# 目次

H	はじめに
	このレポートの対象者・・・・・・・・・・・・・・・・・・・・・・・・ 3
	学習方法・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ 4
学	習に必要な物
	このレポートの学習に必要な物・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
テ	ータベース設計の概要
	データベースとは······ 7
	プログラムとデータのファイルを分ける・・・・・・・・・・・・・・・・ 8
	テーブル等の名前の付け方・・・・・・11
テ	ータベースの新規作成
	データのファイルを作成する・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・12
	プログラムのファイルを作成する・・・・・・・・12
テ	ーブル
	テーブルの設計・・・・・・・18
	テーブルの作成・・・・・・・・20
	サンプルのテーブルを見てみる・・・・・・・・・・・・・・・22
	リレーションシップの作成・・・・・・・・・・・・・・・・・・・・・・・25
ク	エリー
	クエリーとは・・・・・・・・25
	クエリーの種類・・・・・・・・・・・・・・・・・・・・・・・・25
	選択クエリー・・・・・・・・・・・・・・・・・・・・・・・25
	クエリーの新規作成・・・・・・・・27
	サンプルのファイルをインポートして使う・・・・・・・・・・・・・・・・29
フ	<b>/</b> オーム
	フォームとは・・・・・・・・・・・・・・・・・・・・・・・・・・・・・35
	フォームの種類・・・・・・・・・・34
	フォームのプロパティ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	デザイン設計で使うメニュー・・・・・・・39
レ	ンポート
	レポートとは・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	レポートの種類・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	レポートのプロパティ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
7	・クロ
	マクロとは・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・

# 

## はじめに

#### ●このレポートの対象者

最近はワードとエクセルは誰でも使えます、しかしアクセスを使っている人は、なかなかいません。データベースは難しいに違いないと、ほとんどの人が思っているようです。確かに難しいです。プログラム言語を学習したことのない人には、とんでもなく難しいです。VisualBasic で扱うプログラム言語を使いますし、奥が深いです。勉強することは山ほどあります。私もアクセスの参考書を10冊以上買っていろいろ勉強しました。

しかしながら、通常の事務処理をするプログラムを作る時には、アクセスの複雑な機能の少しの部分を使うだけで、十分使用に耐えられるプログラムが作れるのです。アクセスのすべての機能を知る必要はないのです。必要な部分だけ知っておけばいいのです。

ただ、そのことを教えてくれる書籍はありませんし、この機能は覚えなくてもいいと、はっきり書いてある書籍もありません。アクセスを学習する人向けに、さしさわりのない書籍ばかりです。

ここでは、次のような方を対象にアクセスを使ってのプログラミングを説明しています。

- ① エクセルでいろいろ頑張ってるけどデータベースでないと駄目なような気がするけどアクセスが難 しすぎて出来ないのに部長にアクセスで作れと毎日怒られて、悩んで頭が禿げてしまったあなた
- ② 社員が各自、独自の資料を作っていて月末に、集計するのが大変で、残業ばかりしているため、帰りが遅いから奥様やお子様から相手にしてもらえず、孤立しているあなた(統一したプログラムをアクセスで作ることができれば、簡単なのに)
- ③ 会社の基幹業務をソフト会社に作ってもらって、きっちり管理して売上をガンガン伸ばして頑張ろうと思い、見積を出してもらったが、金額をみて、目が飛び出してしまったあなた(アクセスを使って、自分で作ることが出来れば・・・)
- ④ 現在使っている、さほどボリュームのない以前外注で作った簡単なソフトが使いにくいので、少し変更しようと思い、ソフト会社に見積を出してもらったら、高すぎるので、文句を言ったら「ずいぶん昔のソフトで担当者も辞めてしまい変更が難しい」とか言われて、怒って業者と喧嘩してしまったあなた(アクセスで最初から作っても、簡単に出来そうな気がするけど、自分で出来ない)

以上のような、本人にとっては冗談ではすまないあたなにお送りします

#### ●学習方法

ここでは、アクセスの入門書のような、アクセスの全てを最初から学習するような進め方をしていません。通常の入門書は、データベースの概念をおぼえて、次にテーブルについて詳しく説明しています、 そのあとクエリーに進んでいくといった具合です。

そんな事をしていたら、自分の作りたいプログラムはいつまでたっても形になりません。

5歳の子供が日本語の文法を完璧に理解してるでしょうか、それでも話はできます。話が出来るように なってから、難しい文法を勉強した方が効率がよいと思います。

ある程度プログラムが作れるようになったら、その先の複雑な処理方法や使い方をネットで検索すれば、 簡単に理解できたりします。ですからあまり細かいところで、なぜこんな事をしているのだろうか?と 悩まないでください。そうするものなんだということで、前に進んでください。後日、理解が深まれば 自分なりにその部分を変更して使えるようになります。

目標はあくまでも、短期間で会社の基幹業務のデータベースを作成することです。プログラマーになるためにコンピュータ言語を学習するのではないのです。プログラムを綺麗に書く作法とか、少しでも早く動くようなプログラミングも、プログラムのサイズも、いっさい気にしません。結果さえ良ければいいのです。

ただし、使いやすい設計、予期せぬキーを押された時の対処などはしっかりやっておかなければいけません、自分だけが使うプログラムではないからです。入力は別の人が行うのが前提です。誰が入力しても、使用に耐えられるように作らなければいけません。

初めてアクセスを使う人が、入門書を手にして、プログラムを上手に作りあげるのはとても難しい事です。私が初めてアクセスの学習をした時は、どうせやるなら徹底的に勉強しようと思い、初級、中級、上級と3部作になっている本を買ってきてテーブルの作り方から順番に読んでいきました・・・・とても時間がかかりましたし、いざプログラムを作る時に何をどうしていいか、よくわかりませんでした。次にサンプルプログラムのCDが付いている本を買ってきて、それを参考にしてプログラムを作ってみました、これは少し役に立ちました。でもそのプログラムは入力している途中になぜか止まってしまったり、間違えて違うキーを押した時に登録されてしまったりと、改善の余地がかなりありました。自分なりに時間をかけて試行錯誤を繰り返して、少しずつ使いやすく、誰にでも入力できるものになっていきました。

たとえば、大手ソフトメーカーの販売している給与計算をみてみると、必須の入力項目が入力されていなければ、必ず「入力してください」というメッセージが表示されますし、入力が完全に終わらない限り、別の作業に移れないようにもしています。

ここでは、改善の余地のない完成されたプログラム(今のところ苦情は出ていません)をそのまま流 用して、あなたの必要な入力項目に差し替えるだけで、プログラミング出来るように、その作業方法を 中心に進めていきます。 ただし、ある程度のウインドウズの理解、ワードとエクセルくらいは、普通に使える知識がないと理解できないです。

アクセスの基本的な細かい説明はできませんので、項目別に解説してある、リファレンスマニュアル のような本があれば良いかもしれません。

このような趣旨から、プログラマーを目指す方は、これを見ないでください。全然役に立たないと思います。あくまでも仕事で一刻も早く会社の基幹業務を作らなければいけない危機的状況(そんな人居ないかも・・)に置かれたあなたを対象にしています

