LibreOffice　Base マニュアル(データ加工編)　組込 DB：Firebird

～目次～

ページ

BASE のオブジェクトを把握しよう 1

データ型について理解しよう 1

データインポートに向けたチェック事項と事前作業 3

インポート作業に入る前のデータ型指定 4

データインポートと同時に新規テーブルを作成するには？（表計算データ） 4

既存テーブルへインポートするには？（表計算データ） 5

データエクスポート（手作業によるエクスポート） 7

「欲しい項目のデータ一覧」を作成するには？（選択クエリー） 8

データを連結して「欲しい項目のデータ一覧」を作成するには？ 10

クエリーで利用する演算式と条件の書き方 11

クエリーで利用する代表的な演算例 12

クエリーの結果を１つに結合する（ユニオンクエリー） 15

処理実行時に入力した条件でレコードを抽出する（パラメータークエリー） 16

クエリーで利用する「フィルター演算子」 17

下線部をクリックすると該当ページへ移動します、目次へ戻る場合は各ページのページ番号をクリックして下さい

1. BASE のオブジェクトを把握しよう

Base は主に４つのオブジェクトで構成されています。
この４つのオブジェクトを操作して、自分がほしい結果を導き出す作業を行います。

代表的なオブジェクトは4つだけ!

・テーブル　・・・　データを保管する場所（データを格納する単位・データの論理的な集まり）
・クエリー　・・・　仮想的な表（多様な切り口を持つ）、データ連結や計算もクエリーで実施する（問い合わせと結果（SQL 含む））
・フォーム　・・・　入力画面等のインターフェース（データの表示方法（一覧・カード型 etc））
・レポート　・・・　印刷帳票の様式（グループ集計や小計等の多様な見せ方が可能）

※このマニュアルでは「ビュー」や「ストアドプロシージャ」については考慮しません。

《データ加工の３ステップ》

加工するデータ類をテーブルに登録する
クエリーを使って、並び替え・グループ化・集計・計算等を行なう
クエリーの結果をデータとして出力したり、レポートとして帳票を作成する

※出力データ形式

- 表計算ソフトで読める形式（Ms-Excel「OOXML」・「XLS」形式・Calc「ODF」形式・TAB・CSV・プレーンテキスト形式等）
- Writer「ODF」形式（拡張子 ODT）
- PDF 形式

2. データ型について理解しよう

BASE に限らず、Access 等を含むデータベースでの作業においては、データの持つ型を意識する必要があります。
データ型と言うと難しい感じがしますが、表計算ソフトのセルの書式設定と同じと考えてもらって結構です。
Calc や Excel の「セルの書式設定」が「日付」であるデータは日付型のデータ型を指定すればよく、「数値」の書式設定のデータは数値型、「テキスト」の書式設定のデータは文字列型を指定すれば大丈夫です。

Base（Firebird）のデータ型（◎や○のところを押さえておくだけでも良い）

分類	データ型名	範囲
文字列◎	CHAR(m) VARCHAR(m)	半角英数（ASCII）　1～32767 文字 全角（SJIS_0208）　1～16383 文字 CHAR(m)は m 文字の固定長 VARCHAR(m)は最大 m 文字の可変長
整数○	SMALLINT INTEGER	-32,768 ～ 32,767 -2,147,483,648 ～2,147,483,647
長整数◎	BIGINT	符号付 64 ビット整数　Dialect3/FB1.5 以降で使用可能

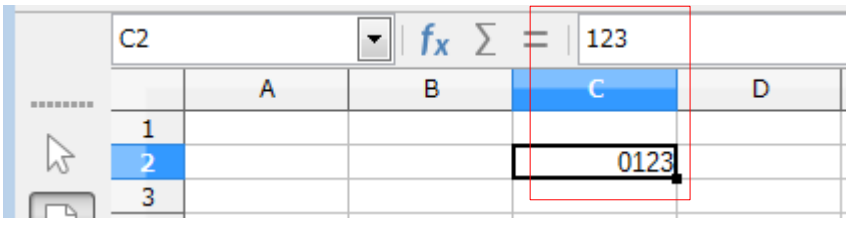
分類	データ型名	範囲
実数	FLOAT DOUBLE PRECISION	3.4×10 の-38 乗～3.4×10 の 38 乗　有効桁数 7 桁 1.7×10 の-308 乗～1.7×10 の 308 乗　有効桁数 15 桁
固定小数点	NUMERIC（m,n） DECIMAL（m,n）	有効桁数 m = 1 ～ 15 小数点以下の桁数 n=1～15（m >= n） 最高 m 桁の有効数字を格納 有効桁数 m = 1 ～ 15 小数点以下の桁数 n=1～15（m >= n） 最低 m 桁の有効数字を格納
日付◎	DATE TIME TIMESTAMP	西暦 100 年 1 月 1 日～32768 年 2 月 29 日までの年月日 00:00:00～23:59:59 までの時分秒 西暦 100 年 1 月 1 日～32768 年 2 月 29 日までの年月日と時分秒
その他	BLOB	グラフィック，文字などの大量データに使用:可変長
論理型	BOOLEAN	真理値の「真 = true」と「偽 = false」

データベースでは項目の事をフィールドと呼びます。
フィールド毎にデータ型（フィールドタイプ）を定義することで規定以外の値が保存されることを防止します。
誤った型でデータを登録すると、数値なのに計算ができなかったり、日付間の期間計算ができない等の不具合が発生しますので、フィールドには正しいデータ型を設定する必要があります。

数値とテキスト値の違いは、計算する値が数値であり、計算しない数値は文字列と考えるとスッキリします。
例:売価は計算対象なので数値型で指定しますが、社員番号は計算しない「コード」なので文字列で指定します。

【データ型の目安】
テキスト型・・・文字・計算しない数字（郵便番号・電話番号・各種コード等）、長文には VARCHAR を利用します
数値型・・・金額や整数・少数点を含む値に利用します（ほとんど INTEGER・BIGINT で事足ります）
固定長テキストの CHAR は入力値が指定文字数に足りない場合、不足位置にはスペースが入りますので注意が必要です。

【数値なのに先頭にゼロが入っている?】　～セルの書式に注意～
通常、セルの値が数値であれば、先頭のゼロは表示されません。（0123:テキスト　　123:数値）
セルの書式設定にある「ユーザー定義」を設定　0000　すると、書式は数値のまま先頭のゼロを表示できます。
先頭にゼロが付いているから、テキストになっていると早合点しないように注意してください。



※CSV ファイルを Excel で開いたときの頭ゼロ消え対応として、書式設定されている場合が多い。
CSV ファイルを開く際には Calc やテキストエディタを利用するようにしましょう。

3. データインポートに向けたチェック事項と事前作業

表計算ソフトや CSV などのデータをテーブルに取り込む作業を「インポート」と呼びます。
素材となるデータを確実にインポートするためのチェック事項と事前作業について確認します。

《データインポートに向けたチェック事項と事前作業》

- 表計算データ・CSV データをインポートする
  - データ内に空行や空列がないか？　→　空行・空列は全て削除
  - セルの結合が無いか？　→　セルの結合は全て解除
  - セルの書式設定を一旦、全て解除します。
  - インポート先テーブルのデータ型に合わせた書式を「セルの書式」としてインポート元データに再設定します。
  - インポートする場合（日付は西暦 2019-04-01 or 2019/04/01 形式とし、金額には￥マークや 3 桁区切りは付けません）

- テキストデータをインポートする
  テキストファイルは直接インポートできない為、一旦 Calc で開いて整形した後、Calc データとしてインポートします。
  - データ内に空行や空列がないか？　→　空行・空列は全て削除
  - インポートする場合（日付は 2019-04-01 or 2019/04/01 形式とし、金額には￥マークや 3 桁区切りは付けません）

