データベースシステム 第8回

理工学部情報科学科 松澤 智史

本日の内容

- ウインドウ関数
- ・副問合せ
- ・ストアドプロシージャ
- ・ストアドファンクション

SQLをファイルから実行

・構文 mysql> source SQLファイル名

```
mysql> source test.sql
Query OK, 0 rows affected (0.01 sec)
Query OK, 0 rows affected (0.00 sec)
mysql> _
```

データベースサンプル

https://dev.mysql.com/doc/index-other.html

- · Archives: the documentation archives
- . About: information about MySQL documentation and the MySQL documentation team

MySQL Server Doxygen Documentation

Title	HTML Online
MySQL Server (latest version)	View

Expert Guides

Language	Title	HTML Online	PDF
English	MySQL Internals	View	
English	MySQL Development Cycle	View	US Ltr A4

Example Databases

Title	Download DB	HTML Setup Guide	PDF Setup Guide		
employee data (large dataset, includes data and test/verification suite)	GitHub	View	US Ltr A4		
world database	Gzip Zip	View	US Ltr		
world_x database	TGZ Zip				
sakila database	TGZ Zip	View	US Ltr A4		
menagerie database	TGZ Zip				

| Tables_in_world | City | Country |

世界の都市のデータ(4000以上)等が含まれる

ウインドウ関数(分析関数)

- 現在の行に関するテーブル全体を舐める計算をする
- 集約関数と同じく集計動作を行う
 - ・ 集約関数は1行に集約するが、ウインドウ関数を使った場合は対象の行はそのまま残る
 - ・ 指定対象の行全てに対して処理が行われる(範囲指定は可能)
- ・ OVER句を用いて表される
- 構文

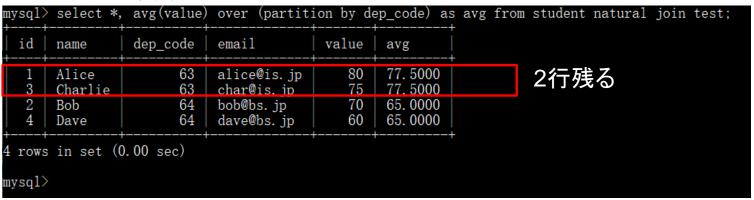
ウインドウ関数 over (partition by 列名) from テーブル名 →fromの前にover句を書くウインドウ関数 over (order by 列名) from テーブル名

- ※ ウインドウ関数でないavg等の集約関数も使える(overを使えばウインドウ関数になる)
- ・範囲対象に使用する列をPARTITION BY句, ORDER BY句で指定する
 - PARTITION BY句は集約関数のGROUP BY句とほぼ同じ動作をする

集約関数

mysq1>	> select	*, avg(valu	ue) as avg fro	om stude	nt natural	join test group by dep_code;
id	name	dep_code	email	value	avg	group by は
1 2	Alice Bob		alice@is.jp bob@bs.jp	80 70	77. 5000 65. 0000	一行に集約
2 rows		(0.01 sec)		,		T

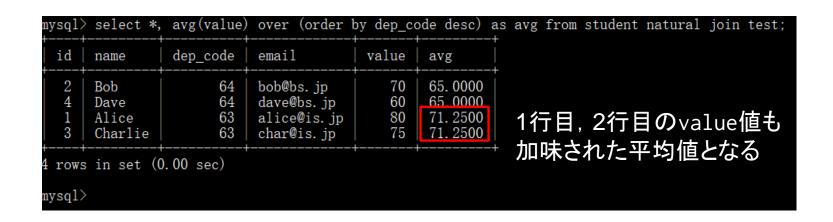
ウインドウ関数



mysql>	> select *,	avg(value)	over (order l	oy dep_co	ode desc)	as avg from student natural join test;
id	name	dep_code	email	value	avg	<u> </u>
2 4 1 3	Bob Dave Alice Charlie	64 64 63 63	bob@bs.jp dave@bs.jp alice@is.jp char@is.jp	70 60 80 75	65. 0000 65. 0000 71. 2500 71. 2500	77.5ではない?
4 rows	s in set (0	0.00 sec)				+

ウインドウ関数のorder by

- order by XXXは, XXXによって順番を並び替える ←ここまでは同じ
- ・ウインドウ関数のorder by句は 行を順番に並べた上で最初の行から現在行までのみを集計の対象にする



ウインドウ関数専用の関数

- ROW_NUMBER()
- RANK()
- DENSE_RANK()
- PERCENT_RANK()
- CUME_DIST()
- NTILE()
- LAG()
- LEAD()
- FIRST_VALUE()
- LAST_VALUE()
- NTH_VALUE()

