2章　検索の基本

　　２－１ SELECT文の基本

SELECT文・・・テーブルからデータを取り出すときに使う文

　　　　テーブルのデータから必要なものをセレクトする、ということ

　　　SELECT文で必要なデータを取り出すことを問い合わせ（クエリ）という。



【使い方】

SELECT　＜商品名＞、＜商品分類＞＜仕入単価＞

FROM ＜テーブル名＞；

　　　　　↓

|  |  |  |
| --- | --- | --- |
| 列名1 | 列名2 | 列名3 |
| Tシャツ | 衣類 | 500 |
| 穴あけパンチ | 事務用品 | 320 |

となる。

※　SELECT＜列名＞←列名を\*にするとすべての列名という意味になる。

SELECT　＊　FROM　＜データベース名＞＜テーブル名＞でも表示できる。

列に別名をつける方法

　AS　＋　キーワード・・・そのものに別名をつけることができる

例

Shohin\_id AS “商品ID ”　キーワードは日本語でも打てるが””が必要。

FROM shohin;

定数をつける

　　文字列定数や数値定数、日付定数があり、それをSELECT文ではつけることができる。

※この時は””ではなく’’をつかう

重複行を省くやり方

　　SELECT　DISTINCTを使うことで重複する行は短縮できる

SELECT　DISTINCT　shohin\_bunrui FROM shohin;

|  |
| --- |
| 商品分類 |
| 衣類 |
| 事務用品 |
| 衣類 |
| キッチン用品 |
| キッチン用品 |

|  |
| --- |
| 商品分類 |
| 事務用品 |
| 衣類 |
| キッチン用品 |

と省略される。

WHEREによる行の選択

WHERE句・・・選択したい行の条件をWHERE句で表現できる。

例

　SELECT　＜列名＞、 これで＜テーブル名＞の中の＜列名＞の中の’条件’

　FROM　＜テーブル名＞ 　　だけが表示される。

WHERE　‘＜条件式＞’；

※WHEREは必ずFROMの直後につけよう。

例

|  |  |
| --- | --- |
| 商品名 | 商品分類 |
| Tシャツ | 衣類 |
| カッターシャツ | 衣類 |

SELECT shohin\_mei,shohin\_bunrui

FROM shohin

WHERE shohin\_bunrui = ‘衣服’;

　　コメントの書き方

・--（マイナスマイナス）　の後に記述ができる。（１行の中でしか記述はできない）

・/\* \*/で囲んだところがコメントになる。

※文の途中に挟むこともできる。

２－２　算術演算子と比較演算子

|  |  |
| --- | --- |
| ＜列名＞ | ＜列名＞\*2 |
| 500 | 1000 |
| 320 | 340 |
| 2800 | 5600 |
| 2000 | 4000 |
| 5000 | 10000 |

算術演算子（+.　-.　\*.　/）

SELECT ＜列名A＞

　　　　　　＜列名A＞　＊　2

FROM　＜テーブル名＞；

右のように表示される。

※1.　\* . / の方が先に計算される。

※2. NULLが計算にあると正解が空白になる。

|  |  |  |  |
| --- | --- | --- | --- |
| ＝ | ～と等しい | ＜＞ | ～と等しくない |
| ＞＝ | ～以上 | ＜＝ | ～以下 |
| ＞ | ～より大きい | ＜ | ～より小さい |

　　　比較演算子（ = . <> . < . > . <=. >=）

SELECT ＜列名＞

FROM　＜テーブル名＞

WHERE＜列名＞　＜演算子＞　＜数字＞

|  |  |  |
| --- | --- | --- |
| 商品名 | 商品分類 | 販売単価 |
| Tシャツ | 衣類 | 1000 |
| カッターシャツ | 衣類 | 4000 |
| 包丁 | キッチン用品 | 2800 |
| 圧力鍋 | キッチン用品 | 6800 |

　 例

SELECT shohin\_mei,shohin\_bunrui,hanbai\_tanka

FROM shohin

WHERE hanbai\_tanka <> 500;

販売単価が５００円じゃないものが表示される。

比較演算子の中では文字同士の計算もできる。

|  |  |  |
| --- | --- | --- |
| 商品名 | 販売単価 | 仕入単価 |
| Tシャツ | 1000 | 500 |
| カッターシャツ | 4000 | 2800 |
| 圧力鍋 | 6800 | 5000 |

　例

SELECT shohin\_mei,hanbai\_tanka、shiire\_tanka

FROM shohin

WHERE hanbai\_tanka – shire\_tanka >= 500;

↑

販売単価から仕入単価を引いたものが５００以上のものの商品名と販売単価、仕入単価の表示

不等号の注意

|  |
| --- |
| ｃｈｒ |
| 1 |
| 2 |
| 3 |
| 10 |
| 222 |

不等号を使うときに

例

SELECT chr このようなものを作ったとき、単純に２より大きな数字が

FROM Chars 表示されるわけではなく、２より大きいデータを選択する。

WHERE chr > ‘2’

この時表示されるのは、3と222だけ。ほかの数字は２よりサイズが小さいため表示されない。

　　　NULLに比較演算子は使えない

検索条件にNULLが入っているとNULLがそのものかどうかが不明なので表示されるものから除外される。

|  |  |
| --- | --- |
| 商品名 | 仕入単価 |
| Tシャツ | 500 |
| 穴あけパンチ | 320 |
| カッターシャツ | 2800 |
| 包丁 | 2000 |
| 圧力鍋 | 5000 |