  ※日付が文字列の場合は DateValue 関数で日付に変換しておく必要があります。

《現在のテーブルにあるデータを削除すべきか?》

- データ処理を目的としてインポートする事例なので、テーブル中のデータは削除をお勧めします。
- 削除しない方法も可能ですが、日付等による対象データの絞込み等のチェックが必要になります。
- 累積データとしての利用がないのであれば、不要なデータは削除しておきましょう。
- 削除方法は SQL 文（クエリー）を発行する方法でも、手作業による一括削除でもかまいません。

手作業による一括削除例
テーブルを開いて削除したいレコードを選択、複数のレコードを選択する場合は続けて SHIFT キーを押した状態で他のレコードを選択した後 DEL キーを押下する。

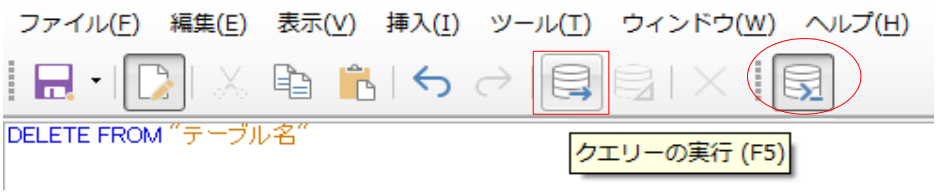
複数のレコードを選択する場合は続けて SHIFT キーを押した状態で他のレコードを選択する

	商品コード	商品名	単価
▸	0001	えんぴつ	100
	0002	シャープペン	200
	0003	替芯0.5	100
	0004	色えんぴつ	150
	0005	けしごむ	100
+			

→

	商品コード	商品名	単価
	0001	えんぴつ	100
	0002	シャープペン	200
	0003	替芯0.5	100
	0004	色えんぴつ	150
▸	0005	けしごむ	100
+			

クエリーによる一括削除例
クエリーを開いて→SQL 表示でクエリーを作成　を選択し入力　DELETE FROM "テーブル名"
と入力し、右端の SQL コマンドを直接実行ボタンを有効にした後、クエリーの実行ボタンをクリック



4. インポート作業に入る前のデータ型指定

Calc（表計算ソフト）のデータをテーブルに取り込む作業手順です。
Calc のセルの書式設定を使って、事前に型指定してしまいます。
Calc のセルの書式設定でデータ型を決めます。（列全体を選択しての書式変更も可）

- 日付型で保存したいとき

  Calc ： 1999-12-31 形式で指定しておく

  ※データが文字列の場合は DateValue 関数で日付に変換しておく

  BASE ：取込時に日付 DATE 型で指定

- 数値型で保存したいとき

  Calc ：数値で指定しておく

  BASE ：取込時に数値 BIGINT/INT 型で指定

- 文字列型で保存したいとき

  Calc ：テキストで指定しておく

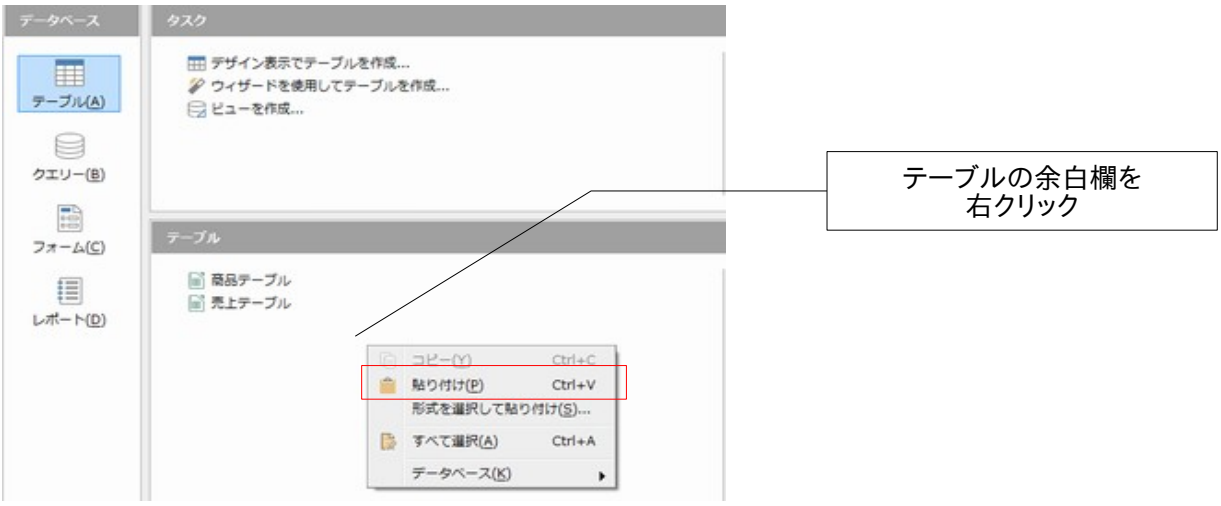
  BASE ：特に指定しなくても大丈夫

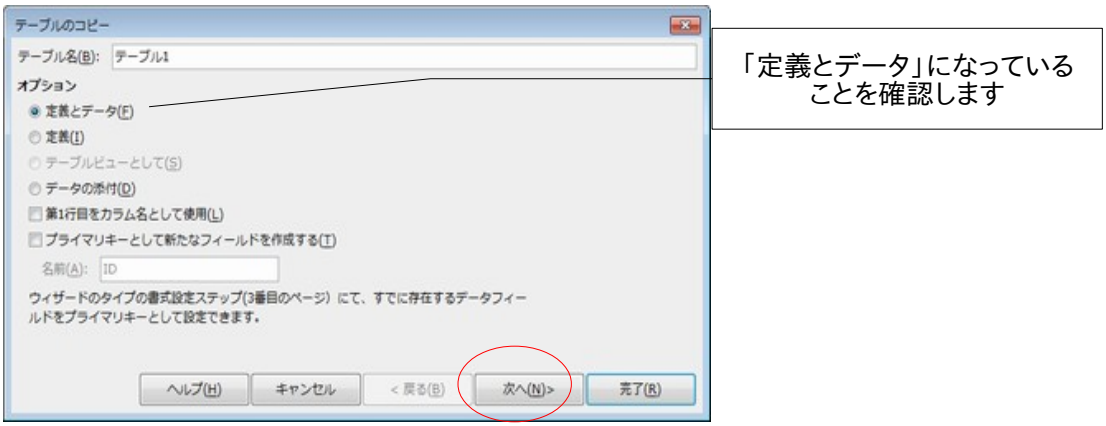
5. データインポートと同時に新規テーブルを作成するには？（表計算データ）

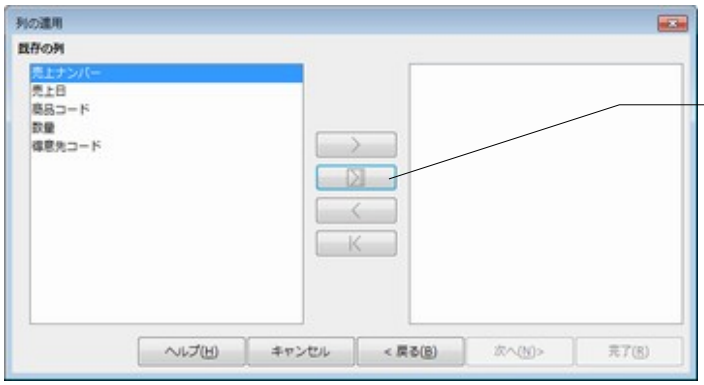
インポート作業は新規テーブルと既存テーブルへのインポートで操作が違います。

データインポートと同時に新規テーブルを作成する場合

- インポートデータを Calc で開きテーブルにインポートする範囲を選択します
- テーブル欄を開いて、テーブル欄の余白スペースを右クリックし「貼り付け」を選択



