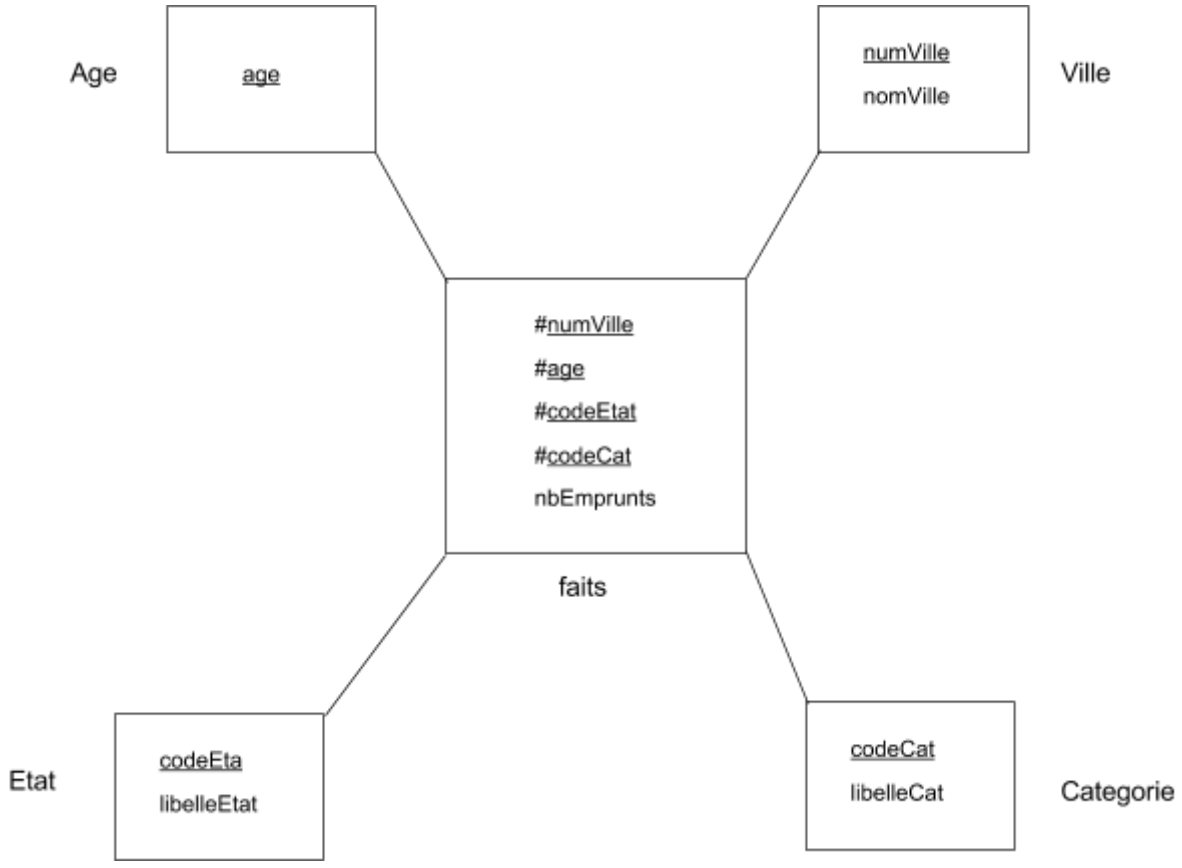
Johan Brunet
Chloé Dalger

Compte rendu TP Datawarehouse

-- Q1



Il s'agit d'un schéma en étoile.

-- Q2.a

```
select ville, age, etat, categorie, count(*) as nb_emp
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
group by ville, age, etat, categorie
order by nb_emp desc;
```

--Q2.b

```
select ville, age, etat, categorie, count(*) as nb_emp
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
```

```
and e.isbn = l.isbn
group by rollup (ville, age, etat, categorie)
order by nb_emp desc;
```

La clause GROUP BY ROLLUP permet de calculer des sous-totaux sur les agrégats.

--Q2.c

```
select ville, age, etat, categorie, count(*) as nb_emp
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
group by cube (ville, age, etat, categorie)
order by nb_emp desc;
```

La clause GROUP BY CUBE permet de montrer les agrégats pour toutes les combinaisons de valeurs pour les colonnes sélectionnées.

-- Q2.d

```
select ville, age, etat, categorie, count(*) as nb_emp
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
group by grouping sets (ville, age, etat, categorie)
order by nb_emp desc;
```

La clause GROUP BY GROUPING SETS génère le même type de résultats que la clause GROUP BY CUBE, mais on peut choisir les groupes de données à agréger.

