



東京ITスクール

TOKYO IT SCHOOL

演算子

目次

1. 算術演算子と比較演算子	1
2. 論理演算子	2
3. 述語	6

1. 算術演算子と比較演算子

1 算術演算子

SQL 文の中には計算式を書くことが出来ます。SQL 文の中で使える四則演算の主な記号は次の通りです。

意味	記号
足し算	+
引き算	-
掛け算	*
割り算	/

ここに挙げた四則演算を行う記号（+、-、*、/）は算術演算子と呼ばれます。演算子とは、演算子の両辺にある値を使って四則演算や文字列の結合、数値の大小比較などの演算を行い、その結果を返す記号のことです。

なお、SQL では通常の計算式と同じように括弧（ ）も使うことが出来ます。括弧（ ）の中にある計算式の優先順が上がり、先に計算が行われます。

2 NULL を含む演算

NULL を含んだ計算は、必ず NULL になります。例えば、5 / 0 のように 0 で割ろうとするとエラーになりますが、NULL を 0 で割る場合だけはエラーにならず NULL になります。

3 比較演算子

WHERE 句の使用の際に学んだのが比較演算子です。主な比較演算子は以下の通りです。

演算子	意味
=	～と等しい
<>	～と等しくない
>=	～以上
>	～より大きい
<=	～以下
<	～より小さい

これらの比較演算子は文字、数値、日付など、ほぼ全てのデータ型の列、値を比較することが出来ます。

4 NULL に比較演算子は使えない

NULL には比較演算子が使用出来ません。NULL かどうかの判定は、「IS NULL」または「IS NOT NULL」演算子を使用します。

2. 論理演算子

論理演算子の学習には以下のような商品テーブル（Item）を利用します。

Item テーブル

item_id (商品ID)	item_name (商品名)	category_id (商品分類)	sel_price (販売単価)	pur_price (仕入単価)	reg_date (登録日)
0001	シャツ	1	1000	500	2009-09-20
0002	ホッチキス	2	500	320	2009-09-11
0003	セーター	1	4000	2800	NULL
0004	包丁	3	3000	2800	2009-09-20
0005	フライパン	3	6800	5000	2009-01-15
0006	フォーク	3	500	NULL	2009-09-20
0007	スプーン	3	880	790	2008-04-28
0008	ボールペン	2	100	NULL	2009-11-11

Item テーブル作成用の SQL 文を以下に示します。

(サンプルコード) Item テーブル

```
CREATE TABLE Item
(item_id      CHAR(4)      NOT NULL,
 item_name    VARCHAR(100) NOT NULL,
 category_id  INTEGER      NOT NULL,
 sel_price    INTEGER      ,
 pur_price    INTEGER      ,
 reg_date     DATE         ,
 PRIMARY KEY (item_id));
```

Item テーブルには、以下の 8 レコードを登録します。

(サンプルコード) Item テーブルへの登録

```
INSERT INTO Item VALUES ('0001', 'シャツ', 1, 1000, 500, '2009-09-20');
INSERT INTO Item VALUES ('0002', 'ホッチキス', 2, 500, 320, '2009-09-11');
INSERT INTO Item VALUES ('0003', 'セーター', 1, 4000, 2800, NULL);
INSERT INTO Item VALUES ('0004', '包丁', 3, 3000, 2800, '2009-09-20');
INSERT INTO Item VALUES ('0005', 'フライパン', 3, 5000, 2800, '2009-01-15');
INSERT INTO Item VALUES ('0006', 'フォーク', 3, NULL, 2800, '2009-09-20');
INSERT INTO Item VALUES ('0007', 'スプーン', 3, 790, 2800, '2008-04-28');
INSERT INTO Item VALUES ('0008', 'ボールペン', 2, NULL, 2800, '2009-11-11');

COMMIT;
```

1 NOT 演算子

算術演算子と比較演算子で、「～ではない」という否定の条件を指定する場合に<>演算子を使うと説明しましたが、同じ否定でも、もう少し広く使える演算子として NOT があります。NOT はそれ単独では使いません。ほかの検索条件と組み合わせて使います。