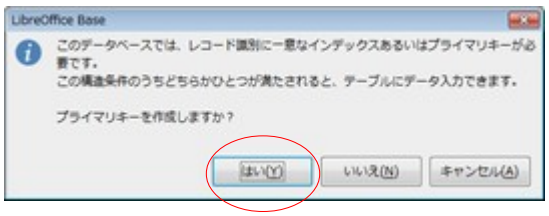




項目名を指定します

- 項目とデータ型を修正して処理を進めます

- プライマリーキー（主キー）の作成について、通常「作成する」ので「はい」を選びます

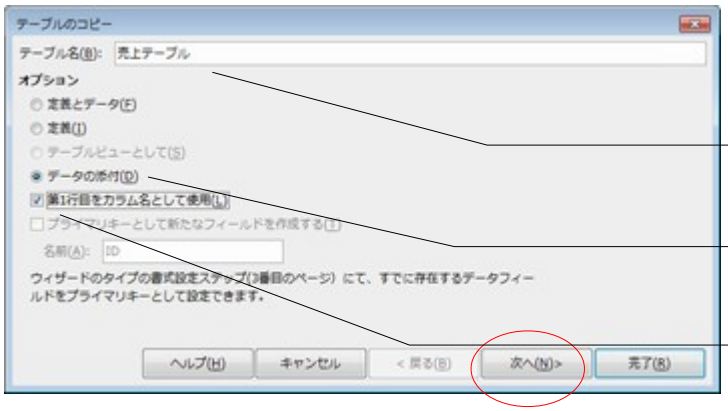


プライマリーキーは作成しなくても処理は可能です。
しかし、削除を含め一切の修正を受け付けないテーブルになりますので、修正等の可能性があるデータテーブルの場合、プライマリーキーは「作成する」をお勧めします。

6. 既存テーブルへインポートするには？（表計算データ）

既存テーブルへインポートする場合

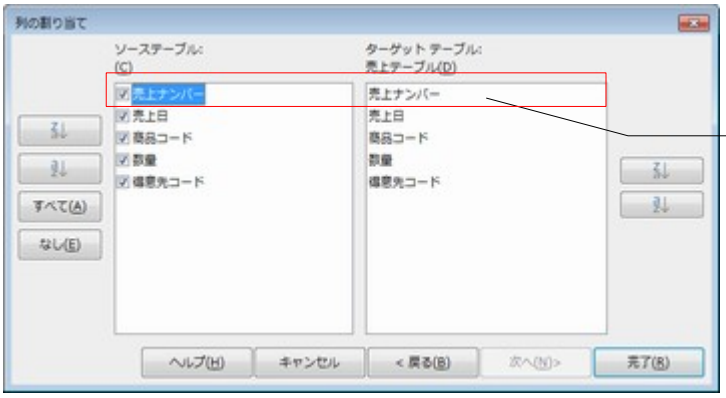
- インポートデータを Calc で開きテーブルにインポートする範囲を選択します
- テーブル欄を開いて、インポート先テーブル名を右クリックし「貼り付け」を選択
- テーブル項目とデータ項目との対応画面が表示されるので、対応項目同士を選択



既存テーブル名になっていることを確認します

データの添付になっていることを確認します

項目まで範囲指定している時はココにチェックを入れます



ソーステーブルとターゲットテーブルの
項目が対応していることを確認します

《オートナンバー型項目への対応》
オートナンバー型の項目があるテーブルへのインポートも既存テーブルへのインポートと同じですが
オートナンバー項目のデータは空欄にして範囲指定する点と取込時のチェックを外す点が違います。

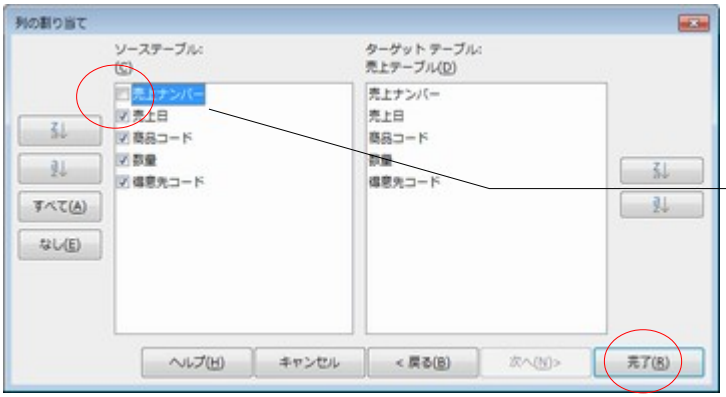
	売上ナンバー	売上日	商品コード	数量	得意先コード
▶	1	2019/05/01	0001	12	0000012
	2	2019/05/03	0001	10	0000011
	3	2019/05/06	0002	18	0000012
	4	2019/05/12	0003	8	0000010
	5	2019/05/16	0001	1	0000001
	6	2019/05/19	0004	10	0000012
	7	2019/05/18	0001	2	0000010
	8	2019/06/01	0002	2	0000011
	9	2019/06/02	0003	10	0000012
	10	2019/06/03	0004	4	0000013
*	<オートフィールド>				

売上ナンバーがオートフィールド(自動採番)になっています。
ここには自動採番された値が入りますので、インポートデータをセットしないように注意しなければなりません。

【インポートデータ側での範囲指定時】　※注意事項参照

G	H	I	J	K
売上ナンバー	売上日	商品コード	数量	得意先コード
	5月1日	0001	12	0000012
	5月3日	0001	10	0000011
	5月6日	0002	18	0000012
	5月12日	0003	8	0000010
	5月16日	0001	1	0000001
	5月19日	0004	10	0000012
	5月18日	0001	2	0000010
	6月1日	0002	2	0000011
	6月2日	0003	10	0000012
	6月3日	0004	4	0000013

オートナンバー型項目のデータは
空欄であることが必要です



ソーステーブル側の
チェックを外します

※注意事項
指定範囲に項目名を含んでいる場合、既存テーブルへインポートする際は「第 1 行目をカラム名として使用（L）」のチェックを忘れずに入れてください。

7. データエクスポート（手作業によるエクスポート）

インポート処理とは逆に表計算ソフトや CSV などのデータとして出力する作業を「エクスポート」と呼びます。
エクスポートは、該当するオブジェクト（テーブルやクエリー）を Calc シートへのドラッグ＆ドロップによって実施します。
※コピー＆貼り付け（いわゆるコピペ）ではなく、必ずドラッグ＆ドロップで作業してください
（コピー＆貼り付けすると、日本語が文字化けします）

	A	B	C
1			
2	商品コード	商品名	単価
3	P00001	,,ã,Ò,Ã	10
4	E00001	Ã,µfSf€	50
5	SP0001	fVff [fvfv	100
6	999001	,,,,,,,,	10
7			
8			
9	商品コード	商品名	単価
10	P00001	えんぴつ	10
11	E00001	消しゴム	50
12	SP0001	シャープペンシ	100
13	999001	sssss	10
14			
15			

