Motivations

- Pas facile de compter en algèbre relationelle!!!
- Impossible de faire des calculs statistiques (min, max, moyenne etc.)
- Ce type de calcul est très utile dans la vraie vie

$\Rightarrow$ SQL a introduit des fonctions pour faire cela

$\Rightarrow$ Les fonctions d'agrégat

Principe

- On utilise le langage de requête pour sélectionner l'information souhaitée
- Au lieu de retourner la liste des valeurs trouvées, on retourne un calcul statistique sur cette liste.

Exemple

- Combien y a-t-il de clients dans la base de données?

```
select *    /* donne la liste des clients */
from CLIENT;
```

Fonctions d'agrégat

Principe

- On utilise le langage de requête pour sélectionner l'information souhaitée
- Au lieu de retourner la liste des valeurs trouvée, on retourne un calcul statistique sur cette liste.

Exemple

- Combien y a-t-il de clients dans la base de données?

```
select count(*)   /* donne le nombre de clients */
from CLIENT;
```

Principe

- On utilise le langage de requête pour sélectionner l'information souhaitée
- Au lieu de retourner la liste des valeurs trouvée, on retourne un calcul statistique sur cette liste.

Exemple

- Quel est le prix moyen des produits?

```
select prixUnitaireProd  /* calcule la liste des prix */
from PRODUIT;
```

Principe

- On utilise le langage de requête pour sélectionner l'information souhaitée
- Au lieu de retourner la liste des valeurs trouvée, on retourne un calcul statistique sur cette liste.

Exemple

- Quel est le prix moyen des produit?

```
select avg(prixUnitaireProd) /* calcule la moyenne des prix */
from PRODUIT;
```

Les fontions d'agrégat

Les principales fonctions

- `count`: compte le nombre de lignes
- `min`: donne la valeur minimale
- `max`: donne la valeur maximale
- `sum`: fait la somme
- `avg`: fait la moyenne

Exemple

On veut le nom du produit le moins cher.

1. Le prix du produit le moins cher

```
select min(prixUnitaireProd)
from PRODUIT;
```

2. Retrouver le nom de ce produit

```
select nomProd
from PRODUIT
where prixUnitaireProd = (
      select min(prixUnitaireProd)
      from PRODUIT
);
```

- La sous-requête ne retourne qu'une seule valeur

$\Rightarrow$ on peut l'utiliser comme une valeur simple

Introduction

- Les fonctions d'agrégat calculent une seule valeur sur le résultat d'une requête
- Souvent insuffisant car on aimerait des statistiques du type
  - Pour chaque produit le nombre d'unités vendues

$\Rightarrow$ Introduction de la notion de regroupements

$\Rightarrow$ Presque toujours associée à une fonction d'agrégat

Exemple

Client	Nom	Prenom	Age	Ville
	Dupont	Jean	52	Orleans
	Duval	Paul	25	Orleans
	Martin	Martine	39	Tours
	Bon	Jean	41	Tours

Les villes où habitent les clients

Exemple

Client	Nom	Prenom	Age	Ville
	Dupont	Jean	52	Orleans
	Duval	Paul	25	Orleans
	Martin	Martine	39	Tours
	Bon	Jean	41	Tours

Les villes où habitent les clients

```
select ville
from client
group by ville;
```

Exemple

Client	Nom	Prenom	Age	Ville
	Dupont	Jean	52	Orleans
	Duval	Paul	25	Orleans
	Martin	Martine	39	Tours
	Bon	Jean	41	Tours

Les villes où habitent les clients

```
select ville
from client
group by ville;
```

Ville
Orleans
Tours

Exemple

Client	Nom	Prenom	Age	Ville
	Dupont	Jean	52	Orleans
	Duval	Paul	25	Orleans
	Martin	Martine	39	Tours
	Bon	Jean	41	Tours

Le nombre de clients dans chaque ville

Exemple

Client	Nom	Prenom	Age	Ville
	Dupont	Jean	52	Orleans
	Duval	Paul	25	Orleans
	Martin	Martine	39	Tours
	Bon	Jean	41	Tours

Le nombre de clients dans chaque ville

```
select ville, count(*) NbCli
from client
group by ville;
```

Exemple

Client	Nom	Prenom	Age	Ville
	Dupont	Jean	52	Orleans
	Duval	Paul	25	Orleans
	Martin	Martine	39	Tours
	Bon	Jean	41	Tours

Le nombre de clients dans chaque ville

```
select ville, count(*) NbCli
from client
group by ville;
```

Ville	NbCli
Orleans	2
Tours	2

Regroupements

Syntaxe

- C'est le mot clé: `group by <liste colonne>`
- `group by` va regrouper les lignes qui sont identiques sur le colonnes apparaissant dans la liste.
- Les colonnes du `select` doivent TOUTES apparaître dans le `group by` ou être des fonctions d'agrégat.

