

employé(no, nom, prénom, adresse, #nodpt)
département(no, nom, #directeur)

+ Différents paramètres d'implantation

Exercise 1

- Evaluer le nombre de valeurs différentes
 - des attributs `no` et `nodpt` de la relation `employé`
 - des attributs `no` et `directeur` de la relation `département`
- Tout employé travaille dans un et un seul département et ne peut diriger qu'un seul département
- Un département a un et un seul directeur

$nv(\text{employé}, \text{no}) = ?$

$nv(\text{employé}, \text{nodpt}) = ?$

$nv(\text{département}, \text{no}) = ?$

$nv(\text{département}, \text{directeur}) = ?$

- $nv(\text{employé}, \text{no}) = 30000$ → clé primaire
- $nv(\text{employé}, \text{nodpt}) = 100$
 - chaque département a au moins un employé
 - son directeur, et donc tous les départements apparaissent dans la relation département
- $nv(\text{département}, \text{no}) = 100$ → clé primaire
- $nv(\text{département}, \text{directeur}) = 100$
 - chaque département a un directeur différent

Exercise 2

Evaluer le coût de production en nombre de transferts de pages des opérations suivants

```
sel(département, nom = "Informatique")
```

```
sel(employé, nodpt = "125")
```

dans le cas d'une sélection par boucle et d'une sélection indexée sur l'index I3 ?

```
join(employé, département, nodpt = no)
```

dans le cas :

- d'une jointure par double boucle
- d'une jointure par boucle indexée employé-index I2
- d'une jointure par boucle indexée département-index I3

- `sel(département, nom = "Informatique")`

coût = 5

→ balayer tous les blocs de la relation
département

`sel(employé, nodpt = "125")`

- Dans le cas d'une sélection par boucle
 - **coût = 3000**
 - balayer tous les blocs de la relation employé
- Dans le cas d'une sélection indexée sur l'index I3
- I3 est un index sur `nodpt`
 - $\text{coût} = \text{card}(\text{employé}) / \text{nv}(\text{employé}, \text{nodpt})$
 - **coût = 30000/100 (exercice 1) = 300**


```
join(employé, département, nodpt = no)
```

Dans le cas d'une jointure par double boucle

- coût = nb(département) + nb(employé)
- **coût = 5 + 3000 = 3005**
- **nb(département) = 5 donc < 10** (Nombre de cases de tampon)
- Les 5 blocs de la relation département peuvent être chargés en une seule fois

`join(employé, département, nodpt = no)`

- Jointure par boucle indexée sur I2
- Index sur clé primaire de `département` \rightarrow `no`

- $P = \text{Employé}$ $Q = \text{département}$
employé(département)

– $\text{coût} = \text{nb}(P) + \text{card}(P) * (\text{nb}(Q) / \text{nv}(Q, \text{no}))$
nv(Q, no) exo1

$$= \text{nb}(\text{employé}) + \text{card}(\text{employé}) * (\text{nb}(\text{département}) / \text{nv}(\text{département}, \text{no}))$$

$$= \mathbf{3000 + 30000 * (5/100)}$$

\rightarrow entier immédiatement supérieur

$$= \mathbf{3000 + 30000 * (1) = 33000}$$

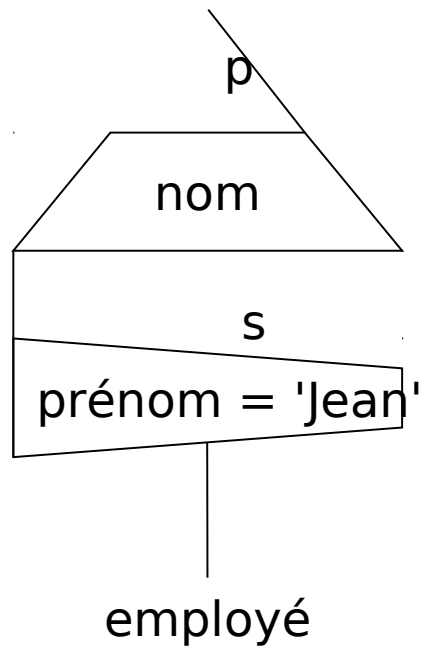
`join(employé, département, nodpt = no)`

