Class

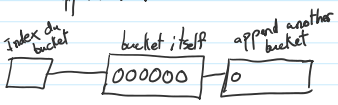
Wednesday, April 3, 2019 10:41 PM

Techniques d'index age:

Hash-based indexing (On veut un temps constant pour trouver un index.)

Use a hash function to determine where a data entry is stored

Hashing function $h : h(r) = \text{bucket dans lequel l'enregistrement } r \text{ appartient.}$

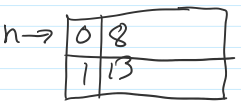


Name	Age	Salary
A	44	3000
B	40	6003
C	44	5004
D	25	3000
E	33	4003
F	29	2007
G	50	5004
H	22	6003

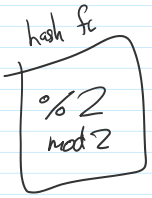
Linear Hashing

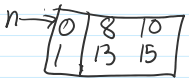
n = which bucket to split M = # of buckets = 2

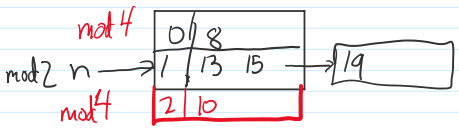
Split policy = Greater than 75% Bucket Capacity = 2



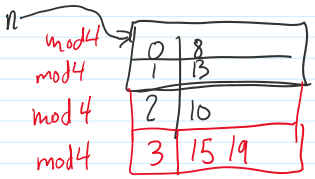
Add: 10, 15, 19, 22





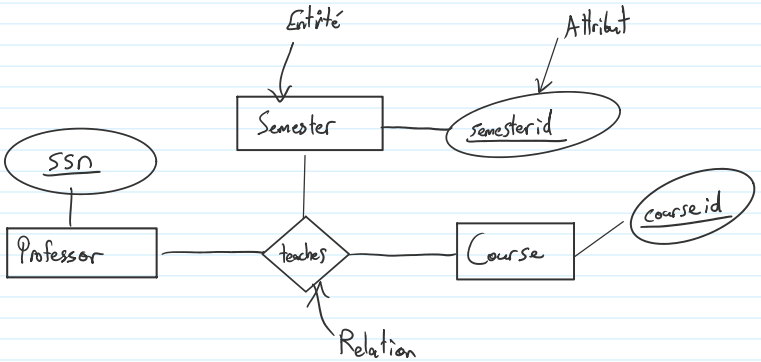


$5/6 \Rightarrow 75\% \rightarrow$ split



Final Review

EER Diagram (DGD 1 et 2)



clé primaire soulignée

flèche : upper bond

gras : lower bond

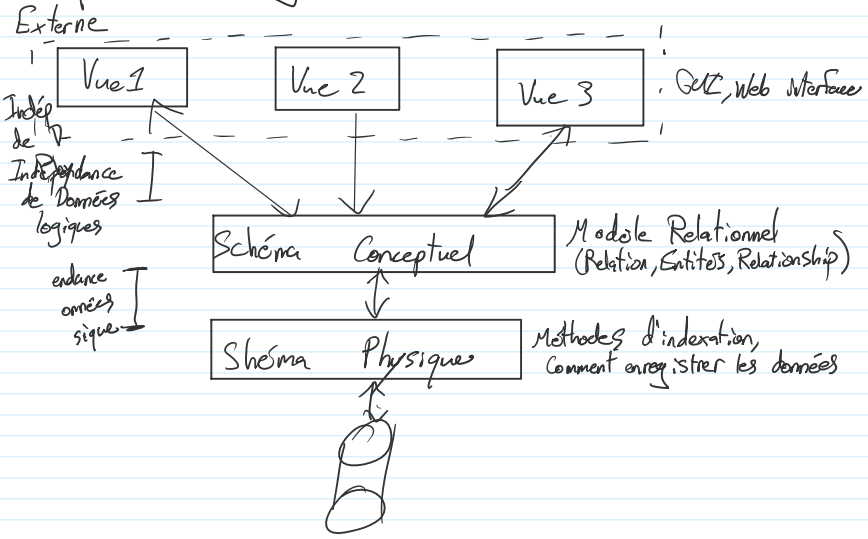
Algèbre relationnel (DGD 3)

Part C →

Part D → le domaine est restreint, le nombre de tuple résultant est limité

- Comprendre requêtes SQL (DGD 4), créer des requêtes SQL
  → Pouvoir créer des contraintes en SQL (primary key, foreign key, check, assertion)

- Comprendre diagramme de DBMS, c'est quoi des vues?



Formes Normales (DGD 6 et 7)

1NF — Les Dépendances Fonctionnelles (FDs)
2NF
3NF — (Normaliser une relation algo de DGD 6)
BCNF — ↳ minimal cover

Préserve les dépendances?
Lossless-join? (algorithme avec le tableau) (DGD 7)
→ Début DGD 8

~~Disque dur, Storage~~ (DGD 8)

Indexes

- Arbres B+ (Bulk loading algorithm)
  - ↳ Comment insérer, deleter des tuples
- Hashing
  - Statique
  - Extensible, Linéaire