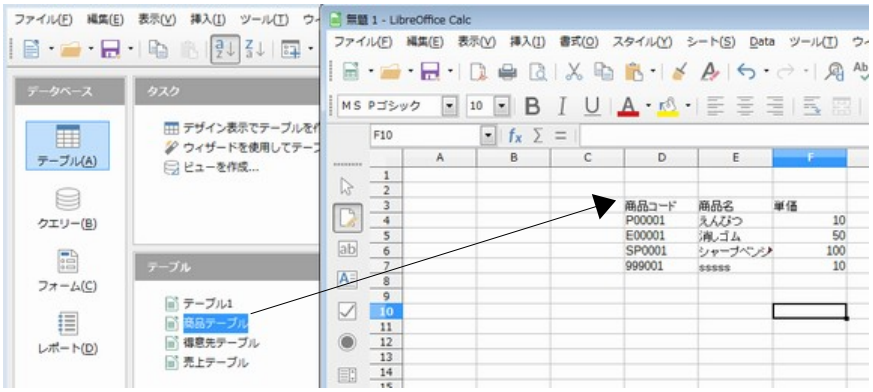
※文字化けした例

※正常出力例

- Calc のシートを開いてデータを出力するスペースを確保します
  - テーブルを出力するとき

    BASE 上で出力するテーブルを選択します

    あらかじめ開いておいた Calc のシート上にドラッグ＆ドロップ

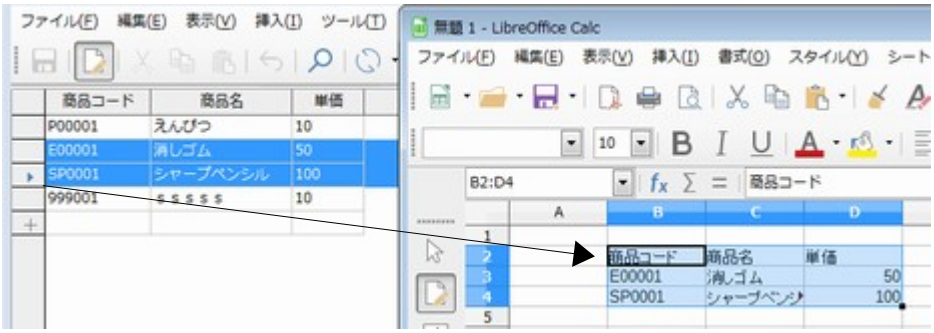


  - クエリーを出力するとき

    BASE 上で出力するクエリーを選択します

    あらかじめ開いておいた Calc のシート上にドラッグ＆ドロップ

※テーブル・クエリー共に全データだけでなく、レコード単位で範囲指定したデータでも出力できます。



- レポートの場合は通常の Writer 出力を Calc 出力へ変更することもできます
  - レポートを編集モードで開きます（レポート名を右クリック→編集を選択した状態です）

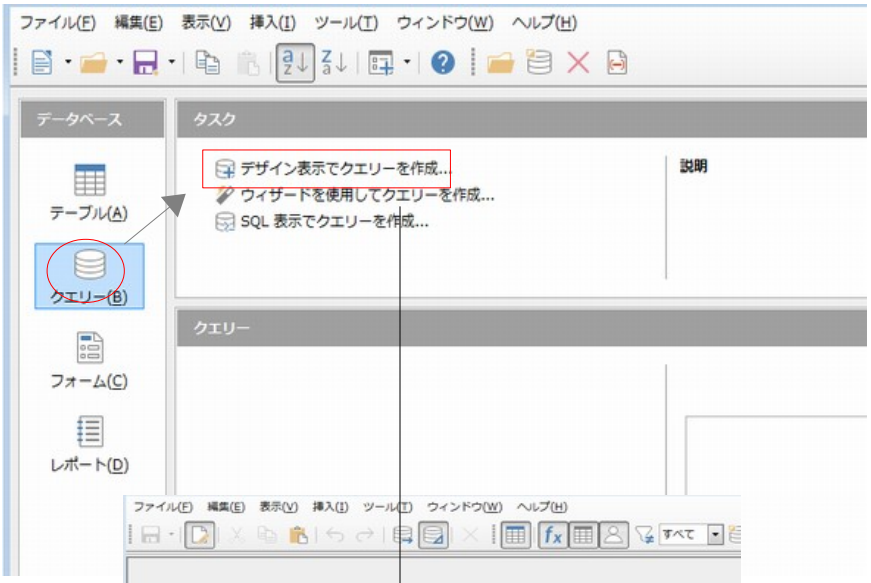
    メニュー　→　ファイル　→　レポート出力形式　→　表計算ドキュメント

    BASE 上でレポートを実行します

    Writer ではなく Calc を使ってレポートが作成されます

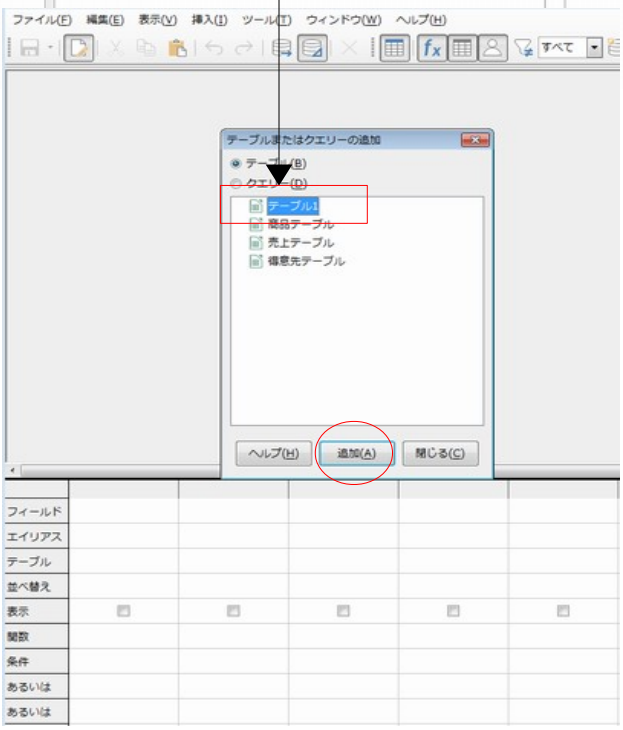
8.「欲しい項目のデーター覧」を作成するには？（選択クエリー）

Calc や Excel 等の表計算ソフトの場合、元データの項目の非表示や項目の削除によって項目絞込みデータを作成しますが、データベースでは「選択クエリー」を作成することで、元データは一切加工せずに欲しい項目の一覧を作成することができます。
作成したクエリーは結果ではなく作業を保存しますので、元データに変更や更新があった場合でも、クエリーの再実行だけで、欲しい時にデータの変更を反映させた結果データを得ることができます。
また、抽出条件を設定すれば、設定した条件に合致したデータを得ることも可能です。



クエリー画面を選択します

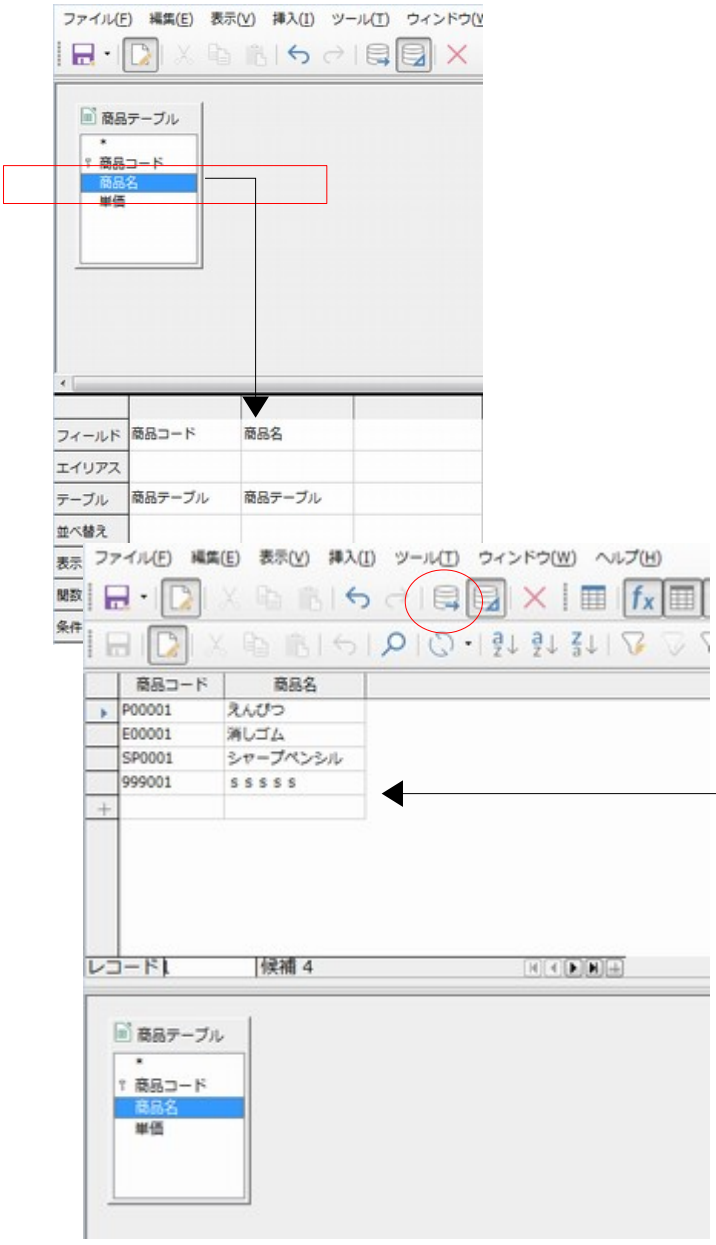
「デザイン表示でクエリーを作成」を選択します



テーブル・クエリーから対象となるデータを選択し「追加」をクリックします。
上段のウィンドウ内に選択したデータが表示されることを確認します。

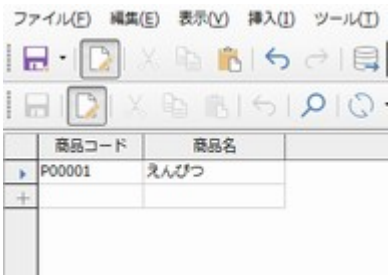
選択したテーブル・クエリーから対象となる項目名をＷクリックします。
下のフィールド欄に選択した項目名が表示されることを確認します。