- Jointure par boucle indexée sur I3
 - Index sur clé étrangère d'`employé` `nodpt`
 - $P = \text{département}$ $Q = \text{Employé}$
département(employé)
 - $\text{coût} = \text{nb}(P) + \text{card}(P) * (\text{nb}(Q) / \text{nv}(Q, \text{nodpt}))$
nv(Q,nodpt) exo1
- $= \text{nb}(\text{département}) + \text{card}(\text{département}) * (\text{nb}(\text{employé}) / \text{nv}(\text{employé}, \text{nodpt}))$
- $= 5 + 100 * (3000 / 100) = 3005$**

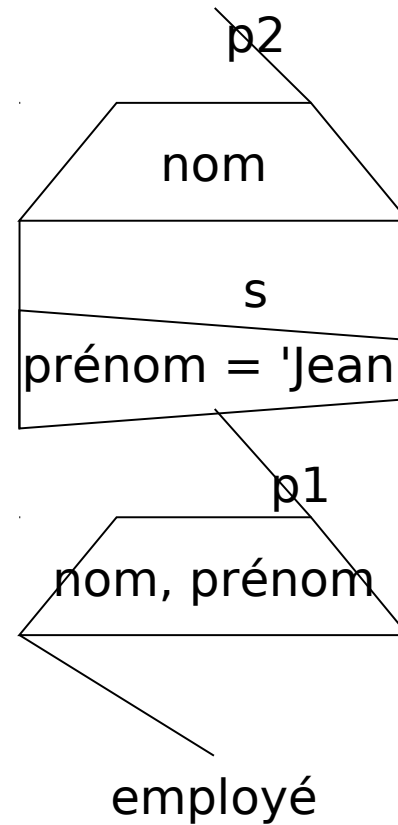
Exercise 3

- Construire tous les arbres de requêtes équivalents
 - en appliquant les transformations étudiées en cours
 - en supposant que les opérations algébriques sont la sélection, la projection et la jointure

```
select nom
From employé
where prénom = 'Jean'
```