# 学習に必要な物

このレポートの学習に必要な物

●インターネットが出来る環境

インターネットで Access 実践入門 next というホームページを見ていただく必要があります。 このサイトからサンプルデータをダウンロードしなければいけませんし、主要な解説もここにあります。

サイトのアドレス http://www.msactrynext.com/

●マイクロソフトアクセスの 2007 以上のバージョンのプログラム

アクセスの製品が必要です。バージョンは 2007 以上です。 サンプルプログラムは 2016 で作成しています。Access 2003 以前のバージョンでは、動作しません。

この解説で使っている画面は、Access2016の画面です。

# データベース設計の概要

#### ●データベースとは?

家庭にはアドレス帳、年賀状、お歳暮を贈った時の送り状の控え、子供の学校の緊急時の連絡網の表、 町内会の名簿、その他生活に必要ないろんな情報があります。それが、家の中の何処にあるか解らない と、とても困ります。

レターケースを買ってきて、引き出しにシールで何が入っているか書いて、種類毎に整理して入れて おかなければいけません。こうしておけば、いつでも必要な物が取り出せます。

このように種類毎に分けてデータを蓄積しておいて、必要に応じてデータを取り出して、分析できるようにしてある物がデータベースです

アクセスは、次のようなオブジェクトで構成されています

- ・テーブルデータです(種類毎に分けられたもの)
- ・クエリー 作業に必要なデータを抽出、並び替え等をしてフォームやレポートで必要な項目を整理しておく、 中間的なデータの集まり
- フォームデータを入力したり、表示するための表
- レポートデータを印刷するもの
- ・マクロ一連の作業を登録しておくもの
- ・モジュール作業をプログラムで記述するもの

ずいぶん簡単に説明していますが、アクセスの学習を始める前にそれぞれを理解する必要はありません。サンプルを見ていくうちに、こんな時にこのように使うのだということを感じていただければそれでいいと思います。

### ●プログラムとデータを別ファイルにする

最近はパソコンが2台ある場合は、簡単にネットワークで繋げることが出来ます。アクセスはネットワーク上で使えるデータベースです。また、データベースの保守を考えても、データとプログラムは別にしておいたほうが便利です。プログラムの修正中でも古いプログラムで入力できますし、プログラムの差し替えも簡単です。

アクセスにはリンクテーブルという概念があります。それを利用します。

外部のデータベースをプログラムを、テーブルに貼り付けることができます。あたかも今開いているファイルの中のテーブルのように扱うことができます(フィールドの追加等、テーブルデザインを変更することは出来ません。データのファイルを直接開けば出来ます)

データのファイルは、実際に保存されるテーブルのみです。作業用の一時テーブルはプログラムの方に 入ります。

プログラムのファイルは、上記以外の作業用のテーブル、クエリー、フォーム、レポート、マクロ、モジュールです。

サンプルの住所録を見てみます。

住所録 Data.accdb です。



デーブルが5つあるだけのデータベースです。 クエリー、フォーム等はありません。 以下は、住所録 Pro.accdb です。

データのファイルにあったデーブルが5つありますが、矢印が付いています。リンクしているテーブルであることを示しています。

to 誕生日 ca は、作業用のテーブルで、実際のデータは入らないので、プログラム側にあります。

#### ・テーブル



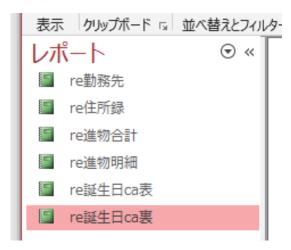
#### ・クエリー



### ・フォーム



#### ・レポート



#### マクロ



Access 実践入門 next では、マクロは使いません。

メニューの表示のために使っているものだけです。これは、リボンを変更して Access のメニューを表示しないようにしています。

マクロを使うと、プログラムの流れが解りにくくなってしまいます。なので、このサイトでは、すべてプロシージャを使って、マクロのコマンド(Docmd)を使っています.

#### ・モジュール



#### ●テーブル等の名前の付け方

これからプログラムを作りますが、いろんな名称を付けることになります。学習書には、名称はすべて英語でと書いてあるものもありますが、私たちはプログラマーではないので、日本語で名称を付けます、アクセスは日本語版しか使いませんし、英語ですと、ぱっと見て、イメージが出来ませんから。

ただし、付け方には注意が必要です。それを以下に説明します。守らないとプログラムが動かないわけではありません。ファイルの参照時のミスの防止等に役に立つからです。

フォーム、フォームの中のフィールド、クエリー等の名前に全角カタカナ、全角英文字等、日本語特 有の全角と半角の両方に存在する文字を使わないようにしましょう。

アルファベットなら、漢字モードを使用しないで、CAPSをはずした小文字で。

全角文字の場合は、カタカナ、英字以外の漢字とひらがなのみで。

ようするに日本語特有の全角と半角の両方に存在するものを使わないということです。

Access のバージョンアップの時に、モジュールでうまく変換できなかったり、フォントによって開発中に全角なのか半角なのか解りにくかったりします。

たとえば、

「住所録コード」ではなく、「住所録番号」に

「フリガナ」ではなく、「ふりがな」に

ただし、入力画面に表示されるラベルの表題は「フリガナ」の方が解りやすいでしょうし、ラベルの表題には、使用しても問題ありません。

サンプルプログラムでも、その使い分けをしています。

作成したテーブル等の名称には、先頭にその種類がわかるな文字を付けておきましょう。 でないと、それが、フォームなのか、クエリーなのか、解らなくて困るからです。

サンプルでは次のように決めています。

テーブル:ta○○

クエリー:qu○○

 $7x-\Delta: fo\bigcirc\bigcirc$ 

レポート:re〇〇

マクロ : ma〇〇

説明は、これくらいにして、プログラムの作成に移ります。

# データベースの新規作成手順

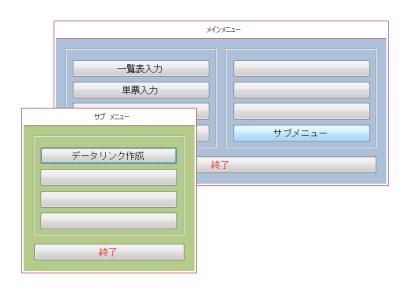
#### ●スタートキットを利用する

Access 実践入門 next では、基本となるデータ用ファイルとプログラム用のファイルを用意しています。

それを使って自分の作りたいデータベースを作成してください。

基本となるフォームが用意されていますので、それをコピー、貼り付けで複製して、 それを手直しして作ることを繰り返します。

・基本のメニューとサブメニュー



基本の一覧表入力



# 基本の単票入力



その他、レポート、モジュールも用意しています 項目名などを変更して、すぐに使えるようになっています 慣れてくれば、色などのデザインを好きな色に変更することで、オリジナルのデータベースが 出来上がります

#### ●データのファイルを作成する

まず、プログラムのファイルとデータのファイルを入れるフォルダを作りましょう。 その中にスタートキットのプログラムファイルとデータファイルをコピーします (startkitd.accdb、startkitp.accdb)