例では「商品テーブル」を処理データとし、商品コードと商品名を表示項目に指定しています。



クエリーの実行（F5）ボタンをクリックすると結果が表示されます。

《抽出条件の設定》
右の例では、検索条件に「えんぴつ」を指定しています。

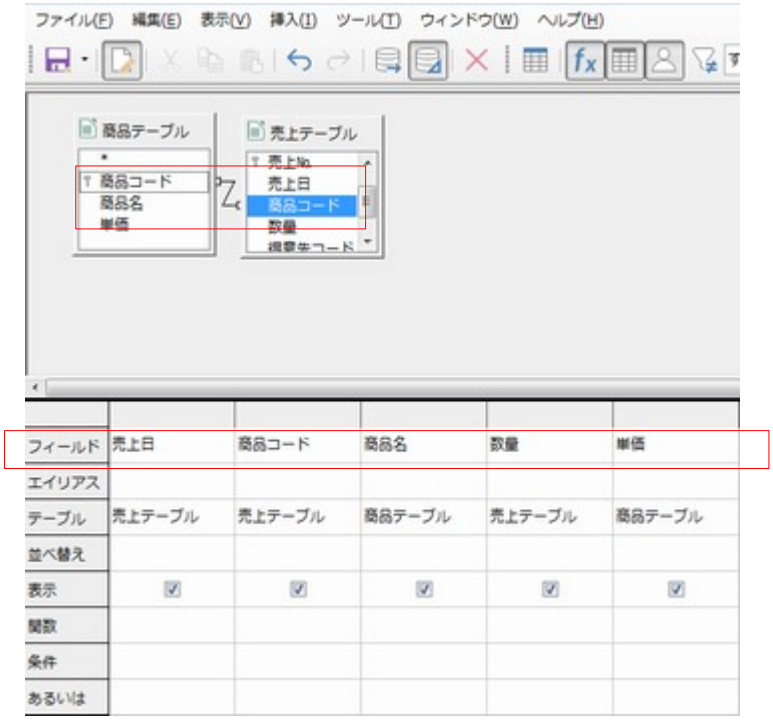


※抽出されました

9. データを連結して「欲しい項目のデーター覧」を作成するには？（選択クエリー）

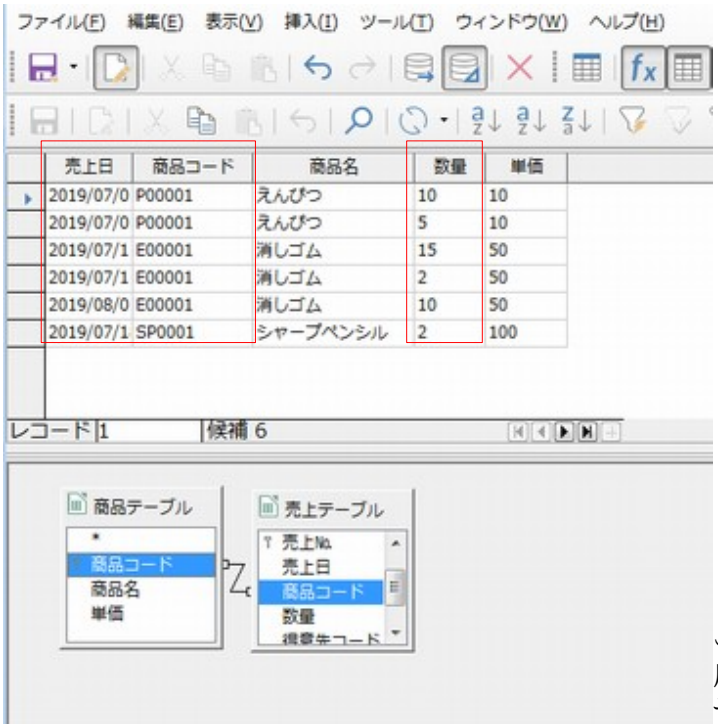
データベースはデータ同士を連結させ「選択クエリー」を作成することができます、データ間の連結には連結キーとなる項目を指定することが必要です。
尚、この処理でも元データは一切加工せずにデーター覧の作成が可能です。

✔ データ間を連結するには、2 つのデータ間で共通となる項目（連結キー）の設定が必要です。
～リレーションについては、LibreOffice　Base マニュアル(入門編)　組込 DB：Firebird　「リレーションシップを設定する」とは？　を参照下さい～



各テーブルの商品コード同士をリレーションで接続します

表示する項目を W クリックしてフィールド欄にセットしていきます。



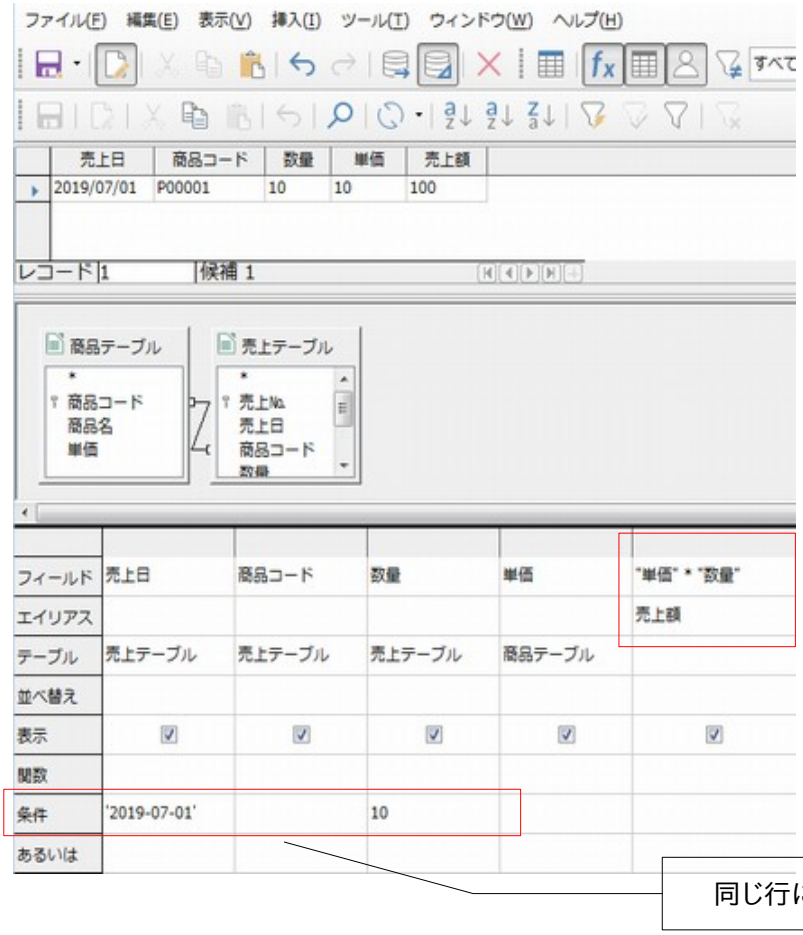
売上テーブルと商品テーブルの両方を連携させることで売上明細を表示させることができました。