Test sur les groupes

- C'est le mot clé `having`
- **Exemple** liste des villes dont l'âge moyen des habitants est inférieur à 40

```
select Ville, avg(age)
from Client
group by Ville
having avg(age)<40;
```

Différence entre where et having

where agit avant
les regroupements

```
select Ville, avg(age)
from Client
where age<40
group by Ville;
```

donne pour chaque ville
la moyenne d'âge des personnes de moins de 40 ans

having agit après
les regroupements

```
select Ville, avg(age)
from Client
group by Ville
having avg(age)<40;
```

donne la liste des villes dont
l'âge moyen des habitants est inférieur à 40

Construction d'une requête avec agrégat

Requête

On veut la quantité totale d'objets achetée par chaque client

Instance

CLIENT

numCl	nom	pnom	adr
101	Aymar	Jean	Orléans
102	Lelivre	Julie	Tours
103	Rico	Léa	Orléans

CMD

numCl	refProd	laDate	qte
101	2501	12/02/2022	1
103	2501	15/02/2022	2
101	2714	12/02/2022	1
103	2853	17/02/2022	1

Construction d'une requête avec agrégat

Requête

On veut la quantité totale d'objets achetée par chaque client

```
select numcl,nom,pnom,qte
from CLIENT natural join CMD;
```

1. Ecrire la requête qui permet de retrouver les informations

Instance

CLIENT

numCl	nom	pnom	adr
101	Aymar	Jean	Orléans
102	Lelivre	Julie	Tours
103	Rico	Léa	Orléans

CMD

numCl	refProd	laDate	qte
101	2501	12/02/2022	1
103	2501	15/02/2022	2
101	2714	12/02/2022	1
103	2853	17/02/2022	1

résultat

numCl	nom	pnom	qte
101	Aymar	Jean	1
103	Rico	Léa	2
101	Aymar	Jean	1
103	Rico	Léa	1

Construction d'une requête avec agrégat

Requête

On veut la quantité totale d'objets achetée par chaque client

```
select numcl,nom,pnom,qte
from CLIENT natural join CMD
group by numcl,nom,pnom;
```

1. Ecrire la requête qui permet de retrouver les informations
2. Regrouper les lignes qui doivent l'être

Instance

CLIENT

numCl	nom	pnom	adr
101	Aymar	Jean	Orléans
102	Lelivre	Julie	Tours
103	Rico	Léa	Orléans

CMD

numCl	refProd	laDate	qte
101	2501	12/02/2022	1
103	2501	15/02/2022	2
101	2714	12/02/2022	1
103	2853	17/02/2022	1

résultat

numCl	nom	pnom	qte
101	Aymar	Jean	1
103	Rico	Léa	2
101	Aymar	Jean	1
103	Rico	Léa	1

Construction d'une requête avec agrégat

Requête

On veut la quantité totale d'objets achetée par chaque client

```
select numcl,nom,pnom,sum(qte)
from CLIENT natural join CMD
group by numcl,nom,pnom;
```

1. Ecrire la requête qui permet de retrouver les informations
2. Regrouper les lignes qui doivent l'être
3. Appliquer la fonction d'agrégat

Instance

CLIENT	numCl	nom	pnom	adr
	101	Aymar	Jean	Orléans
	102	Lelivre	Julie	Tours
	103	Rico	Léa	Orléans

CMD	numCl	refProd	laDate	qte
	101	2501	12/02/2022	1
	103	2501	15/02/2022	2
	101	2714	12/02/2022	1
	103	2853	17/02/2022	1

résultat	numCl	nom	pnom	sum(qte)
	101	Aymar	Jean	2
	103	Rico	Léa	3

Construction d'une requête avec agrégat

Requête

On veut la quantité totale d'objets achetée par chaque client

```
select numcl,nom,pnom,sum(qte)
from CLIENT natural join CMD
group by numcl,nom,pnom;
```

1. Ecrire la requête qui permet de retrouver les informations
2. Regrouper les lignes qui doivent l'être
3. Appliquer la fonction d'agrégat
4. Eventuellement appliquer les sélections sur le groupe (having)

Instance

CLIENT

numCl	nom	pnom	adr
101	Aymar	Jean	Orléans
102	Lelivre	Julie	Tours
103	Rico	Léa	Orléans

CMD

numCl	refProd	laDate	qte
101	2501	12/02/2022	1
103	2501	15/02/2022	2
101	2714	12/02/2022	1
103	2853	17/02/2022	1

résultat

numCl	nom	pnom	sum(qte)
101	Aymar	Jean	2
103	Rico	Léa	3

Construction d'une requête avec agrégat

Requête

On veut les jours où le total des quantités commandées est supérieur à 1

Instance

CLIENT

numCl	nom	pnom	adr
101	Aymar	Jean	Orléans
102	Lelivre	Julie	Tours
103	Rico	Léa	Orléans