では、データのファイルを作成します。

startkitd.accdb の名前を変更します

ファイル名は何でもいいですが、データであることが解るようにしておきます。

○○データ.accdb、○○data.accdb とかでいいと思います。

データはこれでできあがりです。テーブルの作成は後で・・・

#### ●プログラムのファイルを作成する

次に、プログラムのファイルを作成します。

startkitp.accdb の名前を変更します

プログラムであることが解るようにしておきます。

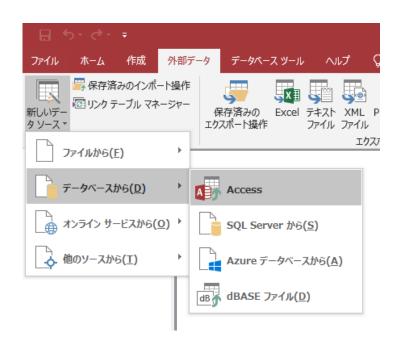
○○プログラム.accdb、○○P.accdb とかでいいと思います。

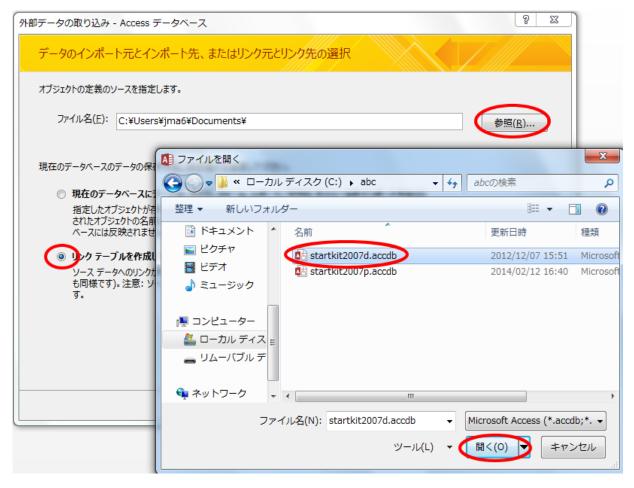
データのファイル名が変わりましたので、データのリンクを変更しなければいけません プログラムのファイルを Shift キーを押した状態で起動してみてください

(Shift キーを押さないとメニューが出てしまいます)

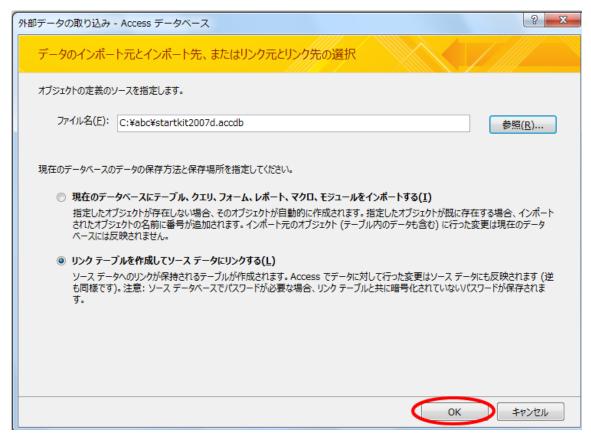


テーブルが4つありますが、右クリックで全部削除します (実際にテーブルが削除されるわけではなく、リンクが解除されるだけです) 全部消えたら、上のメニューから、外部データの新しいデータソースをクリック データベースから、Access をクリック

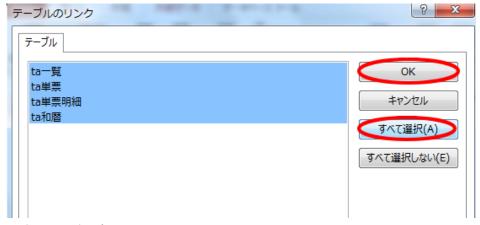




- ・リンクテーブルを作成・・・を選択
- 参照ボタンをクリック
- ・作成したデータ用のファイル(名前が変更されている)を選択して
- 開くボタンをクリック



ファイル名が入りますので、OKボタンをクリック



- すべて選択ボタンをクリック
- OKボタンをクリック



新しいデータ用のファイルにテーブルがリンクされました

それでは、一度 Access を閉じて、もう一度 Shift キーを押さないで起動してみましょう メニューが出れば O K  $\overline{C}$  す

これで、プログラムを作り始める準備はできました。

# テーブル

#### ●テーブルの設計

設計の段階で、この項目も作っておいた方がいいし、この項目もと・・・どんどん増えてしまいがちですが、運用の段階で結局入力するのが大変で使わなくなってしまう項目がたくさんあったりします。 実際に入力可能な項目なのかどうか考えて、不必要なものは設計の段階で省いておきましょう。摘要とか備考とかを作っておいて、そこに入力するようにすれば、だいたい処理可能です。将来のことも考えて、最初にじっくりと考えておいた方がいいでしょう。

だからといって、作ってみないと解らないこともたくさんあります。会社の基幹業務をいきなり作るのではなく、社員名簿あたりから作ってみてはどうでしょう。慣れてくるとテーブルの設計はすぐに出来るようになります。

しかしながら、入力項目が追加になっても、たいしたことはありません。項目が簡単に追加できないようでは役に立たないデータベースです。慣れれば、項目の追加などは、たいした作業ではありません。

設計で大事なことは、何もかも1つのテーブルに入れないで、用途別に分けることです。データベースに慣れていない人が陥りやすいことです。データベースの特性を生かさなければいけません。

たとえば売上履歴付きの顧客管理を作る場合、データとして、氏名、住所、家族の情報、売上の情報、 入会時のアンケートの情報等があったとします。

まず、氏名、住所等の基本的なものは、1人に対して1つの情報となりますが、家族の情報、売上の情報は1人に対して複数の情報が必要になります、たとえば奥さんと子供が2人居た場合、1人に対して3人の家族の情報が必要です。この時に家族の名前や生年月日といった家族の情報を多めに、5項目分フィールドを準備しても良いのですが、そうすると生年月日で検索する時、5項目分順番に見ていかなければいけません。(これは面倒なことなのです)

氏名や住所とは別にして、家族の情報を別のテーブルに分けておくと、データベースの機能を使って簡単に検索することが出来ます。売上についても同じことが言えます、家族は数人しか居ないですが、売上に関しては、将来にわたって何件発生するか解りません。それを氏名、住所と同じテーブルに入れることはできません。

また、アンケートについても、何十項目もあった場合、氏名、住所とは別にしておいた方が、後々解りやすいでしょう。テーブルのフィールド項目は255項目まで作れますが、分類別にテーブルを分けて数十個程度までにしておいた方がいいでしょう。

すべての項目を1つのテーブルに入れてしまっている場合

	> / · (= / • / • C O O O O O O C C C C
<b>■ テーブル1</b>	
フィールド名	データ型
₽▶ 顧客番号	短いテキスト
氏名	短いテキスト
郵便番号	短いテキスト
住所	短いテキスト
電話番号	短いテキスト
携帯電話	短いテキスト
生年月日	日付/時刻型
趣味	短いテキスト
備考	短いテキスト
家族氏名01	短いテキスト
家族性年月日01	日付/時刻型
家族備考O1	短いテキスト
家族氏名02	短いテキスト
家族性年月日02	日付/時刻型
家族備考02	短いテキスト
家族氏名03	短いテキスト
家族性年月日03	日付/時刻型
家族備考03	短いテキスト
入会情報01	短いテキスト
入会情報02	短いテキスト
入会情報03	短いテキスト
入会情報04	短いテキスト
─ 入会情報05	短いテキスト
■ 入会情報06	短いテキスト
- 売上日01	日付/時刻型
売上金額01	数値型
売上品目01	短いテキスト
売上日02	日付/時刻型
売上金額02	数値型
売上品目02	短いテキスト