- 商品名と単価は「商品テーブル」から
- 売上日と商品コードと数量は「売上テーブル」からそれぞれ連携させて一つのデータを成立させています。

これら関係性を使ってデータを再構成する形式のため、リレーショナルデータベースと呼ばれます。

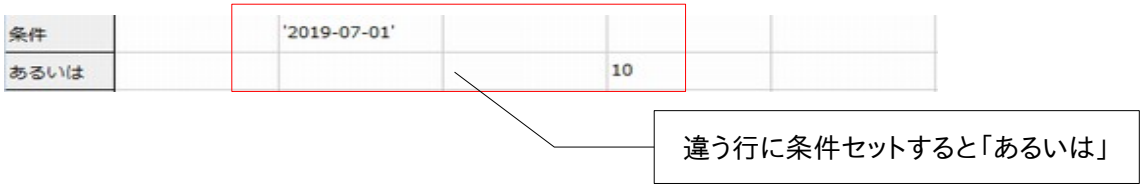
10. クエリーで利用する演算式と条件の書き方

クエリーでは四則演算や文字列結合などの処理を実施できます。
データベースによって多少の方言がありますが、Firebird の事例を挙げておきますので参考にしてください。



記入式: 単価 * 数量
乗算には * (アスタリスク)を使います
" は自動記入されます

※上記例では「売上日が 2019/07/01 で数量が 10 であるデータを抽出しなさい」となります。



※条件のどちらかを「あるいは」行に移動させるだけで、「売上日が 2019/07/01 か、あるいは数量が 10 であるデータ」の条件がセットされます。

11.クエリーで利用する代表的な演算例

データベースの演算式は数多くあります、代表的な「よく使う数式」を例示しますので参考にしてください。

■ **四則演算子**

算術演算子	意味
+	加算
-	減算
*	乗算
/	除算

■ **項目や文字列同士を結合（Access では&で結合していた）**
|| → Shift キー+¥　で入力可
"商品コード" || "商品名"

■ **左から 6 文字分の文字列取得**
LEFT("商品名", 6)

※ 使用文字コードによってはバイト数での抽出になる場合があります。

■ **文字列置換**　商品名のインチを inch に変更する
REPLACE("商品名", 'インチ', 'inch')

■ **文字列をつないで日付に変換（CAST 関数を使う）**
CAST('12' || '-july-' || '2019' AS DATE) "日付作成"
CAST('04' || '/12/' || '2019' AS DATE)

■ **項目の値（卸単価）が NULL のとき0を表示する**　　NULL＝空（カラ）
NULL を別の値に変換する
COALESCE ("卸単価", 0) "卸売り単価"

フィールド	商品名	卸単価	COALESCE ("卸単価", 0)
エイリアス			卸売り単価
テーブル	商品マスタ	商品マスタ	
並べ替え			
表示	☑	☑	☑

■ **各種日付データから項目を取り出す**

EXTRACT(YEAR FROM "処理日付") "処理年"
EXTRACT(MONTH FROM "処理日付") "処理月"
EXTRACT(DAY FROM "処理日付") "処理日"

フィールド	処理日	EXTRACT (YEAR FROM "処理日付")	EXTRACT (MONTH FROM "処理日付")	EXTRACT (DAY FROM "処理日付")
エイリアス		処理年	処理月	処理日
テーブル	売上データ			
並べ替え				
表示	☑	☑	☑	☑
関数				

■ **各種型変換**

CAST("処理日" AS CHAR (10)) "文字列"
CAST("処理日" AS DATE) "日付型"
CAST('12.54' AS NUMERIC (6 , 2)) "数字型"
CAST(12.54 AS CHAR (10)) "文字列の数字"

■ **日付加算**　現在の日付に１を加算

DATEADD(DAY, 1, current_date)

■ **曜日データを得る**

EXTRACT(WEEKDAY FROM "対象日付") "対象曜日"
結果:WEEKDAY 0～6　0 = Sunday

■ **文字列を１文字目から６文字分抽出する**

SUBSTRING ("商品名" FROM 1 FOR 6)

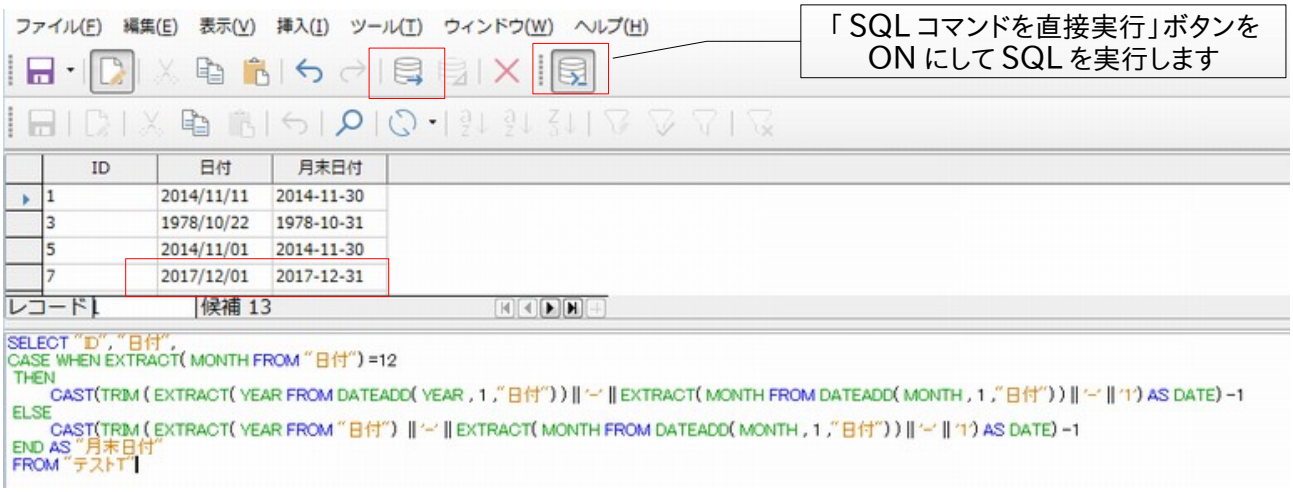
	商品名	単価	SUBSTRING
▶	えんぴつ	10	えんぴつ
	消しゴム	50	消しゴム
	シャープペンシル	100	シャープペン
	ｼｬｰﾌﾟﾍﾟﾝｼﾙ	120	ｼｬｰﾌﾟﾍ

フィールド	SUBSTRING ("商品名" FROM 1 FOR 6)
エイリアス	
テーブル	

※ 使用文字コードによってはバイト数での抽出になる場合があります。

■ **SQLで月末日を算出する(CASE 文)**

※月末日を求める場合は翌月１日を求め、翌月１日からマイナス１日する方法で算出しています。



CASE 文を使って 12 月データを分岐させています（12 月分のみ、年に 1 を加算する必要がある為）

```
SELECT "ID", "日付",
CASE WHEN EXTRACT( MONTH FROM "日付") =12
 THEN
        CAST(TRIM ( EXTRACT( YEAR FROM DATEADD( YEAR , 1 ,"日付") ) || '-' || EXTRACT( MONTH
FROM DATEADD( MONTH , 1 ,"日付") ) || '-' || '1') AS DATE) -1
 ELSE
        CAST(TRIM ( EXTRACT( YEAR FROM "日付")  || '-' || EXTRACT( MONTH FROM
DATEADD( MONTH , 1 ,"日付") ) || '-' || '1') AS DATE) -1
 END AS "月末日付"
 FROM "テストT"
```

