■ 集合演算子(UNION), 集計とグループ化(GROUP BY) ■

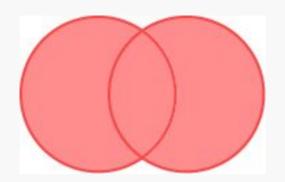
1日目	2日目	3日目	4日目	5日目	6日目	7日目
8日目	9日目	10日目	11日目	12日目	13日目	14日目
15日目	16日目	17日目	18日目	19日目	20日目	21日目

集合演算子 (UNION, UNION ALL, EXCEPT, INTERSECT)

集合演算子とは

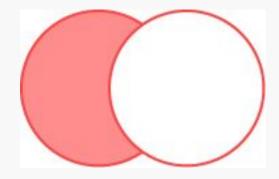
構造のよく似た複数のテーブルに対してSELECTでレコードを取得して、取得結果を 組み合わせるSQL

UNION



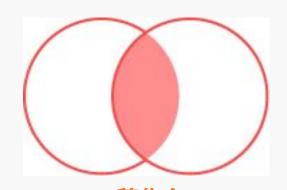
和集合 複数の検索結果を足し 合わせる

EXCEPT*)
(MINUS)



差集合 検索結果をのうち、重 複するものを取り除く

INTERSECT*)

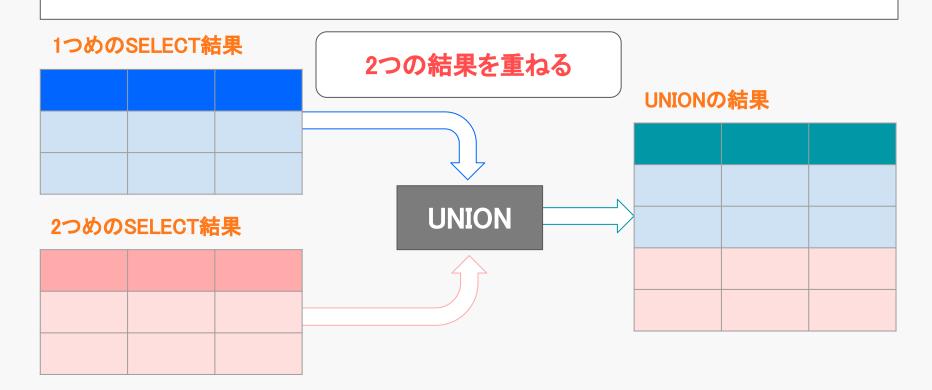


積集合 検索結果で、重複する ものを取り出す

*) MySQLでは、利用できない

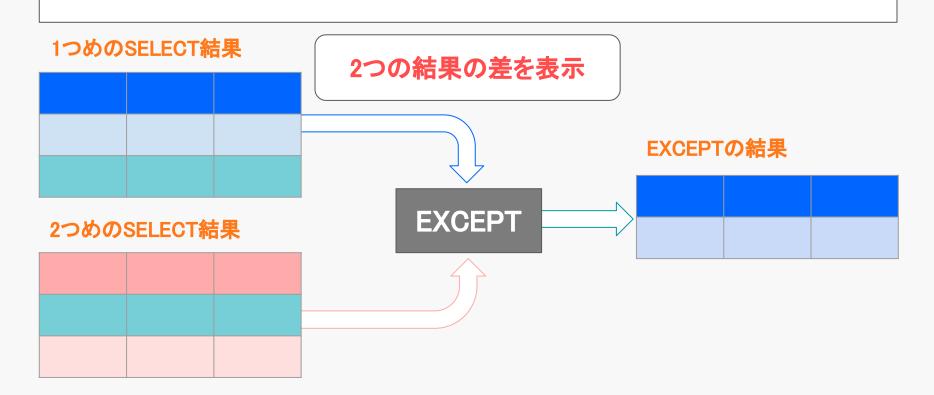
UNION, UNION ALL

検索結果の、和集合を求めるSQL文 UNIONは、重複する行は1つにまとめ、UNION ALLは、重複する行は重複したまま取り出す



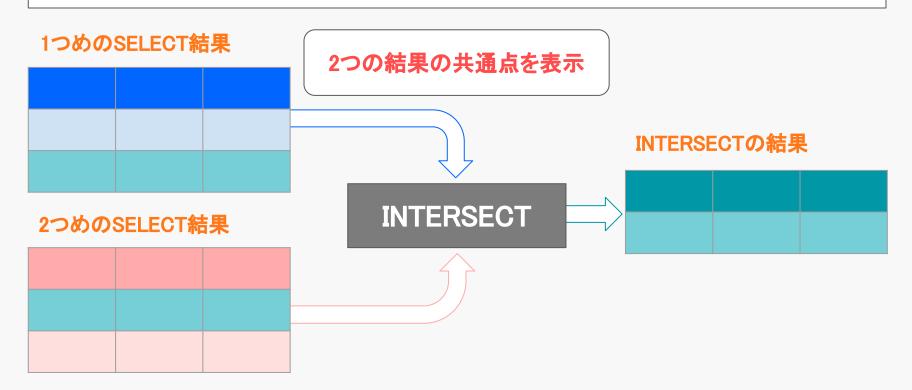
EXCEPT (MINUS)

ある集合と別の集合の和を求めるSQL文 SQL1とSQL2の結果を比較して、SQL1の結果のうちSQL2の結果に存在するものを差し引く



INTERSECT

ある集合と別の集合の積集合を求めるSQL文 SQL1とSQL2の結果を比較して、2つの結果に共通する行を表示する



UNION, UNION ALL, EXCEPT, INTERSECTの書き方

```