# それぞれを、分けてテーブルを作った場合

≡ テーブル2		
	フィールド名	データ型
8▶	顧客番号	短いテキスト
	氏名	短いテキスト
	郵便番号	短いテキスト
	住所	短いテキスト
	電話番号	短いテキスト
	携帯電話	短いテキスト
	生年月日	日付/時刻型
	趣味	短いテキスト
	備考	短いテキスト

□ テープル3\		
フィールド名	データ型	
❷ 顧客番号	短いテキスト	
家族氏名	短いテキスト	
家族性年月日	日付/時刻型	
家族備考	短いテキスト	

Ⅲ テープル4		
/ フィールド名	データ型	
	短いテキスト	
入会情報01	短いテキスト	
入会情報02	短いテキスト	
入会情報03	短いテキスト	
入会情報04	短いテキスト	
入会情報05	短いテキスト	
入会情報06	短いテキスト	

≡ テーブル5		
4	フィールド 名	データ型
8▶	顧客番号	短いテキスト
	売上日	日付/時刻型
	売上金額	数値型
	売上品目	短いテキスト

この場合、すべてのテーブルに、統一された顧客番号が存在しています。

この番号で、誰の情報であるかが解ります。

こちらの方が管理しやすいです

#### ●テーブルの作成

それでは、テーブルを作成します。

#### フィールド名

フィールド名は、データベース設計の概要で述べた名前の付け方に注意して決めてください。 あまり長い名前を付けると、使いにくくなりますが、あまり簡単でも駄目です。

たとえば、個人の情報と会社の情報が、別のテーブルにあったとします。

その場合は、「住所」というフィールドをテーブルが違うので、両方に作ることはできますが、条件を指定する時などで、[住所]だけでは、どちらのテーブルの[住所]か解らないためにエラーが出ることがあります。(どのテーブルの住所ということを細かく指定しないといけない)

そのことからも、データファイルの中でたくさんのテーブルがありますが、同じフィールド名を使わないようにしましょう。

たとえば、上記の例では、個人の情報のテーブルでは[個人氏名]、[個人住所]・・・・、会社の情報の テーブルでは[会社名]、[会社住所]・・・・、といった名前の付け方をしましょう。

#### データ型

データ型には、次のようなものがあります

- ① 短いテキスト型(数値、日付以外のもの)
- ② 数值型 (数值)
- ③ 日付/時刻型(日付を入れる)
- ④ オートナンバー型 (入力する人に見せない内部で持つ番号として使用)

他にも長いテキストや Yes/No 型等がありますが、私は通常、上記4つしか使いません

#### ① 短いテキスト型

数値と日付以外は全て、この型を使います。255文字まで入力できますので、備考などの文字列入力でも問題なく使えます。

たとえば、顧客管理で顧客番号をテキスト型で3文字に設定したとします。その時は必ず3文字で入力するようにします。001、002、011、のようになります。桁数を揃えておかないと並び替えた時に数字の順序がおかしくなります。それがいやなら、数値型にしましょう。半角、全角等の入力の区別は、入力するフォームで制御できます。半角と全角の区別なく文字数を数えます。アイウエオ、あいうえお、いずれも5文字です。半角で入力しても文字がたくさん入力できるわけではありません。

#### ② 数值型

数字をあつかう場合はすべてこの型を使います。

小数点以下の桁を使う場合は、テーブルのデザインの下の設定で、フィールドサイズを倍精度浮動小数 点型にします、小数点を使わない場合は、長整数型にしておきます。

フィールドサイズで、その他の型は使わなくてもいいと思います。

#### ③ 日付/時刻型

日付をあつかう場合は、すべてこの型を使います。

#### ④ オートナンバー型

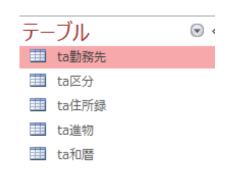
たとえば、顧客の情報を入力するテーブルで、顧客番号は入力された順番にデータベースに番号付けをまかせる場合等に、この型を使います。そうすれば、1、2、3・・、と付けてくれます。そのかわり削除された番号は、空き番号になります。綺麗に順番通りには付きません。私たちが番号を制御することはできません。

ここでの顧客番号は、入力者にとって必要ないものであれば、オートナンバー型でいいと思います。 検索する時はフリガナや名前で探せばいいですし、一覧表の印刷も、フリガナの50音順でいいのです から・・・

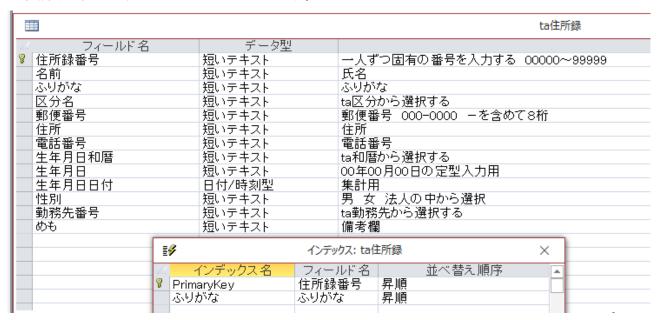
また、主キーの必要ないテーブルでは、id 番号として、この型でフィールドを作っています。そうしておくと、作られた順番で並んでいるので、保守しやすくなります。(テーブルには必ず主キーを作りましょう)

#### ●サンプルのテーブルを見てみる

住所録 Data.accdb を見てみます 5つのテーブルがあります。



ta 住所録は次のようなデザインになっています。



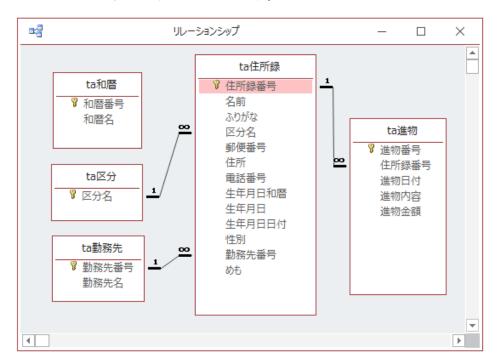
テーブルの内容、作り方は、ホームページを参照してください

#### ●リレーションシップの作成

上のメニューより、リレーションシップを選びます



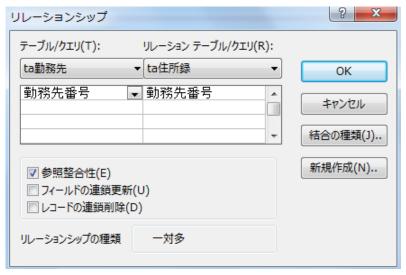
サンプルでは次のようになっています。



ta 住所録の勤務先番号から ta 勤務先の勤務先番号へ線が繋がっています。 その線にマウスポインタを合わせて、右クリックすると次の窓が開きます

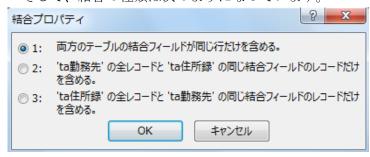


リレーションシップの編集をします。



参照整合性にチェックが入っています。

そして、結合の種類は次のようになっています。



このあたりの言い回しは難しいですが、出来上がった線を見れば何となく解りますでしょ? ta 勤務先は1つで、ta 住所録の方は、もし同じ会社の人が居たら2 個、3 個となります。 なので、1 と $\infty$ になっています