ROW_NUMBER関数

• 1からNまでの現在行の数を返す

mysq1>	> select *,	row_number	() over (order	by dep	_code)	as RN	from stu	ıdent	natural	join	test;
id	name	dep_code	email	value	RN						
1 3 2 4	Alice Charlie Bob Dave	63 63 64 64	alice@is.jp char@is.jp bob@bs.jp dave@bs.jp	80 75 70 60	1 2 3 4						
4 rows	s in set (0). 00 sec)									

mysql>

ID	Name	CountryCode	District	Population	Ranking	
 1024	Mumbai (Bombay)	IND	Maharashtra	10500000	1	
2331	Seoul	KOR	Seoul	9981619	2	
206	São Paulo	BRA	São Paulo	9968485	3	
1890	Shangha i	CHN	Shangha i	9696300	4	
939	Jakarta	IDN	Jakarta Raya	9604900	5	
2822	Karachi	PAK	Sindh	9269265	6	
3357	Istanbul	TUR	Istanbul	8787958	7	
2515	Ciudad de México	MEX	Distrito Federal	8591309	8	
3580	Moscow	RUS	Moscow (City)	8389200	9	
3793	New York	USA	New York	8008278	10	
1532	Tokyo	JPN	Tokyo-to	7980230	11	
1891	Peking	CHN	Peking	7472000	12	
456	London	GBR	England	7285000	13	
1025	Delhi	IND	Delhi	7206704	14	
608	Cairo	EGY	Kairo	6789479	15	
1380	Teheran	IRN	Teheran	6758845	16	
2890	Lima	PER	Lima	6464693	i 17 l	
1892	Chongging	CHN	Chongging	6351600	18	
3320	Bangkok	THA	Bangkok	6320174	19	
2257	Santafé de Bogotá	COL	Santafé de Bogotá	6260862	20	

例:世界の都市の人口ランキング上位20都市

RANK関数

・現在行の順位を返す(同率の番号を飛ばす)

```
mysql> select *, rank() over (order by dep_code) as R from student natural join test;
                  dep code
                              email
                                             value
                                                     R
 id
       name
                                                80
75
70
60
       Alice
                              alice@is.jp
                        63
                                                     1
3
3
       Charlie
                        63
                              char@is.jp
       Bob
                        64
                              bob@bs.jp
       Dave
                        64
                              dave@bs.jp
 rows in set (0.00 sec)
mysql> _
```

DENSE_RANK関数

・現在行の順位を返す(同率の番号を飛ばさない)

```
mysql> select *, dense rank() over (order by dep code) as R from student natural join test;
                            email
 id
                 dep code
                                           value
                                                   R
      name
      Alice
                       63
                            alice@is.jp
                                              80
                       63
                                              75
      Charlie
                             char@is.jp
                                                   1
                             bob@bs.jp
                                              70
       Bob
                       64
                                              60
       Dave
                       64
                             dave@bs.jp
 rows in set (0.00 sec)
mysq1>
```

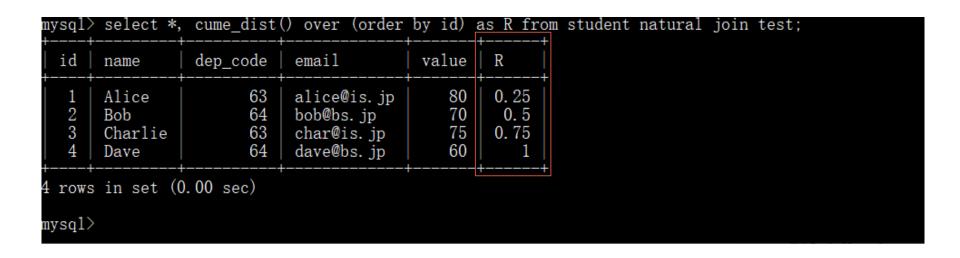
PERCENT_RANK関数

- 現在行の相対順位比率
- ・計算方法は(ランキング-1)/(全行数-1)

```
select *, percent rank() over (order by id) as R from student natural join test;
 id
                dep code
                           emai1
                                        value
                                                R
      name
      Alice
                      63
                           alice@is.jp
                                           80
                                            70
                      64
                           bob@bs.jp
                                                0. 33333333333333333
      Bob
   3
                      63
                                            75
                                                Charlie
                           char@is.jp
      Dave
                      64
                           dave@bs.jp
                                           60
 rows in set (0.00 sec)
mysql> _
```

CUME_DIST関数

- 現在行の相対順位比率
- ・計算方法は(ランキング/全行数)