例えば、「販売単価（sel_price）が 1000 円以上」の行を選択する SELECT 文は以下の通りです。

(サンプルコード) NOT 演算子

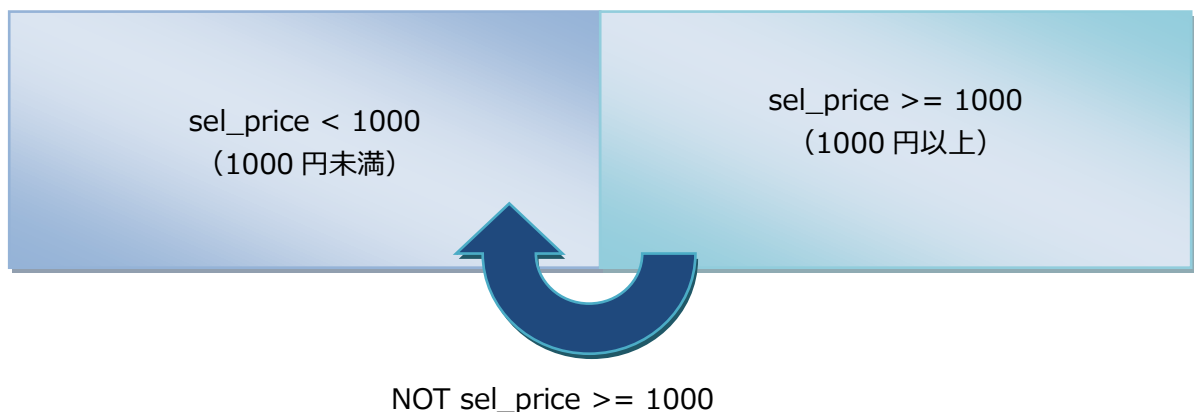
```
SELECT item_name, category_id, sel_price  
FROM Item  
WHERE NOT sel_price >= 1000;
```

(実行結果)

ITEM_NAME	CATEGORY_ID	SEL_PRICE

ホッチキス	2	500
スプーン	3	790

NOT 演算子をつけたときの検索条件の変化のイメージを以下に示します。



2 AND 演算子と OR 演算子

WHERE 句では AND 演算子や OR 演算子を使って、複数の検索条件を組み合わせることが出来ます。

AND 演算子は「両辺の検索条件が両方とも成り立つときに、全体の検索条件として成り立つ」という働きがあります。一方、OR 演算子は「両辺の検索条件のうち、どちらか一方あるいは両方が成り立つときに全体の検索条件として成り立つ」という働きがあります。

つまり、AND は「かつ」、OR は「または」に相当します。

例えば、Item テーブルから「商品分類がキッチン用品（category_id = 3）でかつ販売単価が 3000 円以上（sel_price >= 3000）の商品」を選択する検索条件を表わすには、AND 演算子を使います。以下に例を示します。