参照整合性とは、ta 勤務先のデータを削除しようとしたとき、ta 住所録にその勤務先番号の人が登録されていれば、参照できなくなるので、削除できませんという、メッセージが出て、削除できません。プログラムの中では、そんなエラーが出ないように処理していますが、自分が保守作業で、テーブルを直接開いて作業しているときなどに、必要なデータが間違って削除され、データの矛盾を起こさないためにも必要です。

テーブルを作ってみて、リレーションシップを作成してみてください。

どちらかが主キーになっているものが繋がります。

ドラッグ、ドロップで線が繋がります

リレーションシップの窓は最初は、空ですので、作ったテーブルを全部貼り付けて、線を設定してください。線の付かないテーブルも貼っておきます。

そうすることで、データ全体の構造が解ります。

これで、データの accdb が完成しました。

## クエリー

#### ●クエリーとは?

クエリーとは、「問い合わせる」という意味らしいです。何か聞き慣れない言葉ですね~。 テーブルを作成する時に、種類毎に分けて作成するように書きました。フォームやレポートには、何を 基にしてフォームやレポートを作るかという指定(レコードソースと言います)があります。そこで指 定されるのが通常クエリーとなります。テーブルをいくつか合わせて、1つのデータの集合体を作り、 それを基に、フォームやレポートを作ることになります。 使っているうちに、クエリーとはこういう物だということが解ってきます。

#### ●クエリーの種類

クエリーには次のような種類のものがあります

- ・選択クエリー 条件にあったレコードを抽出したり、並び替えたりする。 一番、使用頻度の高いクエリーです。
- ・更新クエリーそのクエリーに入っているレコードを一括で変更するデータを一度に変更する時に使います
- ・追加クエリーデータにレコードを追加する
- ・削除クエリー データのレコードを削除する

その他、クロス集計クエリー等がありますが、私は上記以外のクエリーは使いません

#### ●選択クエリー

ほとんどの場合は選択クエリーを使います。この選択クエリーの理解が、アクセスの理解の重要な部分です。その他のクエリーはサンプルの中で出てくるので、その時に使い方を見てください。

選択クエリーのサンプルをいくつか見てみます。

住所録の一覧表に使うクエリーです。

ta 住所録と ta 勤務先をクエリーに含めることで、勤務先の名称も同時に表示できます。 ふりがな順になります。



一覧表から、誰かを選択したときの編集画面に使います。 抽出条件に、一覧表で選択された住所録番号が設定してあります。



ホームページを参照して、それぞれの使い方を見て、真似をしてクエリーを作ってみてください。

クエリーは、フォームやレポートのレコードソースとして作りますので、フォームを作る上で、どの テーブルの、どのフィールドが必要か考えながら作ります。

#### ●クエリーの新規作成

上のメニューから、作成タグのクエリデザインをクリック

ウィザードは使いません。フォームやレポートにもウイザードはありますが、使わないでください。 ウイザードを使うと、作成手順が覚えられませんし、使いにくいものが出来上がります。



#### 次の画面になります



Ta 住所録を選んで、追加ボタンをクリックして、この窓を閉じます (テーブルを複数使う場合は、選択して追加ボタンを繰り返して閉じます) 住所録番号の上のアスタリスク (\*) を下にドロップした場合は、フィールド名が ta 住所録\*となります。クエリーを開いたとき、フィールド全部が表示されます。

その右のふりがなは、並び替えを設定するめににドロップしています、表示のチェックはオフです (オフにしていないと、ふりがなが2列出てしまいます)



Shift キーで全フィールドを選択してドロップした場合です ふりながのところに、並び替えを設定しています

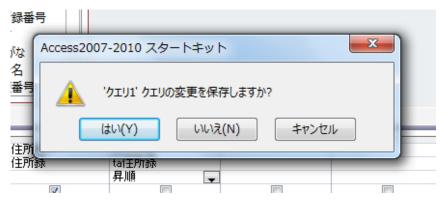


上記2つの方法で、表示結果は同じですが、上の方法を使ってください。

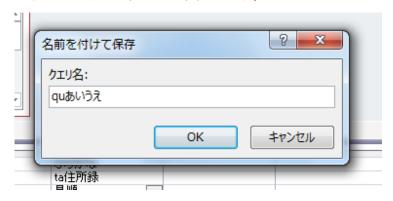
違うのは、プログラム作成中に後から気がついて、フィールドを追加したような場合、クエリを修正しなくてもいいのが、上の方法です

下の場合は、追加したフィールドをドロップしておかなければいけません。

クエリーのデザインを閉じるときに、保存を聞いてきます。



名前を登録して、OKで、完了です。



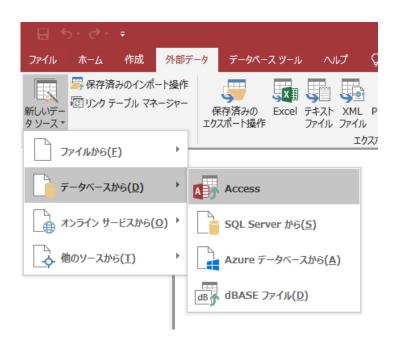
#### ●サンプルのファイルをインポートして使う

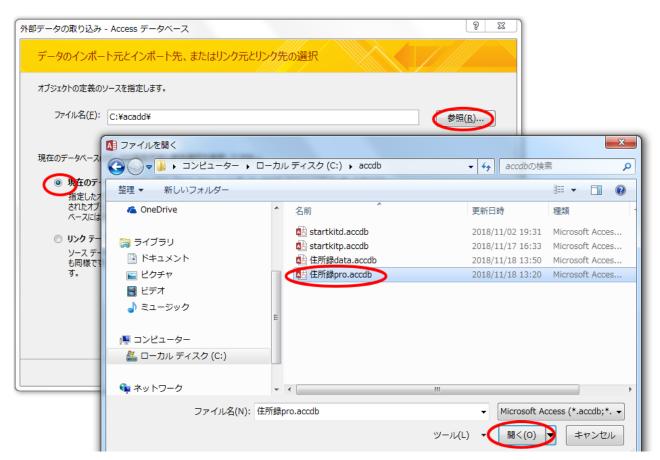
クエリーはテーブルが違うので、そのままは使えませんが、フォームやレポートは、最初からデザインを作るのは大変なので、自分のプログラムのファイルにインポートして使いましょう。