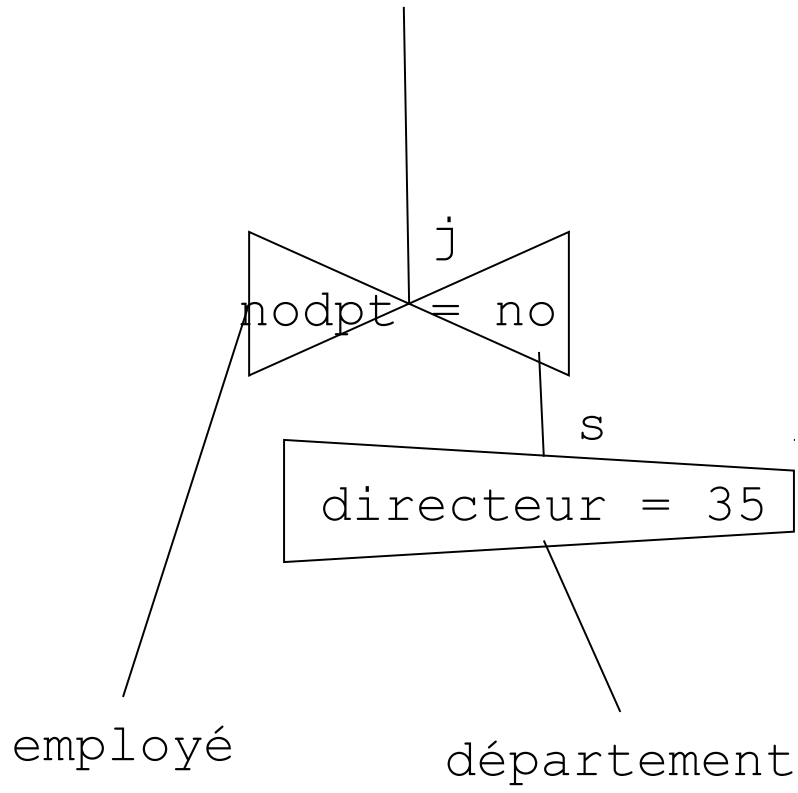


R1a

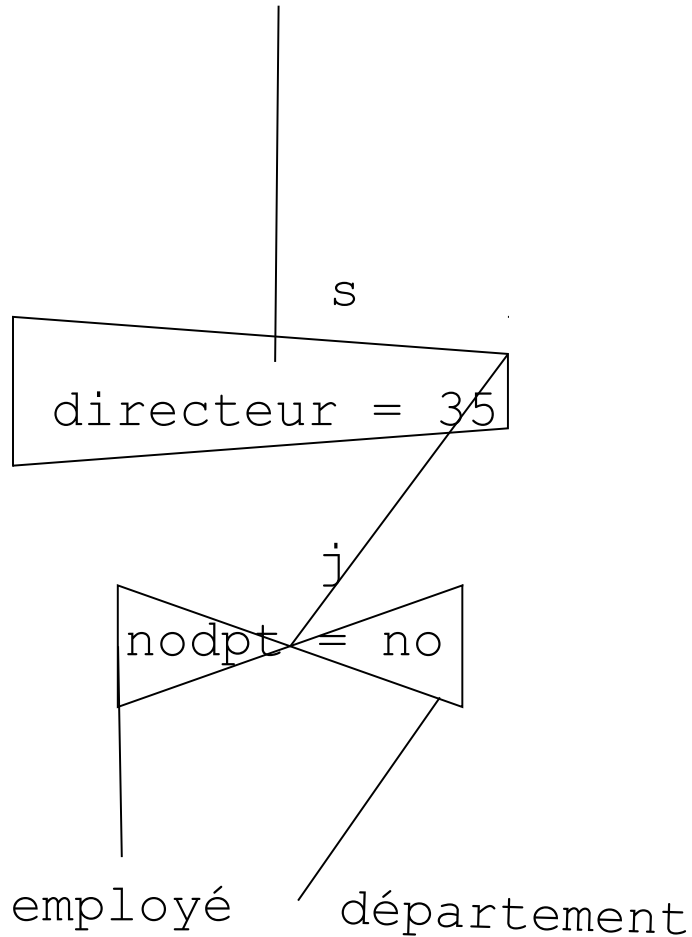


R1b

```
select *
from employé, département
where nodpt = no
  and directeur = 1965
```