# UNIONのSQL文
# (UNIONをUNION ALL, EXCEPT, INTERSECTにすると同じように実行できる)
SELECT * FROM table1
UNION
SELECT * FROM table2:
#複数の検索結果をつなぐ
SELECT * FROM table1
UNION
SELECT * FROM table2
UNION
SELECT * FROM table3:
```

集合演算子を使う際の注意点

- 各SQLの取得するカラム数を合わせること
- •ORDER BYを利用する場合は、1つ目のSQLのカラム名を用いること

集計関数 (SUM, COUNT, MIN, MAX, AVG)

データを集計する

DBを利用する上で、データを集計して分析することが重要になる。

集計関数を用いると、特定の列の合計値、最大値、最小値などを求めることができる



number	
1	
2	
:	





代表的な集計関数一覧

MySQLでは、以下のような集計関数が用いられます。

詳細) https://dev.mysql.com/doc/refman/5.6/ja/group-by-functions.html

関数名	説明	
SUM	各行の値の合計値を計算する	
MAX	各行の値の最大値を計算する	
MIN	各行の値の最小値を計算する	
AVG	各行の値の平均値を計算する	
COUNT	各行の値の数をカウントする	

集計関数での注意点(型により集計が異なる)

集計する型に応じて結果は異なるため、どの型で集計したのか注意する必要がある

関数名	数値型	文字列型	日付型
SUM	合計値	×(実行できない)	×
MIN	最小値	並び替えて最初の文字列	最も古い日付
MAX	最大値	並び替えて最後の文字列	最も新しい日付
AVG	平均值	×	×
COUNT	行数	行数	行数

集計関数での注意点(NULLに対する集計)

集計関数も、NULLに対しては注意する必要がある

関数名		集計時のNULLの扱い	全行がNULLの場合の集計結果
SUM		無視される	NULL
MIN			
MAX			
AVG			
COUNT*)	列名指定	無視される	0
COUNTY	* 指定	NULLを含んでカウントする	全行数