まず、サンプルデータをダウンロードしておかなければいけません。ホームページから、住所録をダウンロードしておいてください。

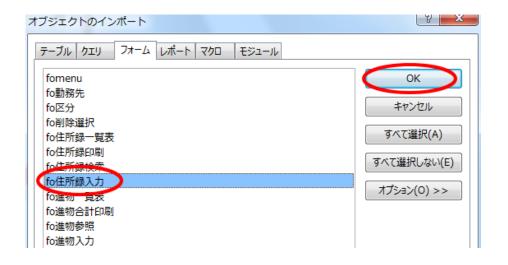
スタートキットに、住所録からフォームをインポートしてみましょう

上のメニューから、外部データの新しいデータソースをクリック データベースから、Access をクリック



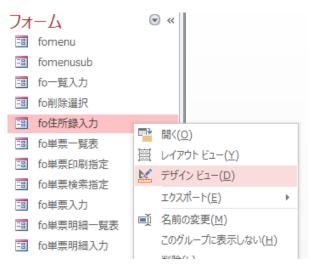


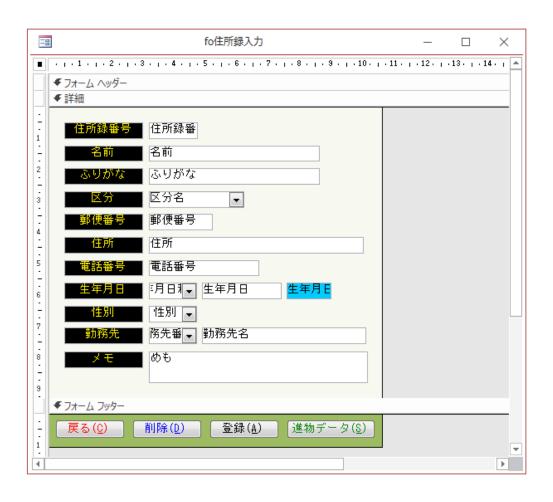
インポートしたいオブジェクトがある、データベースを選択します



フォームの fo 住所録入力を選択して、OK

インポートされましたので、デザインを見てみましょう 右クリックのデザインビュー





このように、サンプルの中の使えそうなものを、どんどんインポートして、コピーしたり、修正したり、 そのままオブジェクト名を変更して利用したりしてください。

特に、フォームやレポートはデザインを最初から作るのは大変ですから・・・

#### フォーム

#### ●フォームとは?

フォームは、テーブルにデータを登録するための入力画面、テーブルのデータを表示するための確認 画面、データを検索するときの条件を入力するための指定画面等、いろんな使い方をします。

逆に言うと、いろんな使い方が出来るということは、とても設定が複雑だということです。ここでは、フォームの機能について、最初から説明することをしません。フォームはスタートキットプログラムから、使えそうなものを、インポートして修正しながら作っていきます。新規に作ることはしません。新規に入力フォームを作っていくには膨大な設定をしなくてはいけません。最初は出来上がっているものを修正して使う方が、早くプログラムを作ることが出来ます。

ある程度慣れてくると、自分なりに色とか、形を統一していくことで、オリジナルのデザインが決まってきます。それが自分の財産になり、それをコピーしてどんどんフォームを増やしていけます。

#### ●フォームの種類

フォームはデータの表示のしかたの違いで、次の2つに分かれます。

単票フォーム

単票フォームはさらに2つに分類できます。

1つのデータレコードを1つの画面に表示したり、入力したりするフォームデータを持たない、案内板や指定画面としてのフォーム

・帳票フォーム

複数のデータレコードを一覧表の形で、1つの画面に表示したり、入力したりするフォーム

簡単にいうと、上記3種類しかありません。

単票入力

番号	
名前	
ふりがな	
一覧より	•
文字	
数值	
生年月日	•
性別	▼
日付	
備考	
戻る( <u>C</u> )	