例 結果

SELECT shohin\_mei,shire\_tanka

FROM shohin

WHERE shire\_tanka <> 2800

この時フォークとボールペンは仕入単価が

NULLなので表示できない。

もし表示するなら ではなく このように表示する。

SELECT shohin\_mei,shire\_tanka SELECT shohin\_mei,shire\_tanka

FROM shohin FROM shohin

WHERE shire\_tanka = NULL WHERE shire\_tanka IS NULL

論理演算子

先ほどの計算式に　NOT　をつけることで　～ではないもの　という条件式を作ることができる。

|  |  |  |
| --- | --- | --- |
| 商品名 | 商品分類 | 販売単価 |
| Tシャツ | 衣類 | 1000 |
| カッターシャツ | 衣類 | 4000 |
| 包丁 | キッチン用品 | 2800 |
| 圧力鍋 | キッチン用品 | 6800 |

例

SELECT shohin\_mei,shohin\_bunrui,hanbai\_tanka

FROM shohin

WHERE hanbai\_tanka >=1000;

これにNOTをつけると

|  |  |  |
| --- | --- | --- |
| 商品名 | 商品分類 | 仕入単価 |
| Tシャツ | 衣類 | 500 |
| 穴あけパンチ | 事務用品 | 320 |

SELECT shohin\_mei,shohin\_bunrui,hanbai\_tanka

FROM shohin

WHERE NOT hanbai\_tanka >=1000;

となる。

条件を否定するのはNOT演算子だが表示方法がほかにあるので無理に使う必要はない。

　　　AND演算子とOR演算子

AND演算子・・・（⋀）かつ、とも呼び論理積、X　AND　Y　はXでありYでもあるというイメージ

OR演算子・・・（∨）または、とも呼び論理和、X　OR　Y　はXまたはYであるというイメージ

例

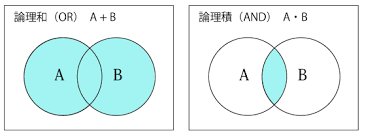
|  |  |
| --- | --- |
| 包丁 | 2000 |
| 圧力鍋 | 5000 |

SELECT shohin\_mei,shiire\_tanka

FROM shohin

　　　　WHERE shohin\_bunrui = 'キッチン用品'

　 　　　AND hanbai\_tanka >= 3000 となる。

;

SELECT shohin\_mei,shiire\_tanka

　　　　 FROM shohin

WHERE shohin\_bunrui = 'キッチン用品'

　 　　 　OR hanbai\_tanka >= 3000;

カッコ（）をつけると強くなる！

例

|  |  |  |
| --- | --- | --- |
| Tシャツ | 衣服 | 2009-09-20 |
| 穴あけパンチ | 事務用品 | 2009-09-11 |
| 包丁 | キッチン用品 | 2009-09-20 |
| フォーク | キッチン用品 | 2009-09-20 |

SELECT shohin\_mei,shohin\_bunrui,torokubi

FROM shohin

WHERE shohin\_bunrui = '事務用品'

AND torokubi ='2009-09-11'

OR torokubi ='2009-09-20';

そのまま打つと事務用品以外も出てしまい思ったものが表示されない。

そこで（）をつけることでその中のものが優先的に選定される

|  |  |  |
| --- | --- | --- |
| 穴あけパンチ | 事務用品 | 2009-09-11 |

SELECT shohin\_mei,shohin\_bunrui,torokubi

FROM shohin

WHERE shohin\_bunrui = '事務用品'

AND (torokubi ='2009-09-11'

OR torokubi ='2009-09-20'); （）をつけたことでANDとORのところが優先された。

　　論理演算と真理値

論理演算子・・・NOT、AND、ORの３つの演算子,真（TRUE）、偽（FALSE）

真理表　・・・論理演算をした結果を表にしたもの

AND OR NOT

|  |  |  |
| --- | --- | --- |
| X | Y | X AND Y |
| 真 | 真 | 真 |
| 真 | 偽 | 偽 |
| 偽 | 真 | 偽 |
| 偽 | 偽 | 偽 |

|  |  |  |
| --- | --- | --- |
| X | Y | X OR Y |
| 真 | 真 | 真 |
| 真 | 偽 | 真 |
| 偽 | 真 | 真 |
| 偽 | 偽 | 偽 |

|  |  |
| --- | --- |
| X | NOT X |
| 真 | 偽 |
| 偽 | 真 |

　NULLを含む真理値

三値理論・・・NULLをつかった真理値、NULLの場合は不（アンノウン）

AND OR

|  |  |  |
| --- | --- | --- |
| X | Y | X AND Y |
| 真 | 真 | 真 |
| 真 | 偽 | 偽 |
| 偽 | 真 | 偽 |
| 偽 | 偽 | 偽 |

|  |  |  |
| --- | --- | --- |
| X | Y | X AND Y |
| 真 | 真 | 真 |
| 真 | 偽 | 偽 |
| 真 | 不 | 不 |
| 偽 | 真 | 偽 |
| 偽 | 偽 | 偽 |
| 偽 | 不 | 偽 |