règle(s) qui justifient la réponse ?

Est-ce qu'une requête est meilleure que l'autre si on considère les deux requêtes comme plans d'exécution? Expliquez votre réponse.

Il n'y a pas d'index sur le titre du journal.

On a seulement un index dense sur la table Journaliste sur l'attribut JID : I-Journaliste-JID.

3. (1 point) Donner le meilleur plan d'exécution physique sous forme arborescente. Les noeuds de l'arbre sont des opérateurs comme balayage séquentiel (BS), traversée d'index (TI), accès direct (par adresse) à une table (AD), tri, fusion, boucles imbriquées (BI), etc.

4. (1 point) Expliquez en détail ce plan.

Soit *Bal* le nombre de blocs de la table Journal et *Biste* le nombre de blocs de la table Journaliste. Soit *BTI* le nombre de blocs lus lorsqu'on traverse l'index I-Journaliste-JID.

5) (1 point) Quel est le coût en nombre de lectures de blocs pour exécuter ce plan?

Solution :

1) Les noms des rédacteurs du journal le Monde qui s'appellent Jean.

2) a) Oui, car la sélection est commutative avec la jointure – règle 5

2) b) Suivant les heuristiques vues en cours, la requête (2) qui exécute les sélections en premier avant les jointures a plus de chances (en limitant le nombre d'uplets de la jointure entre Journaliste et Journal) d'avoir un coût inférieur à (1)

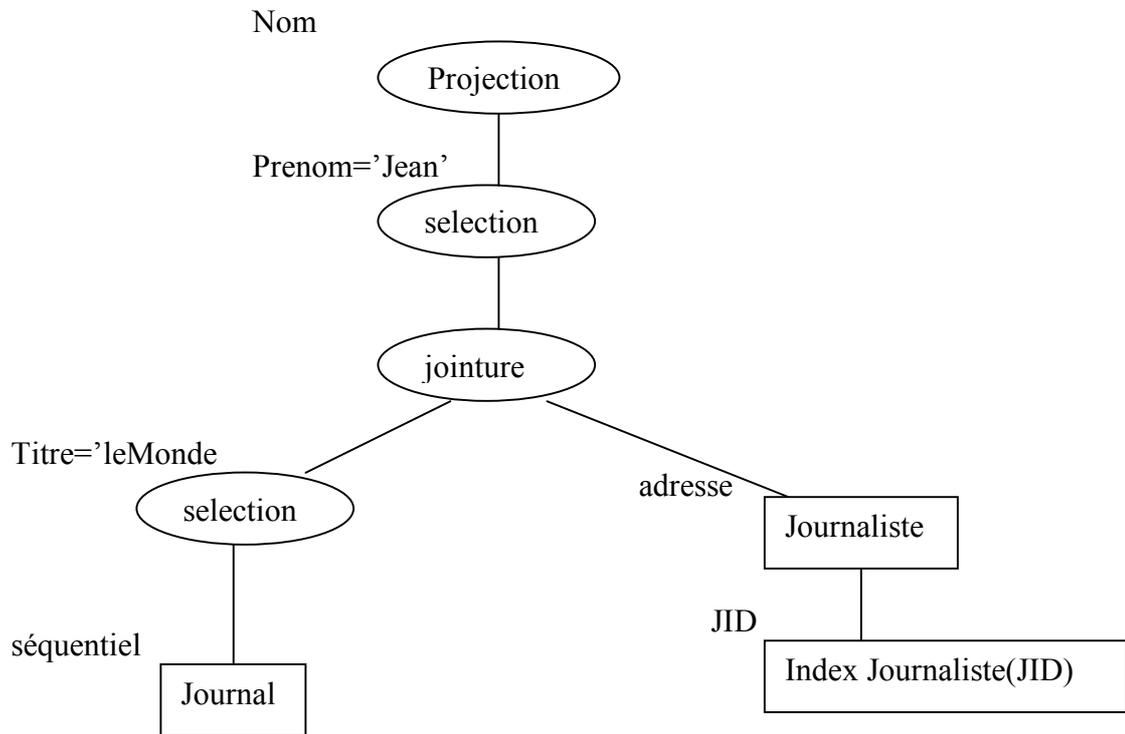
4) Réponse :

- par balayage séquentiel de la table Journal, on obtient les nuplets correspondant aux rédactions du Monde (sélection sur le titre).

- On fait ensuite une jointure par boucles imbriquées : la valeur de l'attribut ReID de ce nuplet sert de clé d'accès à l'index I-Journaliste-JID. La traversée de cet index donne les adresses des nuplets de la table Journaliste correspondant aux rédacteurs du Monde. Cette table est accédée directement (par adresse du nuplet correspondant au ReID du Monde).

- On projette le nuplet obtenu alors sur le champ Nom après avoir effectué une sélection sur le prénom.

3)



5) $Cout = Bal * \log(Biste).$