※12 月に 1 を加算しない場合、前年の年末が表示されてしまいます。

■ オートナンバー型（自動採番）の数字をリセットします

ALTER TABLE "対象テーブル名" ALTER COLUMN "対象項目" RESTART

例） ALTER TABLE "データテーブル" ALTER COLUMN "Auto_ID" RESTART

■ テーブル内のデータを削除するには（DELETE 文）

DELETE は、レコードを削除するための構文で、すべてのレコードを削除、もしくは条件式を満たす特定のレコードだけを削除することができます。使い方は、FROM 句に対象となるテーブル名を指定し、特定のレコードのみを削除する場合は WHERE 句に条件式を設定します。

DELETE FROM "データテーブル"

WHERE を省略するとすべてのレコードが削除されます。

DELETE FROM "データテーブル" WHERE "Auto_ID" = 60

任意のレコードを削除するには、WHERE 句に条件式を設定します。

例では Auto_ID が 60 のデータレコードを削除します。

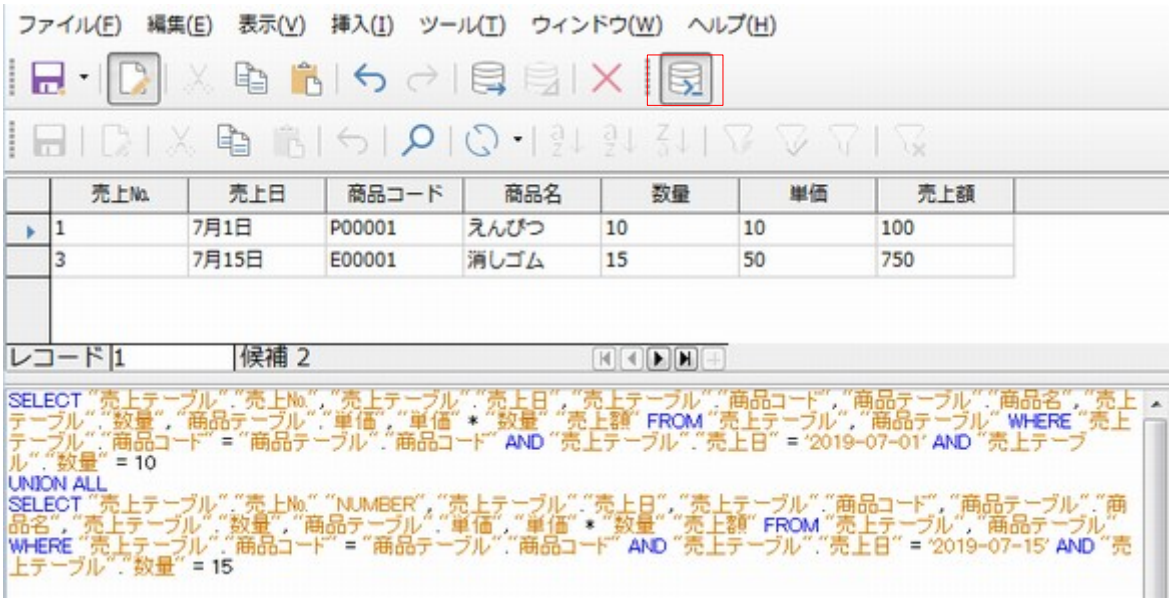
■ テーブル削除（テーブルそのものを削除します）

DROP TABLE "データテーブル"

※SQL を実行するよりもテーブルを右クリックして「削除」する方法をお勧めします。

12. クエリーの結果を１つに結合する（ユニオンクエリー）

ユニオンクエリーは複数のクエリーを 1 つの結果に結合する処理です。
結合するクエリーの項目数は、お互い同じである必要があります。



SELECT "売上テーブル"."売上№", "売上テーブル"."売上日", "売上テーブル"."商品コード", "商品テーブル"."商品名", "売上テーブル"."数量", "商品テーブル"."単価", "単価" * "数量" "売上額" FROM "売上テーブル", "商品テーブル" WHERE "売上テーブル"."商品コード" = "商品テーブル"."商品コード" AND "売上テーブル"."売上日" = '2019-07-01' AND "売上テーブル"."数量" = 10
UNION ALL
SELECT "売上テーブル"."売上№" "NUMBER", "売上テーブル"."売上日", "売上テーブル"."商品コード", "商品テーブル"."商品名", "売上テーブル"."数量", "商品テーブル"."単価", "単価" * "数量" "売上額" FROM "売上テーブル", "商品テーブル" WHERE "売上テーブル"."商品コード" = "商品テーブル"."商品コード" AND "売上テーブル"."売上日" = '2019-07-15' AND "売上テーブル"."数量" = 15

《UNION 結合の使い分け》

UNION ALL	結合した全明細を表示する
UNION	全項目で同じ（まったく同じ）明細がある場合、行を統合します

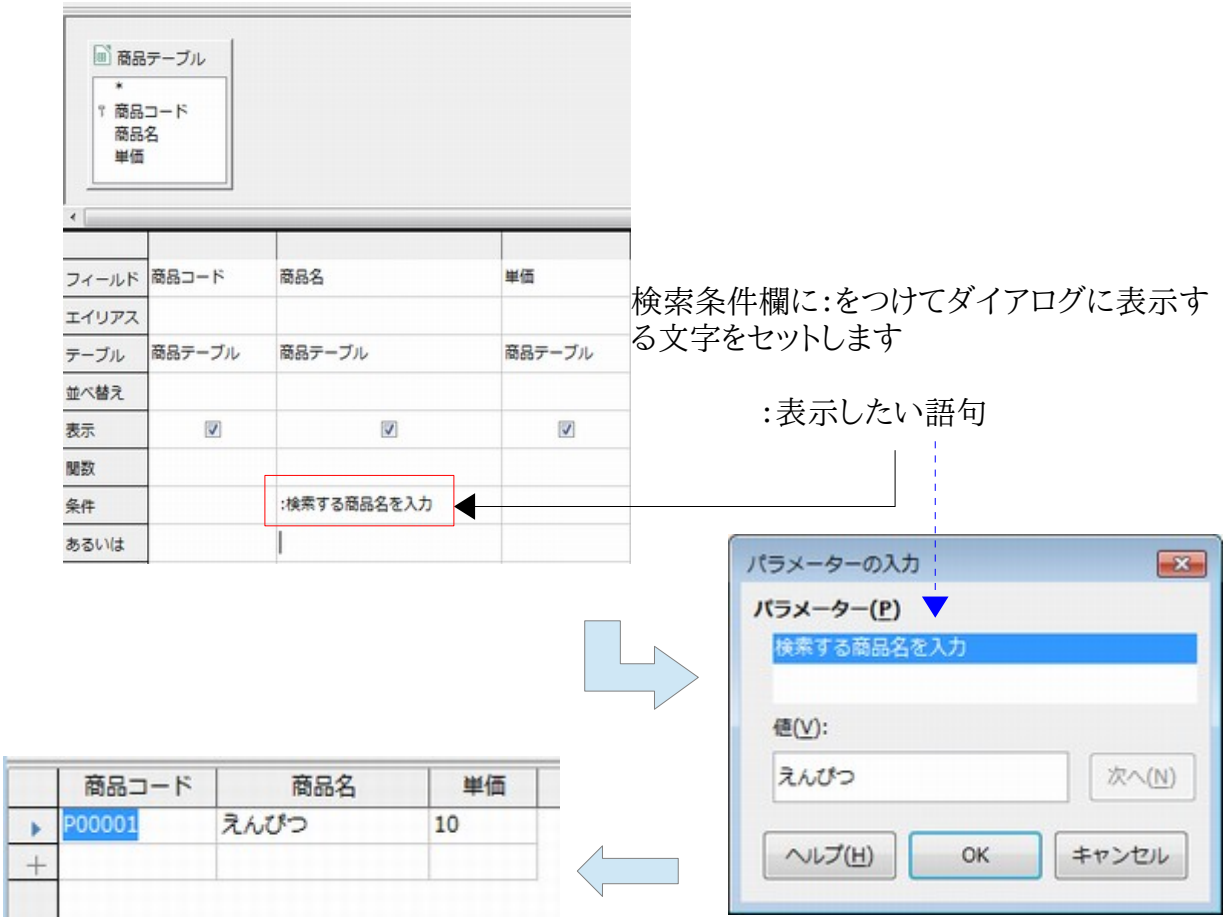
※全明細が必要な処理では必ず UNION ALL を利用して結合します

※ユニオンクエリーと SUM 関数と GROUP を利用したグループ化構文例

```
SELECT "テキスト", SUM( "整数" ) "合計"  FROM
（
SELECT "テキスト", "整数" FROM "テーブル 1"
UNION ALL
SELECT "テキスト", "整数" FROM "テーブル 2"
）
GROUP BY "テキスト"
```

