

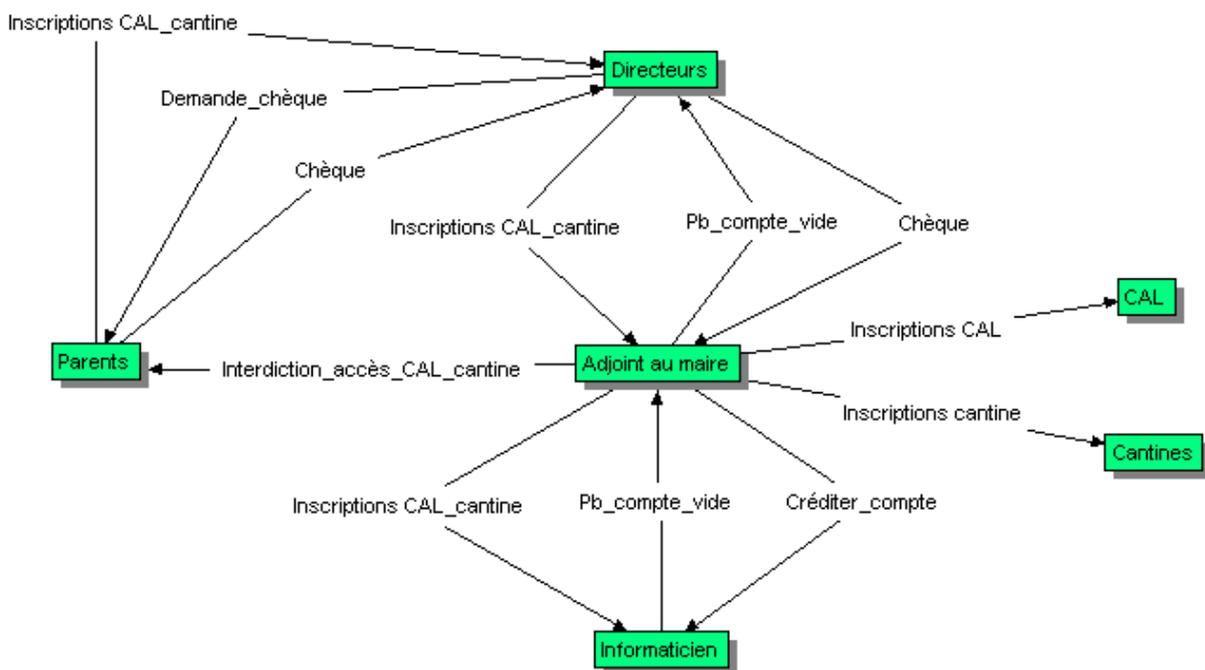
# Examen de CESI du 10 mars 2007

## - éléments de correction -

Gilles Simon

### 1. Diagramme de flux (2 points)

Ce diagramme fait apparaître les flux d'information décrits explicitement dans l'énoncé ainsi que deux flux non précisés dans l'énoncé entre l'adjoint au maire et les centres d'accueil, qui doivent bien connaître la liste des enfants qu'ils vont accueillir (mais je ne vous ai pas pénalisés si ces flux n'apparaissaient pas sur votre copie) .

