

# データベース設計論

---

集約演算と入れ子問合せ

# 集約演算

---

# 集約演算

- WHERE節に該当する全てのタプルの属性にたいする集約演算を行うことができる

<code>sum(<i>expression</i>)</code>	合計値を計算する
<code>avg(<i>expression</i>)</code>	平均値を計算する
<code>count(<i>expression</i>)</code>	<b>nullでない</b> 行数を計算する
<code>count(*)</code>	行数を計算する
<code>max(<i>expression</i>)</code>	最大値を計算する
<code>min(<i>expression</i>)</code>	最小値を計算する

*expression* : 属性, もしくは属性を使った式

# 集約演算を含む問合せ例

- SHARP\_JPをフォローしている人の人数

```
SELECT count(*)  
FROM follow f
```

\*を指定した場合は  
タプル数を数える

```
WHERE f.followee_account='SHARP_JP'
```

- TANITAofficialの最新ツイート日時

```
SELECT max(datetime)
```

```
FROM tweet t
```

```
WHERE t.account = 'TANITAofficial'
```

# GROUP BY句をつかってグループ毎に集約演算

- GROUP BYを使って属性の値ごとにデータをグルーピングできる

例) アカウントごとにフォロワー数を表示する

```
SELECT followee_account, count(*)  
FROM follow f  
GROUP BY f.followee_account
```

集約演算count(\*)は  
group byで指定した属性  
の値が同じタプルの集合  
で演算を行う

Jin_Akanishi	3
RyoNishikido_JP	3
SHARP_JP	4
TANITAofficial	3
kingjim	3
okazaki_taiiku	2

# GROUP BY句を使うときの注意

- GROUP BY句を使った場合、SELECT節にはGROUP BYで指定した属性か集約演算しか書けない  
(ダメな例) アカウトごとに表示名とフォロワー数を表示する

```
SELECT u.name, count(*)  
FROM follow f, user u  
WHERE f.followee_account = u.account  
GROUP BY f.followee_account;
```

ただし、DBMSによってはエラーを出さずに、そのグループに該当する  
タプルの中から適当に一つの値を出力する場合もある

- SQLiteもエラーを出さないので注意

# 異なり数をカウントする

- 集約演算countで重複する値をカウントしないようにするには属性値の前に distinct をつける

アカウントごとにいいね  
の総数をカウント

```
SELECT t.account, count(*)  
FROM favorite f  
      tweet t  
WHERE f.tweet_id = t.id  
GROUP BY t.account;
```

account	count(*)
Jin_Akanishi	2
RyoNishikido	5
SHARP_JP	6
TANITAoffici	5
kingjim	1
massan	1
okazaki taii	1

アカウントごとにいいねを  
してくれた人数をカウント

```
SELECT t.account, count(distinct f.account)  
FROM favorite f  
      tweet t  
WHERE f.tweet_id = t.id  
GROUP BY t.account;
```

account	count(distinct f.account)
Jin_Akanishi	1
RyoNishikido	1
SHARP_JP	4
TANITAoffici	2
kingjim	1
massan	1
okazaki taiiku	1

# HAVING節を使って集約演算を含んだ検索条件を書く

- 集約演算を含んだ検索条件はWHERE節には書けない
  - HAVING節に書こう

2人以上にリツイートされているツイートのid

```
SELECT t.replied_id, count(*)  
FROM tweet t  
WHERE t.replied_id IS NOT NULL  
GROUP BY t.replied_id  
HAVING count(*) >= 2
```

replied_id	count(*)
7	2
17	2



# ORDER BY句を使って並べ替え

SHARP\_JPのツイートを古い順に並べる

```
SELECT t.datetime, t.content  
FROM tweet t  
WHERE t.account = 'SHARP_JP'  
ORDER BY datetime
```

アカウントをフォロー数の多い順に並べる

```
SELECT f.followee_account, count(*)  
FROM follow f  
GROUP BY f.followee_account  
ORDER BY count(*) desc
```

# 課題1

- アカウント毎にアカウント名と最新のツイート日時を表示する問合せをSQLで書こう

```
SELECT t.account, max(t.datetime)  
FROM tweet t  
GROUP BY t.account;
```

テーブルを一つしか使っていないので、テーブル名を前につける必要はない

```
SELECT account, max(datetime)  
FROM tweet  
GROUP BY account;
```

## 課題2

- これまでのツイートでいいねされた総数が3以上のアカウントの表示名を求める問合せをSQLで書こう

```
SELECT u.name, count(*)  
FROM favorite f, tweet t, user u  
WHERE f.tweet_id = t.id  
      and t.account = u.account  
GROUP BY u.account, u.name  
HAVING count(*)>=3
```