13. 処理実行時に入力した条件でレコードを抽出する（パラメータークエリー）

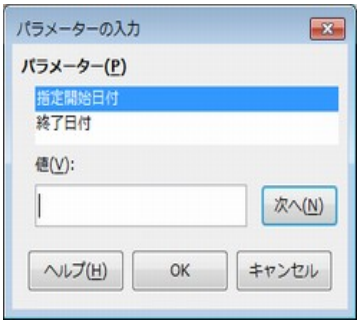
パラメータークエリーは、抽出条件を固定せず、クエリーを実行すると［パラメーターの入力］ダイアログが表示され、そのテキストボックスに入力した条件でレコードを抽出できるクエリーです。
毎回、異なる条件設定でレコードを抽出することができます



《Like 演算子による、あいまい検索例》
SELECT "商品コード", "商品名", "単価" FROM "商品テーブル" WHERE "商品名" LIKE '%' || :検索する商品名を入力 || '%'

LIKE '%' || :検索する商品名を入力 || '%'

《抽出条件を2つセットする検索例》
WHERE "生年月日" > :指定開始日付 AND "生年月日" < :終了日付



14. クエリーで利用する「フィルター演算子」

クエリー利用時の「フィルタリング条件」には、さまざまな処理演算子や SQL コマンドが利用できます。
関係演算子以外にも、クエリー処理を行う SQL コマンドが各種存在しており、BASE のコマンド内でこれらを使用した場合は、BASE が自動的に対応する SQL コマンドに変換します。

演算子	意味	条件が成立する場合
=	等しい	指定値とフィールド値が一致する。
<>	等しくない	指定値とフィールド値が一致しない。
>	大きい	指定値よりもフィールド値が大きい。
<	小さい	指定値よりもフィールド値が小さい。
>=	大きいか同等	指定値とフィールド値が同等か、その方が大きい。
<=	小さいか同等	指定値とフィールド値が同等か、その方が小さい。

BASE コマンド	SQL コマンド	意味	条件が成立する場合
IS EMPTY	IS NULL	空(NULL)	データフィールドが空です。
IS NOT EMPTY	IS NOT NULL	空でない	データフィールドが空ではない。
LIKE（任意の文字数のワイルドカード文字は* 1 文字のワイルドカード文字は？）	LIKE (任意の数の文字に対応する% プレースホルダー) 一文字のワイルドカード文字 _ ）	検索パターンと一致する	指定値がフィールド値に含まれている。ワイルドカード文字「*」を利用した「あいまい」検索を実施します。ワイルドカード文字「*」および「%」は複数の任意の文字です。LibreOffice の疑問符「?」や SQL クエリーの下線「_」は、任意の一文字を表します。
NOT LIKE	NOT LIKE	検索パターンと一致しない	指定値がフィールド値に含まれていない。
BETWEEN x AND y	BETWEEN x AND y	区間 [x,y] に存在する	データフィールドの値は x 値と y 値の間の値である。
NOT BETWEEN x AND y	NOT BETWEEN x AND y	区間[x,y]には存在しない	データフィールドの値は x 値と y 値の間の値ではない。
IN (a; b; c...) 全ての値リストで区切りに使っているセミコロンに注意してください。	IN (a, b, c...)	a, b, c..を含む	指定値 a, b, c,...のどれかがフィールド値に含まれている。多くの任意の値を対象にすることができ、クエリーの結果は論理「OR」で求められます。a, b, c... といった値は数値でも文字でもかまいません
NOT IN (a; b; c...)	NOT IN (a, b, c...)	a, b, c..を含まない	指定値 a, b, c,...のどれかがフィールド値に含まれていない。
= TRUE	= TRUE	「True」値	「True」値がフィールド値に含まれている。
= FALSE	= FALSE	「False」値	「False」値がフィールド値に含まれている。

《使用例》
```
SELECT * FROM "売上データ" WHERE "処理日" IS NULL
SELECT * FROM "売上データ" WHERE "処理日" IS NOT NULL
SELECT "商品名" FROM "商品マスタ" WHERE "商品名" LIKE '%PC%'
SELECT * FROM "売上データ" WHERE "処理日" BETWEEN '2019/04/01' AND '2019/06/10'
SELECT "商品 ID", "商品名", "グループ名" FROM "商品マスタ" WHERE "商品 ID" IN ( 5, 7, 9 )
```

≪Base マニュアル　～データ加工編～作成にあたって≫

データベースソフトは表計算やワープロと違い、画面を開いても具体的な使い方が判らない事や、標準ソフトとしてインストールされた PC が少なかったこと事も手伝って、Office ソフトの中でも特に敷居が高いと感じる存在だと思います。
　LibreOffice に標準ソフトとして装備されているデータベースソフト（Base）は、データベースに OSS-DB として実績のある Firebird を組込むことによって、デスクトップデータベースとして安定度の高い運用が可能となりました。
　本マニュアルはデータベースマニュアルにありがちな、構造理解やデータ管理手法等ではなく、主に日常業務で発生するデータ処理にスポットをあて、日頃の事務作業のツールの1つとして Base を位置づける内容に絞り込んで解説しました。
　いままでデータベースに触れたことが無いという人は、事務ツールの一つとして Base を実際に自分で操作し、事務処理の効率化を自ら体験してみてください。
　データベースソフトを通常業務処理に利用し、事務ツールとしての利点を最大限に生かすことができれば、Calc（表計算）や Writer（ワープロ）だけを利用していた頃と比べ、さらに高い業務の効率化を実現できることでしょう。
「全員の PC にデータベースソフトが無い」という理由だけでデータベースでの運用をあきらめ、Calc や Excel といった表計算ソフトでの運用を実施した事例はなかったでしょうか？
「やったことが無い」という理由だけで、システムによるデータ照合をあきらめ、手作業でのデータマッチングを実施したことはなかったでしょうか？
　データベースを具体的に操作することで、これまでのワープロや表計算ソフトでは難しかったデータ処理や加工といった応用についても具体的な処理方法がイメージできるようになるかもしれません。
　データ処理は習うよりも慣れることが重要です、Access 等のデスクトップデータベースを使ってこられた方は Access と Base の違いを認識し利活用の幅を広げてみて下さい。
　ワープロや表計算ソフトで実施可能な事務作業の効率化は、ほぼ完成されていると言って良い現在、現場には BI ツールやクラウド処理といったデータベース知識を要する技術導入に向けた、さらなる効率化が求められています。
データベースを利用すれば、業務とデータ処理の可視化が進み、更なる効率化への第一歩を踏み出す準備として大きく前進できます。
　データベースをワープロや表計算を超えた「第三の事務処理ツール」として位置づけ、本格的に業務で活用してみましょう。
きっと今までには無い視点でのデータ利活用の視点が養え、見えていなかった情報が見えるようになるはずです。
本マニュアルが皆様の OSS データベースソフト（Base）の利活用の一つとなれば幸いです。

【ありがとうございます】
本マニュアルで紹介した処理手順及び利用している SQL 構文は、LibreOffice 関連 Web サイトの掲載情報を参考に適宜修正・追加し作成しました。
各 Web サイトへ情報を提供してくださった方々、Web ページ作成者の皆様にこの場を借りて御礼申し上げます。

令和元年 10 月 1 日

JA 福岡市

※OSS（Open Source Software）・・・オープンソースソフトウェアとは、利用者の目的を問わずソースコードを使用、調査、再利用、修正、拡張、再配布が可能なソフトウェアの総称です。