(サンプルコード) AND 演算子

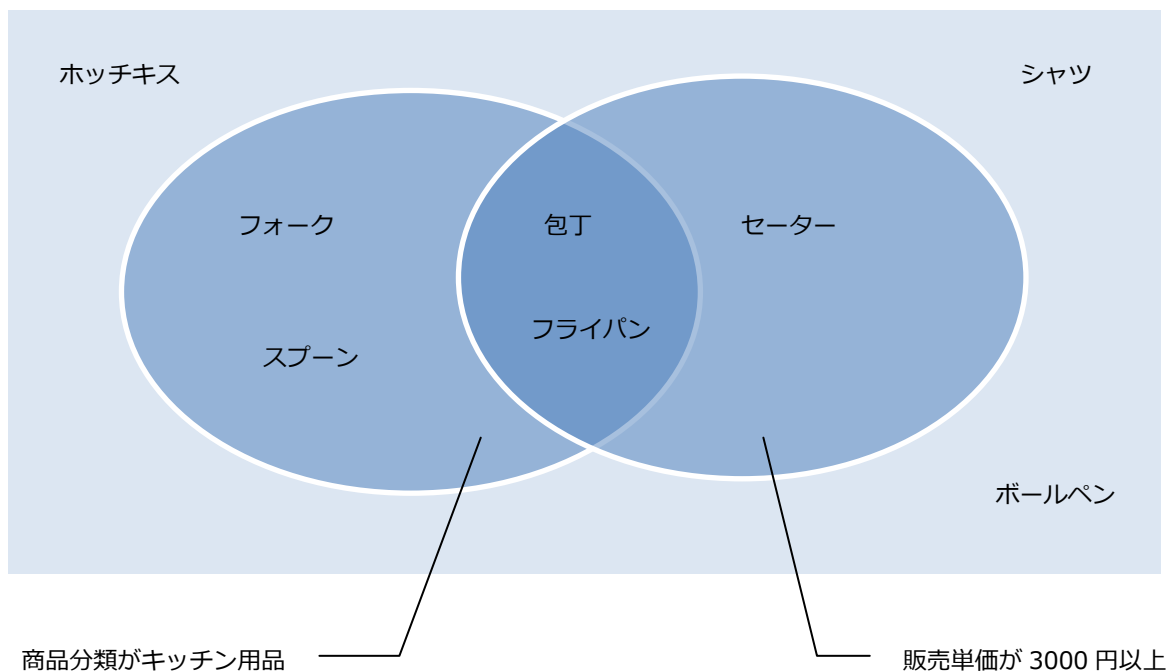
```
SELECT item_name, pur_price  
FROM Item  
WHERE category_id = 3  
AND sel_price >= 3000;
```

(実行結果)

ITEM_NAME	PUR_PRICE

包丁	2800
フライパン	2800

AND 演算子の働きを以下に示します。



一方、「商品分類がキッチン用品 (category_id = 3) または販売単価が 3000 円以上 (sel_price >= 3000) の商品」を選択する検索条件を表わすには、OR 演算子を使います。

(サンプルコード) OR 演算子

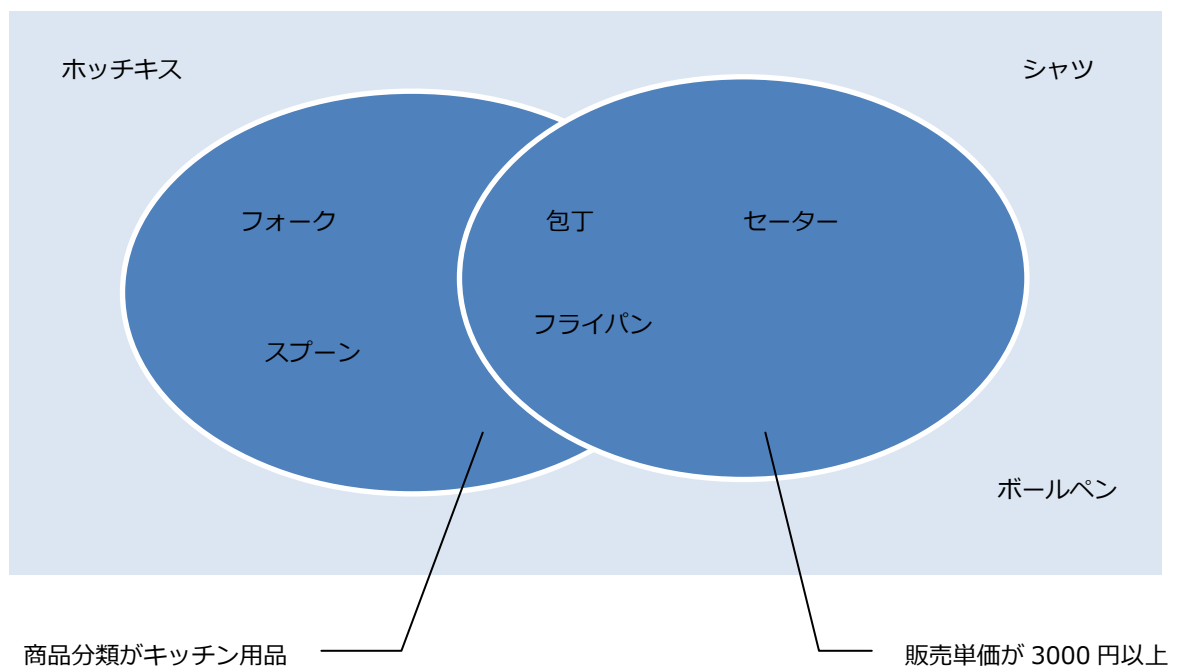
```
SELECT item_name, pur_price  
FROM Item  
WHERE category_id = 3  
OR sel_price >= 3000;
```