R2a

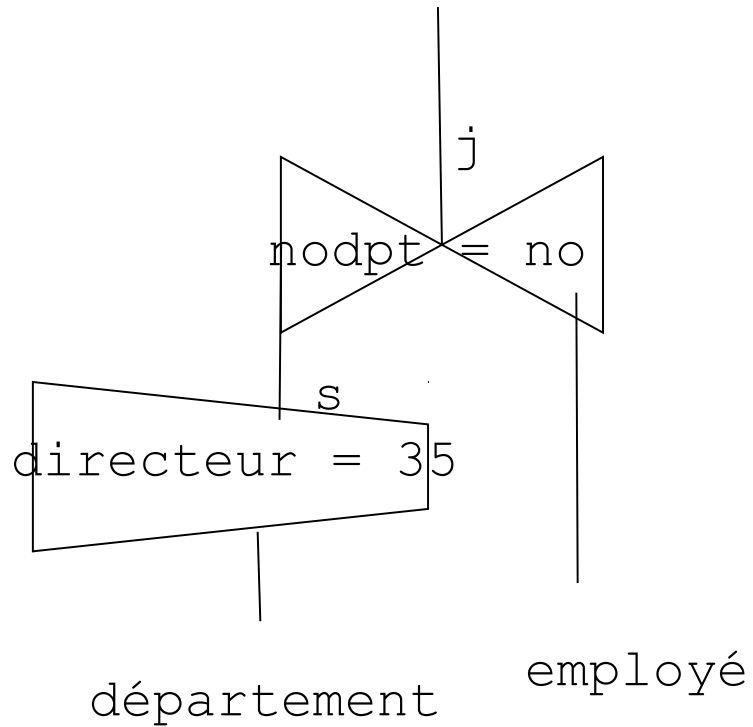


R2b

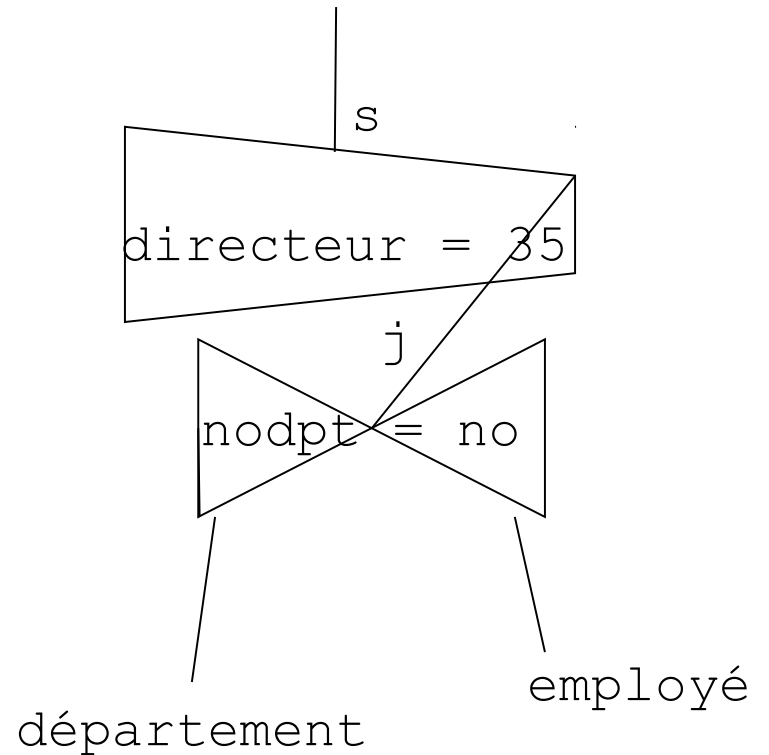
```

select *
from employé, département
where nodpt = no
and directeur = 1965

```



R2c



R2d

Exercise 4

Une BDR

```
inscription(nom_étudiant, code_ue)
ue(code, sigle_ufr)
responsable(nom_enseignant, code_ue)
ufr(sigle, nom)
```

Card(inscription) = 50 000

Card(ue) = 300

Card(enseignant) = 450

Card(ufr) = 5

Une requête

```
select      nom_étudiant
from        inscription i, ue, responsable r, ufr
where       i.code_ue = ue.code
and         ue.sigle_ufr = ufr.sigle
and        ue.code = r.code_ue
and         r.nom_enseignant = 'Dupont'
and         ufr.nom = 'Lettres'
```

```
inscription(nom_étudiant, code_ue)
ue(code, sigle_ufr)
responsable(nom_enseignant, code_ue)
ufr(sigle, nom)
```

Construire le « meilleur » arbre de requête

- Heuristique :
 - ne retenir que les arbres de jointure gauche (un arbre dont les opérandes droits des jointures ne contiennent pas de jointure) – support 329
 - puis placer les relations de base par ordre de cardinalité croissante en partant en bas à gauche et en arrivant en haut à droite
 - enfin, descendre les sélections aussi bas que possible