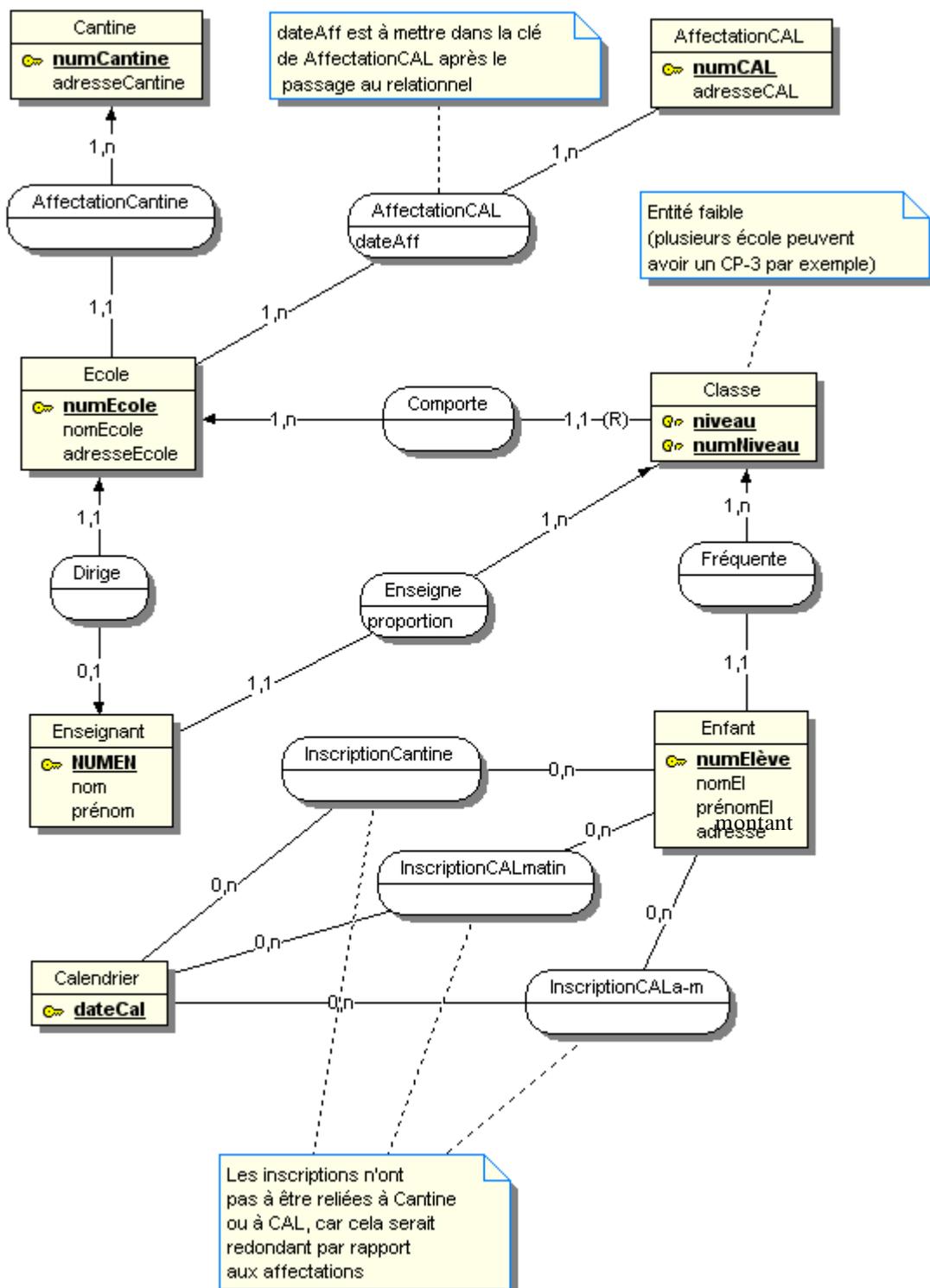


### 2. Schéma e/a (8 points)

Page suivante se trouve un schéma e/a possible, quelques remarques :

- L'énoncé disait « la clé de l'entité Classe devra comporter le niveau de la classe (CP, CE1, ...) ». Choisir comme clé de Classe l'attribut niveau seul serait une erreur puisque plusieurs classes de même niveau peuvent être prises en compte par le SI. Ajouter un numéro de classe à cette clé (noClasse, niveau) est aussi une erreur puisqu'on a alors une clé non minimale (niveau dépend de noClasse). Choisir noClasse (seul) comme clé ne répond pas à la contrainte de l'énoncé : il fallait donc définir Classe comme entité faible, identifiée par école (cf. schéma e/a page suivante). Comme plusieurs classes d'un même niveau peuvent coexister dans une école, on ajoute numNiveau à la clé de classe (par exemple, s'il y a 3 CP et un CE1, on a un CP-1, CP-2, CP-3, CE1-1), qui n'a rien à voir avec un numéro de classe. UN SEUL ETUDIANT SUR 80 a pensé à l'entité faible !!!
- Les affectations des CAL aux écoles sont tournantes, il faut donc bien mettre une cardinalité n-n entre Ecole et CAL, avec une date d'affectation dans l'association AffectationCAL.

- Remarque plus subtile (je vous ai très peu pénalisé pour cela) : pour modéliser les inscriptions des enfants en CAL et en cantine, la plupart d'entre vous ont associé Enfant à Cantine et à CAL en ajoutant la date d'inscription dans l'association. Cette solution introduit de la redondance dans la base de données puisqu'on sait déjà à quelle cantine et à quels CAL (suivant la date) est associée l'école (et donc tous les enfants de cette école). Pour prendre en compte les inscriptions, il suffit donc de connaître pour chaque enfant ses dates d'inscription en cantine ou en CAL, sans avoir à préciser de quelle cantine et de quel CAL il s'agit. Ceci peut se faire de la manière indiquée dans le schéma.



### 3. Schéma conceptuel des traitements (5 points)

La réalisation du SCT est normalement très simple si on suit les étapes vues en cours : diagramme de flux, graphes de précédence, SCT. Même sans faire le graphe de précédence, cette succession d'étapes implique un point important sur lequel j'ai beaucoup insisté en cours : TOUS les flux du diagramme de flux doivent apparaître comme événements dans le SCT, et il peut y en avoir d'autres en plus (par exemple, des événements de type temporel comme « Début de semaine » qui déclenche le ramassage des tableaux d'inscription par les directeurs d'école). AUCUN ETUDIANT n'a repris tous les flux du diagramme de flux dans son SCT !!! La plupart ont d'ailleurs mis des événements complètement déconnectés du diagramme de flux !

### 4. Niveau logique de la saisie des inscriptions (3 points)

Vous avez eu tendance à compliquer cette question, je voulais principalement la requête d'algèbre relationnel qui récupère le montant du compte d'un enfant ( $\pi_{\text{montant}}(\sigma_{\text{numElève}=\text{noEL}}(\text{Enfant}))$ ), décrémente ce montant en fonction des inscriptions et alerte en cas de montant négatif ou inférieur à -30.

### 5. Trigger Oracle (2 points)

Il fallait choisir un trigger de niveau ligne en AFTER UPDATE sur Enfant, puisque c'est le montant final (tenant compte de l'opération de débit en cours) qui est concerné et qu'on affiche :

```
CREATE OR REPLACE TRIGGER debit
AFTER UPDATE ON Enfant FOR EACH ROW
DECLARE
    new_montant NUMBER;
BEGIN
    SELECT montant INTO new_montant
    FROM Enfant
    WHERE numEleve = :new.numEleve
    IF (new_montant < 0) THEN
        DBMS_OUPUT.PUT_LINE(-20000, "Attention, compte débiteur")
    END IF;
END debit;
```

ou, encore plus simple :

```
CREATE OR REPLACE TRIGGER debit
AFTER UPDATE OF montant ON Enfant FOR EACH ROW
BEGIN
    IF (:new.montant < 0) THEN
        DBMS_OUPUT.PUT_LINE(-20000, "Attention, compte débiteur")
    END IF;
END debit;
```