(実行結果)

ITEM_NAME	PUR_PRICE

セーター	2800
包丁	2800
フライパン	2800
フォーク	2800
スプーン	2800

OR 演算子の働きを以下に示します。



3. 述語

SQL で抽出条件を記述するときには、「述語 (predicate)」を使うことができます。述語とは、関数の一種で、演算子とよく似ています。実は、=、<、>、<>などの比較演算子は、正確には比較述語という述語の一種です。

この節では利用頻度の高い述語として、「LIKE」、「BETWEEN」、「IN」の3つを学びます。

1 LIKE

文字列が一致するかどうかを調べる際には、「=」を使用します。LIKE は「=」をよりあいまいにしたものです。具体的には、LIKE を使用することにより、「○○と一致する」ではなく、「○○という文字が含まれている」という条件を作ることができます。

このような LIKE を使用した検索を「あいまい検索」と呼ぶ場合もあります。また、あいまい検索をする文字列には、「%」や「_」を付与する必要があります。これらをワイルドカードと呼びます。

「%」は任意の文字列を指し、「_」は任意の 1 文字を指します。これらを文字列の後ろに付けて LIKE 検索を行うことを前方一致検索と呼びます。逆に前に付けると後方一致検索となります。例えば、「株式会社%」とすると、「株式会社」で始まるという意味になり、「株式会社システムシェアード」などが検索出来ます。あいまいな部分の文字列の長さが決まっている場合は、「_」を使用し、決まっていない場合は「%」を使用すると良いでしょう。

「%」や「_」は文字列のどこに埋め込んでも構いません。複数回使用することも可能です。特に使用頻度が高いのは、「%」を前後両方に付けて「%○○%」などとすることで、文字列のどこかに「○○」が含まれているものを検索する中間一致検索です。

以下に使用例を示します。こちらは item_name のどこかに「ー」(伸ばし棒)が含まれている列を検索しています。

(サンプルコード) LIKE

```
SELECT item_name FROM Item WHERE item_name LIKE '%ー%';
```

(実行結果)

ITEM_NAME
セーター
フォーク
スプーン
ボールペン

2 BETWEEN

BETWEEN は、範囲の絞り込みに使用します。具体的には、「○○以上△△以下」を「BETWEEN
○○ AND △△」と表すことが出来ます。「より大きい」や「より小さい」ではありませんので
注意してください。

以下に使用例を示します。

(サンプルコード) BETWEEN

```
SELECT item_name, sel_price  
FROM Item  
WHERE sel_price BETWEEN 500 AND 3000;
```

(実行結果)

ITEM_NAME	SEL_PRICE

シャツ	1000
ホッチキス	500
包丁	3000
スプーン	790

3 IN

「○○=△△」という形式の条件を複数回、「OR」で結び付けることが多々あります。もしも
○○の部分と同じで△△のみが違う場合、(つまり同じ列に対して値だけを変えて OR 条件とし
ている場合) IN を使用すると非常に簡潔な SQL 文を記述することが出来ます。

具体的には、「○○=△△ OR ○○=▲▲ OR ○○=□□」は、「○○ IN (△△, ▲▲, □□)」
と記述することが出来ます。逆に「いずれも含まれない」としたい場合は、「IN」を「NOT IN」
とすることも出来ます。

以下に使用例を示します。

(サンプルコード) IN

```
SELECT item_name, sel_price FROM Item WHERE sel_price IN (500, 790, 1000);
```

(実行結果)

ITEM_NAME	SEL_PRICE

シャツ	1000
ホッチキス	500
スプーン	790