NTILE関数

- ・ 引数で指定した数に分割する
- ・100行を5分割する場合は20行ずつ採番される

```
select *, ntile(2) over (order by id) as NT from student natural join test;
                             emai1
                 dep code
                                           value
                                                    NT
  id
       name
       Alice
                       63
                             alice@is.jp
                                              80
                                               70
       Bob
                       64
                             bob@bs.jp
                                               75
       Charlie
                       63
                             char@is.jp
                                              60
                       64
                             dave@bs.jp
       Dave
 rows in set (0.00 sec)
mysql> _
```

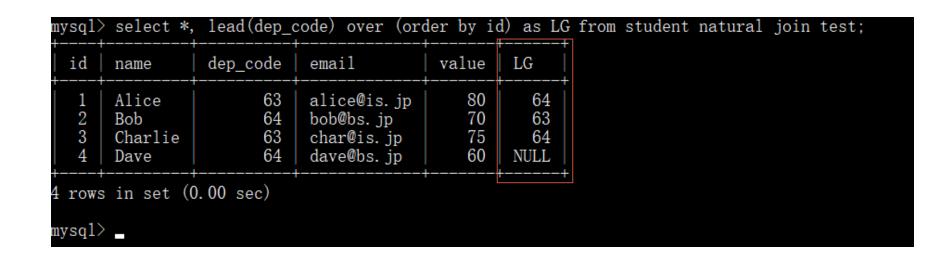
LAG関数

・前の行を返す

```
mysql> select *, lag(dep_code) over (order by id) as LG from student natural join test;
  id
                 dep_code
                             emai1
                                           value
                                                   LG
       name
                             alice@is.jp
      Alice
                       63
                                              80
                                                   NULL
       Bob
                       64
                             bob@bs.jp
                                              70
                                                      63
   3
                                              75
                       63
                                                      64
       Charlie
                             char@is.jp
                       64
                                              60
                                                      63
                             dave@bs.jp
       Dave
4 rows in set (0.00 sec)
mysq1> S
```

LEAD関数

・後の行(または後の行の1つ)を返す



FIRST_VALUE関数

・最初の行の値を返す

```
select *, first_value(name) over (order by id) as FV from student natural join test;
                 dep code
                            emai1
                                           value
                                                   FV
 id
      name
      Alice
                            alice@is.jp
                                              80
                                                   Alice
                       63
                       64
                                              70
                                                   Alice
      Bob
                            bob@bs.jp
  3
                       63
      Charlie
                                              75
                            char@is.jp
                                                   Alice
                       64
                            dave@bs.jp
                                              60
                                                   Alice
      Dave
 rows in set (0.00 sec)
mysql> _
```

LAST_VALUE関数

・最後の行の値を返す(ただしその行までの最後の値)

```
select *, last value(email) over (order by id) as LV from student natural join test;
mysql>
 id
                 dep code
                                           value
                                                   LV
                             email
      name
      Alice
                       63
                            alice@is.jp
                                              80
                                                   alice@is.jp
                       64
                                              70
      Bob
                            bob@bs.jp
                                                   bob@bs.jp
                                              75
                       63
      Charlie
                            char@is.jp
                                                   char@is.jp
                       64
                                              60
      Dave
                            dave@bs.jp
                                                   dave@bs.jp
 rows in set (0.00 sec)
mysql> _
```

NTH_VALUE関数

- 1から数えたN番目の行の値を返す
- Nが3の場合, 1~2行目はNULL

```
select *, nth_value(email, 3) over (order by id) as NV from student natural join test;
                 dep code
                            email
                                           value
                                                   NV
 id
      name
                            alice@is.jp
                                                   NULL
      Alice
                       63
                                              80
                            bob@bs.jp
      Bob
                       64
                                              70
                                                   NULL
                                              75
      Charlie
                       63
                            char@is.jp
                                                   char@is.jp
                       64
                                              60
                            dave@bs.jp
                                                   char@is.jp
      Dave
 rows in set (0.00 sec)
mysql> _
```

副問合せ

• SQL文の中に入れ子でSELECT文を入れることができる

()内のSELECT文を副問合せ、 冒頭のSELECT文を主問合せという

ストアドプロシージャ

・いくつものSQL文を1つにまとめ、CALLというコマンドで 実行できるようにしたもの

・構文 CREATE PROCEDURE プロシージャ名(引数) BEGIN

SQL文

END

ストアドプロシージャの例

```
mysql> delimiter //
{	t mysql}> create {	t procedure} {	t pl}()
    -> begin
   -> select * from student;
   -> end
   -> //
Query OK, 0 rows affected (0.01 sec)
mysql> delimiter ;
mysql> call pl;
                 dep_code
 id
                             email
       name
                        63
                             alice@is.jp
       Alice
                             bob@bs.jp
       Bob
                        64
                             char@is.jp
       Charlie
                        63
                        64
                             dave@bs.jp
       Dave
4 rows in set (0.00 sec)
Query OK, 0 rows affected (0.02 sec)
```