データを持たない指定画面の単票フォーム

検索指定

名前の一部分	
フリガナの一部分	
検索	戻る

複数のデータレコードを一覧表の形で表示する帳票フォーム

単票一覧表

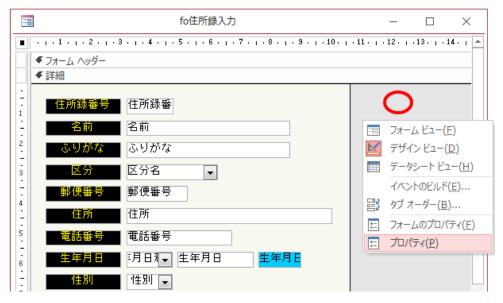


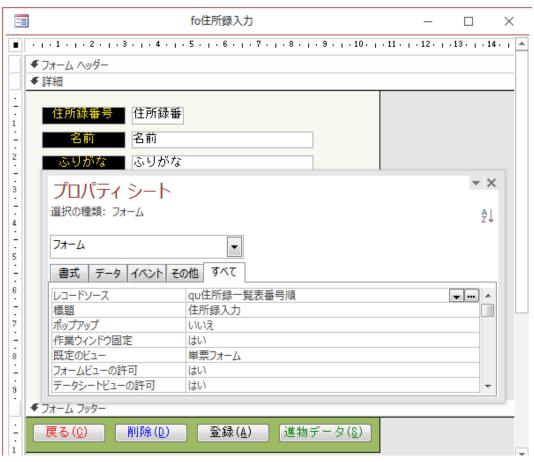
#### ●フォームのプロパティ

フォームには各部分毎にプロパティがあります。(各種設定) それを少し紹介します。

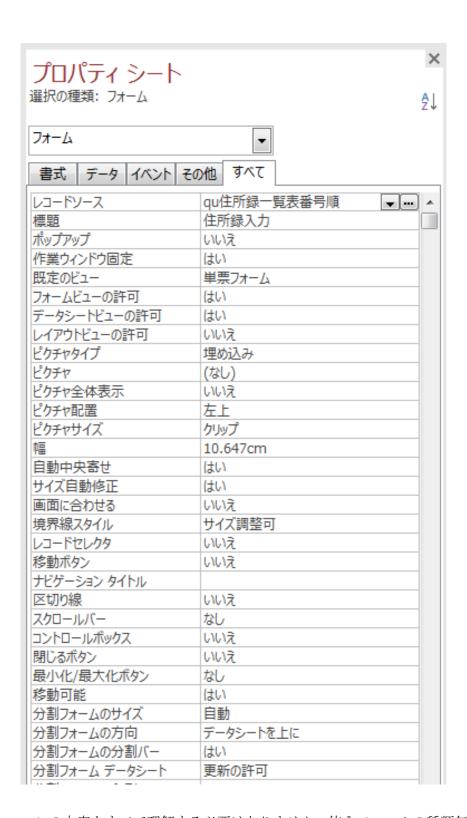
#### フォーム全体のプロパティ

デザイン画面の何もないところで右クリックで、プロパティをクリックすると表示されます



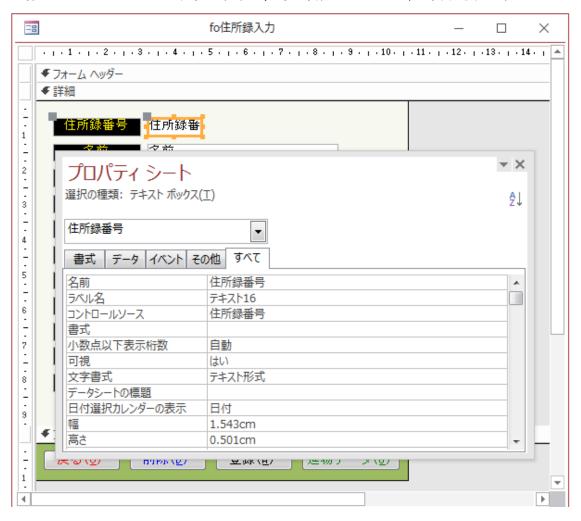


プロパティの内容は、次のようになっています。これでも一部分です。



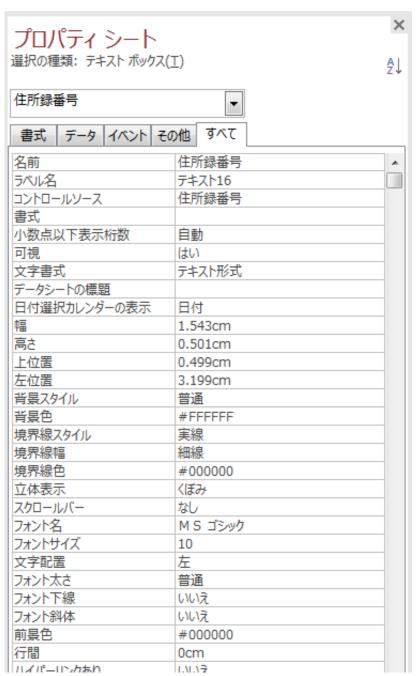
この内容をすべて理解する必要はありません。使うフォームの種類毎に、必要な部分だけ覚えればいいですし、種類毎に設定のしかたは、だいたい決まっていますので、フォームをコピーして使えば、ほとんど変更する必要もありません。

コントロールのプロパティ 各コントロールの上でクリックすると、その項目のプロパティが表示されます



ここでは、名前のコントロールになります。

プロパティの内容は、次のようになっています。これでも一部分です。



入力が漢字かどうか、コントロールの色、形、場所、入力の順序等、いろいろありますが、コントロールもコピーして使いますので、コントロールソースとIME入力モードを変更するくらいです。

フォームの細かい説明は、ホームページを参照してください。

### ●デザイン設計で使うメニュー

コントロールをもう1つコピーしたり、コントロールを綺麗に並べたり、デザインの細かい部分を修正していくために、使うメニューを紹介しておきます。

・コントロールと表題 (ラベル) をもう1つ作る白いのがテキストボックス (住所録番号を入力する) です

濃いグレーのものがラベル (テキストボックスが何かを示すもの、テキストボックスに必ず必要なものではない)です

コピーは、四角の右上でクリックしたまま、マウスを移動して、2つのコントロールを選択します

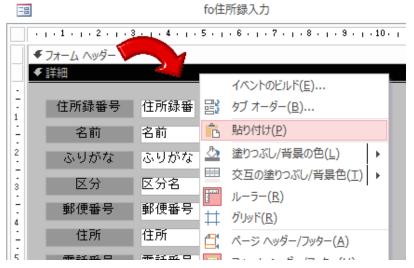
==	3	fo住所録入力		
	1 2	3 <b>. <sub>1</sub> . 4 . <sub>1</sub> . 5 . <sub>1</sub></b> . 6 . <sub>1</sub> . 7 . <sub>1</sub> . 8 . <sub>1</sub> . 9 .		
	<b>♥</b> フォーム ヘッダー			
	▼詳細			
1	住所録番号一	住所録番		
-	名前	名前		
2	ふりがな	ふりがな		
3	区分	区分名    ▼		

-8		fo住所録入力
	1 - 1 - 1 - 2 - 1 -	3 . 1 . 4 . 1 . 5 . 1 . 6 . 1 . 7 . 1 . 8 . 1 .
	フォーム ヘッダー 詳細	
-	住所錄番号	住所録番
<del>-</del>	名前	名前
2	ふりがな	ふりがな

住所録番号の、コントロールと表題の2つが選択されました



右クリックで、メニューからコピーを選択

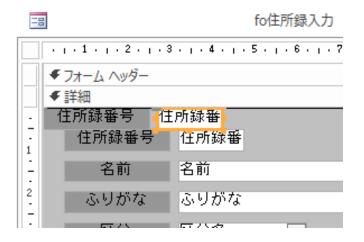


詳細を左クリックして選択後、

右クリックで、メニューから貼り付け

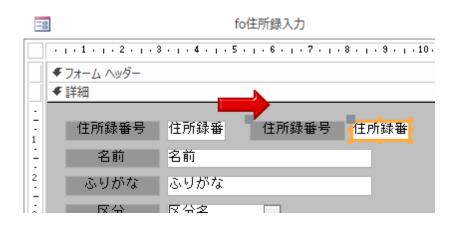
(詳細をクリックすることで、フォームの詳細部分の下に貼り付けられる)

(貼り付けたい場所の上のコントロールを選択後に貼り付けると、その下に貼り付けられます) (いろいろ試してみてください)



すぐ下に同じものが貼り付けられました。

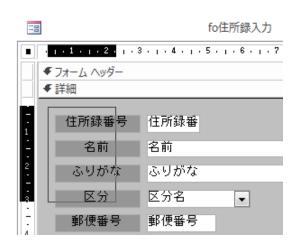
貼り付けられた、コントロールの右下へマウスポインタを移動して、クリックしたままでマウスを移動して、右下に移動できます。

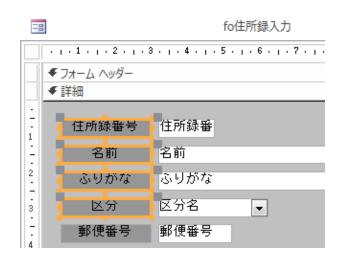


通常ラベルとテキストボックスはセットで移動しますが、マウスポインタをコントロールの左上でクリックして移動した場合、テキストボックスのみ移動となりますいろいろ試してみてください。

・複数のコントロールの位置を修正する

マウスで複数のコントロールを選択します。 ここでは、ラベルの住所番号から、区分までを選んでいます。

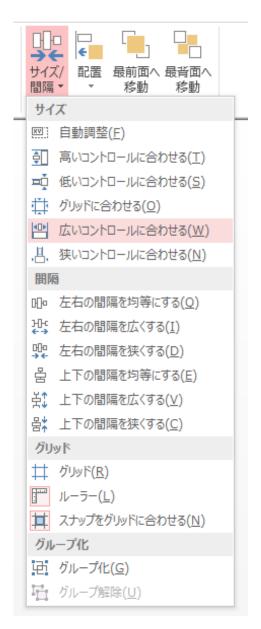




上のメニューの書式から、配置、左、で左端にラベルがそろいます。



メニューの書式から、サイズ変更、広いコントロールに、で一番広いコントロールにそろいます。



その他、いろいろあります

メニューの書式から、上下の間隔、間隔を均等にする、で上下の位置を調整します。

書式の中には、指定したコントロール(複数指定可)の右詰、中央、左詰、文字の色、コンロールの 背景色、コントロールの形等を決定できます。



このあたりのデザインでの処理は、慣れてくれば、簡単です。 効率よくできるように練習しましょう。

フォームは、コピーして使いましょう。毎回新しいのを作っていては、時間がかかります。

フォームの細かい理解は必要ありません。サンプルの中の、使えそうなフォームのコントロールを探して、コピーして貼り付けて、自分のフォームを完成させていってください。デザインが出来上がったら、入力順を変更して、IME入力モードを設定して、その項目の入力チェック処理のプロシージャをコピーしてきて貼り付けたりという作業を機械的にすれば、出来上がりです。