^{*)} countには、SELECT COUNT(カラム名), SELECT COUNT(*)の2通りの記述方法がある

集計関数の書き方

SUMで合計値を計算する

SELECT SUM(number) FROM table_name

AVGで平均値を計算する

SELECT AVG(number) FROM table_name

MIN, MAXで最小値、最大値を計算する

SELECT MIN(number), MAX(number) FROM table_name

COUNTで行数を計算する

SELECT COUNT(*) FROM table_name

グループにわける (GROUP BY)

データをグループ分けして、グループ毎に集計する

GROUP BYを用いると、要素ごとにグループ分けして集計できる

部署	給料
営業	10000
経理	20000
営業	20000
開発	30000
開発	40000
経理	30000



部署	給料
経理	20000
経理	30000

部署	給料
開発	30000
開発	40000

AVGで部署毎に平均値を 求める

部署	給料
営業	15000
経理	25000
開発	35000

GROUP BYの書き方

SELECT column1, SUM(column2) FROM table GROUP BY column1

GROUP BY column1

FROM table

column1	column2
A	
В	
С	



column1	column2
Α	

column1	column2
В	

column1	column2
С	

SELECT column1, SUM(column2)

column1	SUM (column2)
A	
В	
С	

GROUP BYの書き方

column1でグループ化して集計する

SELECT column1, SUM(column2), AVG(column2) FROM table_name GROUP BY column1

column1とcolumn2でグループ化して集計する

SELECT column1, column2, SUM(column3), AVG(column3) FROM table_name GROUP BY column1, column2

#WHEREで絞り込んでから、グループ化する

SELECT column1, MIN(number) FROM table_name WHERE column1<"OO" GROUP BY column1

#集計結果をORDER BYで並び替える

SELECT column1, COUNT(*) FROM table_name GROUP BY column1 ORDER BY COUNT(*)

GROUP BYでCASE文を利用する①

```
# GROUP BY内にCASE文を記述する
SELECT CASE
  WHEN name IN ("香川県", "高知県", "愛媛県", "徳島県") THEN "四国"
  ELSE "その他"
 END AS "district".
  count(*)
FROM prefectures
GROUP BY — GROUP BYの中にCASEを記述
CASE
  WHEN name IN ("香川県", "高知県", "愛媛県", "徳島県") THEN "四国"
  ELSE "その他"
END
```

GROUP BYでCASE文を利用する②

```
# SELECTのカラム内にCASE文を記述する
SELECT
  age,
  CASE
  WHEN name < 20 THEN "未成年"
  ELSE "成人"
  END AS "分類",
  count(*)
FROM users
GROUP BY
name
```

HAVING

HAVINGとは

グループ化して集計した結果に対して、絞込みをする場合に用いるSQL

SELECT 部署, AVG(給料) GROUP BY 部署



HAVING AVG(給料) > 20000



部署	AVG(給料)
営業	15000
経理	25000
開発	35000

HAVINGの書き方

SUM(column2) が10000よりも大きい場合

SELECT column1, SUM(column2), AVG(column2) FROM table_name GROUP BY column1 HAVING SUM(column2) > 10000

column1とcolumn2でグループ化して、AVG(column1)の値で絞込み集計する

SELECT column1, column2, SUM(column3), AVG(column3) FROM table_name GROUP BY column1, column2 HAVING AVG(column1) < 10000

#WHEREで絞り込んでから、グループ化して、HAVINGで絞り込む

SELECT column1, MIN(number) FROM table_name WHERE column1<"OO" GROUP BY column1 HAVING MIN(number) < 10000

#集計結果をORDER BYで並び替える

SELECT column1, COUNT(*) FROM table_name GROUP BY column1 HAVING COUNT(*) < 100 ORDER BY COUNT(*)

GROUP BYを使わない場合のHAVINGの書き方

HAVINGはGROUP BYとセットで利用する印象があるが(古いSQLではGROUP BYとセットでないと利用できないが)、実際はHAVINGだけで利用することもできる

HAVINGでは、集計結果の比較に用いる