CMD

numCl	refProd	laDate	qte
101	2501	12/02/2022	1
103	2501	15/02/2022	2
101	2714	12/02/2022	1
103	2853	17/02/2022	1

Construction d'une requête avec agrégat

Requête

On veut les jours où le total des quantités commandées est supérieur à 1

```
select laDate,qte
from CMD ;
```

1. Ecrire la requête qui permet de retrouver les informations

Instance

CLIENT

numCl	nom	pnom	adr
101	Aymar	Jean	Orléans
102	Lelivre	Julie	Tours
103	Rico	Léa	Orléans

CMD

numCl	refProd	laDate	qte
101	2501	12/02/2022	1
103	2501	15/02/2022	2
101	2714	12/02/2022	1
103	2853	17/02/2022	1

résultat

laDate	qte
12/02/2022	1
15/02/2022	2
12/02/2022	1
17/02/2022	1

Construction d'une requête avec agrégat

Requête

On veut les jours où le total des quantités commandées est supérieur à 1

```
select laDate,qte
from CMD
group by laDate ;
```

1. Ecrire la requête qui permet de retrouver les informations
2. Regrouper les lignes qui doivent l'être

Instance

CLIENT

numCl	nom	pnom	adr
101	Aymar	Jean	Orléans
102	Lelivre	Julie	Tours
103	Rico	Léa	Orléans

CMD

numCl	refProd	laDate	qte
101	2501	12/02/2022	1
103	2501	15/02/2022	2
101	2714	12/02/2022	1
103	2853	17/02/2022	1

résultat	laDate	qte
	12/02/2022	1
	15/02/2022	2
	12/02/2022	1
	17/02/2022	1

Construction d'une requête avec agrégat

Requête

On veut les jours où le total des quantités commandées est supérieur à 1

```
select laDate,sum(qte)
from CMD
group by laDate ;
```

1. Ecrire la requête qui permet de retrouver les informations
2. Regrouper les lignes qui doivent l'être
3. Appliquer la fonction d'agrégat

Instance

CLIENT

numCl	nom	pnom	adr
101	Aymar	Jean	Orléans
102	Lelivre	Julie	Tours
103	Rico	Léa	Orléans

CMD

numCl	refProd	laDate	qte
101	2501	12/02/2022	1
103	2501	15/02/2022	2
101	2714	12/02/2022	1
103	2853	17/02/2022	1

résultat	laDate	sum(qte)
	12/02/2022	2
	15/02/2022	2
	17/02/2022	1

Construction d'une requête avec agrégat

Requête

On veut les jours où le total des quantités commandées est supérieur à 1

```
select laDate,sum(qte)
from CMD
group by laDate
having sum(qte)>1 ;
```

1. Ecrire la requête qui permet de retrouver les informations
2. Regrouper les lignes qui doivent l'être
3. Appliquer la fonction d'agrégat
4. Eventuellement appliquer les sélections sur les groupes (having)

Instance

résultat	laDate	sum(qte)
	12/02/2022	2
	15/02/2022	2

Définition

- Une vue permet de donner un nom à une requête qui sera utilisée ensuite comme une table (principalement en consultation).
- A chaque appel de la vue la requête est ré-exécutée.
- Attention ce n'est pas une vraie table!

Syntaxe

- **Création:**
  ```
  create or replace view nomDeVue as requete
  ```
- **Destruction:**
  ```
  drop view nomDeVue
  ```
- **Exemple**:
  ```
  create or replace view LesPrenoms(prenom) as
  select distinct prenom from CLIENT;
  ```

Exemple

```
create or replace view AgeCli(numCli, nom, prenom, age) as
    select numCli, nomCli, prenomCli, (CURDATE()-dateNaisCli)/365
    from CLIENT;
```

Utilisation des vues

- Afficher l'âge des clients qui ont effectué une commande en 2023
- ```
  select nom, age
  from AgeCli natural join COMMANDE
  where YEAR(dateCom)=2023;
  ```
  ```

Utilité

- Construction de requêtes complexes
- Structuration de la requête

Exemple

Afficher l'âge des clients qui ont effectué une commande en 2023

```
with AgeCli(numCli, nom, prenom, age) as
  (select numCli, nomCli, prenomCli, (CURDATE()-dateNaisCli)/365
  from CLIENT)
select nom, age
  from AgeCli natural join COMMANDE
  where YEAR(dateCom)=2023;
```

Les vues temporaires

Syntaxe

`with nomDeVue as ( requeteVue ) requete`

Si besoin de plusieurs vues, on les sépare par des `,`

Exemple

Afficher l'âge des clients qui ont effectué une commande en 2023

```
with AgeCli(numCli, nom, prenom, age) as
  (select numCli, nomCli, prenomCli, (CURDATE()-dateNaisCli)/365
  from CLIENT),
  Commande2023 as (select numCli, nomCli from COMMANDE where YEAR(dateCom)=2023)
select nom, age
  from AgeCli natural join Commande2023;
```