-- Q2.e

```
select decode(ville, null, 'Toutes villes confondues', ville) as
ville, age, decode(etat, null, 'Tous etats confondus', etat) as
etat, categorie, count(*) as nb_emp
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
group by rollup (ville, age, etat, categorie)
order by nb_emp desc;
```

Johan Brunet
Chloé Dalger

--Q1

La densité du cube est :

Nombre de résultats du GROUP BY CUBE ÷ Nombre de cases du cube

Soit : *131 ÷ 2*3*10*5 = 131 ÷ 300 ≈ 0.4367*

--Q2.a

```
select decode(ville, null, 'Toutes villes confondues', ville) as
ville, age, decode(etat, null, 'Tous etats confondus', etat) as
etat, categorie, count(*) as nb_emp,
rank() over (order by age desc) as rank
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
group by rollup (ville, age, etat, categorie)
order by rank desc;
```

--Q2.b

```
select decode(ville, null, 'Toutes villes confondues', ville) as
ville, age, decode(etat, null, 'Tous etats confondus', etat) as
etat, categorie,
RATIO_TO_REPORT(count(*)) over(partition by ville) as
ratio_emprunt,
rank() over (order by age desc) as rank
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
group by rollup (ville, age, etat, categorie)
order by ville desc;
```

--Q2.d

```
select categorie, max(nb_emp)
from
    (select ville, age, etat, categorie, count(*) as nb_emp
    from abonne a, exemplaire e, livre l, emprunt em
    where a.num_ab = em.num_ab
```

Johan Brunet
Chloé Dalger

```
    and e.numero = em.num_ex
    and e.isbn = l.isbn
    group by (ville, age, etat, categorie)
    order by nb_emp desc)
group by categorie;
```

--Q3.a

```
select decode(ville, null, 'Toutes villes confondues', ville) as
ville, age, decode(etat, null, 'Tous etats confondus', etat) as
etat, categorie, count(*) as nb_emp,
rank() over (order by age desc) as ranky
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
and rownum < 3
group by (ville, age, etat, categorie)
order by ranky desc;
```

--Q3.b

```
select decode(ville, null, 'Toutes villes confondues', ville) as
ville, age, decode(etat, null, 'Tous etats confondus', etat) as
etat, categorie, count(*) as nb_emp,
rank() over (order by age desc) as ranky
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
and rownum < 3
group by (ville, age, etat, categorie)
order by ranky asc;
```

--Q3.c

```
select categorie, count(*) as nb_emprunt,
rank() over (order by count(*) desc) as ranky
from livre l, exemplaire e, emprunt em
where e.numero = em.num_ex
and e.isbn = l.isbn
group by categorie;
```

Johan Brunet
Chloé Dalger

--Q4.a

Les vues matérialisées, contrairement aux vues standard, dupliquent les données. Elles sont utilisées pour améliorer les performances d'accès aux données.

--Q4.b

```
create MATERIALIZED VIEW MV1
refresh start with sysdate next sysdate+1
enable query rewrite
as
select ville, age, etat, categorie, count(*) as nb_emp
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
group by ville, age, etat, categorie
order by nb_emp desc;

select * from MV1;
```

--Q4.c

```
select STALENESS from user_mviews;
```

--Q4.d

```
create MATERIALIZED VIEW MV1
refresh start with sysdate next sysdate+1 --mise à jour
enable query rewrite
as
select ville, age, etat, categorie, count(*) as nb_emp
from abonne a, exemplaire e, livre l, emprunt em
where a.num_ab = em.num_ab
and e.numero = em.num_ex
and e.isbn = l.isbn
group by ville, age, etat, categorie
order by nb_emp desc;
```

Johan Brunet
Chloé Dalger

--Q4.e

```
insert into EMPRUNT VALUES (922143, 1010, '15-03-16', '24-08-16',
'26-04-17', 1);
```

--Q4.f/g

```
select STALENESS from user_mviews;
commit;

begin
dbms_mview.refresh('MV1');
end;
```

--Q5.a
Ce sont des données qui changent peu au cours du temps et dont on ne peut pas prévoir les évolutions.

--Q5.b
Pour les données 'bibliothèque', une dimension à évolution lente (slowly changing dimention) est la dimension 'abonné'. En effet certains champs tels que l'adresse ou le numéro de téléphone peuvent changer, mais avec une faible périodicité.

--Q5.c
Quelques façons de gérer ce phénomène :
- écrasement des données : on remplace l'ancienne valeur par la nouvelle
- ajout d'un champ : on ajoute un champ pour stocker la nouvelle valeur tout en conservant l'ancienne
- ajout d'une table historique : on stocke l'ancienne valeur dans une table 'historique' et on la remplace dans la table concernée

Les biais possibles sur les données 'bibliothèque' :
- perte de données concernant un emprunt ou un abonné si l'on choisit la méthode de l'écrasement des données.
- une forte augmentation de la taille de la base de données si l'on choisit la méthode des tables d'historique.

--Q6.a
Un ETL est un outil informatique permettant de rassembler un vaste ensemble de données hétérogènes afin de

Johan Brunet
Chloé Dalger

--Q6.b
Les transformations que l'on peut faire grâce à un ETL sont :
- formatage des données
- nettoyage des données : homogénéiser les unités d'une dimension, repérer et corriger les erreurs de saisie (ex : entrer la rue au lieu du pays), ...
- contextualiser : regrouper dans une même dimension toutes informations qui la concerne et provenant de différentes sources (ex : toutes les informations d'un employé)

-- Q6.c
Outils propriétaires :
- Cognos Data Manager3
- Dell Boomi
- IBM InfoSphere DataStage4
- iWay Data Migrator
- Microsoft SQL Server Integration Services (SSIS)
- Oracle Data Integrator de son ancien nom Sunopsis
- SAP Data Services qui se nomme aussi Data Integration

Outils open-source :
- Talend
- Vanilla ETL