ストアドプロシージャの例2

```
mysql> delimiter //
mysql> create procedure p2(i integer)
   -> begin
   -> select * from student where id=i;
   -> end
   -> //
Query OK, 0 rows affected (0.01 sec)
mysql> delimiter ;
mysq1 > call p2(3);
                dep_code | email
  id
       name
   3
       Charlie
                       63
                            char@is.jp
 row in set (0.01 sec)
Query OK, 0 rows affected (0.02 sec)
```

ストアドプロシージャの利点

- ・作成されたときにプリコンパイルされるのでSQL文を一つずつ 呼ぶより高速に動作する
- ・プログラムから呼び出される場合などは、SQL文複数を送る よりプロシージャを一つ呼ぶ方が通信量削減に繋がる

ストアドプロシージャの一覧

```
SELECT
ROUTINE_SCHEMA, /* ストアドプロシージャがあるデータベース */
ROUTINE_NAME, /* ストアドプロシージャの名前 */
ROUTINE_TYPE /* プロシージャとファンクションのどちらかを示す */
FROM
information_schema.ROUTINES
WHERE
ROUTINE_TYPE = 'PROCEDURE'; /* プロシージャのみ抽出 */
```

※ファイルにしておくと便利

実行例

nysql> source sho	w_procedure.sql;	
ROUTINE_SCHEMA	ROUTINE_NAME	ROUTINE_TYPE
sys	create_synonym_db	PROCEDURE
sys	execute_prepared_stmt	PROCEDURE
sys	diagnostics	PROCEDURE
sys	ps_statement_avg_latency_histogram	PROCEDURE
sys	ps_trace_statement_digest	PROCEDURE
sys	ps_trace_thread	PROCEDURE
sys	ps_setup_disable_background_threads	PROCEDURE
sys	ps_setup_disable_consumer	PROCEDURE
sys	ps_setup_disable_instrument	PROCEDURE
sys	ps_setup_disable_thread	PROCEDURE
sys	ps_setup_enable_background_threads	PROCEDURE
sys	ps_setup_enable_consumer	PROCEDURE
sys	ps_setup_enable_instrument	PROCEDURE
sys	ps_setup_enable_thread	PROCEDURE
sys	ps_setup_reload_saved	PROCEDURE
sys	ps_setup_reset_to_default	PROCEDURE
sys	ps_setup_save	PROCEDURE
sys	ps_setup_show_disabled	PROCEDURE
sys	ps_setup_show_disabled_consumers	PROCEDURE
sys	ps_setup_show_disabled_instruments	PROCEDURE
sys	ps_setup_show_enabled	PROCEDURE
sys	ps_setup_show_enabled_consumers	PROCEDURE
sys	ps_setup_show_enabled_instruments	PROCEDURE
sys	ps_truncate_all_tables	PROCEDURE
sys	statement_performance_analyzer	PROCEDURE
sys	table_exists	PROCEDURE
sakila	rewards_report	PROCEDURE
sakila	film_in_stock	PROCEDURE
sakila	film_not_in_stock	PROCEDURE
test_db	pr3	PROCEDURE
test_db	pr1	PROCEDURE

ストアドファンクション

- 基本的にストアドプロシージャと同じであるが 関数であるため、返り値が必要
- ・構文
 CREATE FUNCTION ファンクション名(引数) RETURNS 型
 DETERMINISTIC
 BEGIN
 文
 END
- DETERMINISTICは返り値が決定的(入力が同じ場合は出力が同じ)の場合に指定する
 おき場合はNOT DETERMINISTICを指定する
 - ※違う場合はNOT DETERMINISTICを指定する

ストアドファンクションの例

```
|mysql> delimiter //
mysql> create function f(i integer) returns integer deterministic
   -> begin
   -> return (i+i);
   -> end
   -> //
Query OK, 0 rows affected (0.01 sec)
mysql> delimiter ;
mysql> select f(10);
  f(10)
     20
 row in set (0.00 sec)
mysq1>
```

まとめ

- ウインドウ関数を用いるとグループ化せずに 様々な集計が可能になる
- ・SQL文(SELECT)を入れ子にして記述する問合せを 副問合せと呼ぶ
- ・SQLを用いた一連の処理をまとめたものを ストアドプロシージャと呼びCALLで実行できる
- ユーザ定義の関数(ストアドファンクション)を 定義することができる

質問あればどうぞ