出来上がったフォームは、開いてデータ入力してみて、不具合があれば、その都度修正します。また 入力してみます。その繰り返しで完成させていきます。デザインの段階であれこれチェックしないで、 どんどん開いて、プログラムを動かしてみて、修正するようにしましょう。

# レポート

### ●レポートとは?

レポートは、入力されたデータを印刷するものです。ハガキに宛名を印刷したり、住所録を印刷したりします。その時に、50 音順で印刷したり、グループ毎に印刷して、1つのグループが終わったら改ページしたり出来ます。

フォームと違って、そこでデータを入力したりしませんので、簡単です。

### ●レポートの種類

レポートはデータの表示のしかたの違いで、次の2つに分かれます。

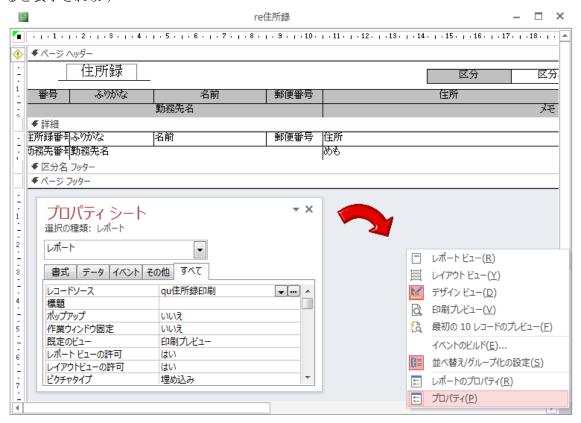
- ・単票レポート 1つのデータレコードを1つのページに印刷する。 ハガキの印刷がこれにあたります。
- ・帳票レポート 複数のデータレコードを一覧表の形で、1つのページに印刷する。 住所録の一覧表がこれにあたります。レポートはこの場合がほとんどです。

帳票レポートでは、分類毎の小計、最後の総合計の計算等の機能も持っています。 また、ある分類が終わった後の、改ページ等の機能も持っています。

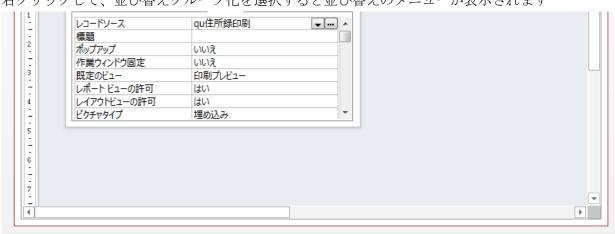
### ●レポートのプロパティ

レポートにも各部分毎にプロパティがあります。(各種設定) フォームには無かった、並び替え、グループ化の指定もあります。(印刷順序、グループ毎の処理) それを少し紹介します。

レポート全体のプロパティ。デザイン画面の何もないところで右クリックして、プロパティを選択すると表示されます



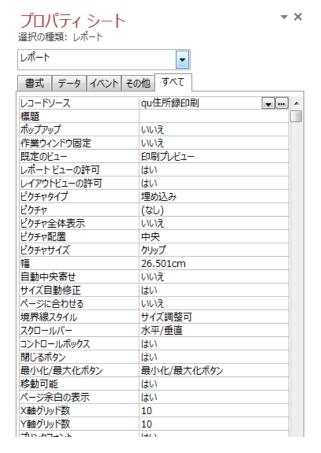
右クリックして、並び替えグループ化を選択すると並び替えのメニューが表示されます



#### グループ化、並べ替え、集計

ガループ化: **区分名 ▼** 昇順 ▼ , その他 ▶

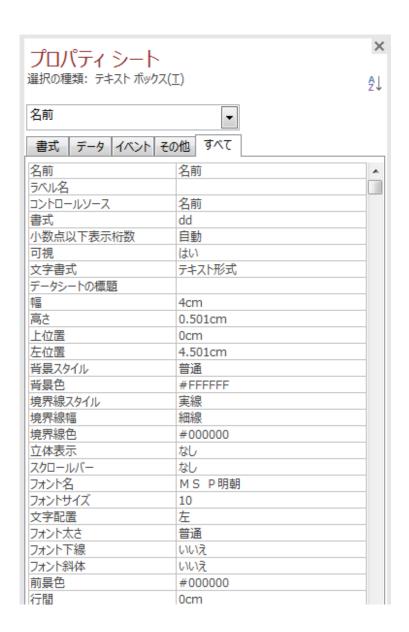
プロパティの内容は、次のようになっています。いろいろありますが、ほとんど設定はありません。



コントロールのプロパティは、各コントロールの上でクリックすると表示されます



ここでは、名前のコントロールになります。



コントロールの色、形、場所、いろいろありますが、レポートの場合はコピーしたものを、ほとんど そのまま使います。

レポートも、コピーして使いましょう。毎回新しいのを作っていては、時間がかかります。

レポートも細かい理解は必要ありません。サンプルの中の、使えそうなレポートの部品を集めてきて 貼り付けて、自分のレポートを完成させていってください。デザインが出来上がったら、プレビュー画 面で確認します。不具合があれば、その都度修正します。

## マクロ

### ●マクロとは?

マクロはエクセルにもありますが、決まった一連の処理を記述しておいて、それを動かせばプログラムが手順を案内してくれるようなものです。

しかしここでは、メニューの表示にしか使いません。

マクロは、次は何をするか記述しているだけですので、プログラムの動きを追跡するには、マクロを 開けて何をしているか確認して、フォームを開くと書いてあれば、そのフォームを開けて確認してみて、 またマクロに戻って続きは何をしているか見て・・・・、ということになって、追跡しにくいです。

フォームに埋め込んだモジュールを使って、プログラムを進める作り方をします。 マクロで使っている記述は、"docmd"という形で、すべてモジュールに書き換えることが出来ます。

独自メニューの作り方は、そのままコピーして、使ってください。 作成したプログラムを起動したときは、プログラムを作る時に出てくる、アクセスのメニューはすべて 隠してしまいます。(入力する人には必要ないです)

マクロは覚える必要はありません。独自メニューの設定で使うだけです。

# モジュール

### ●モジュールとは?

モジュールはデータベースウインドのマクロの下にあります。テーブルや、クエリーなどのオブジェクトの一つです。アクセスでプログラムを作る場合は、このモジュールは避けて通れません。使い方を覚えなくてはいけません。

マクロには限界があります。複雑な計算をさせることは出来ませんし、テーブルの中のレコードを順番に開いて、それを評価しながら作業することも出来ません。ここではマクロを使わないで、プログラムを作ることを勧めています。

プログラムを学習したことのない人には、ここが一番の難関かもしれません。

Visual Basic のプログラミングの作法がそのまま使われます。プログラム言語は難しそうに思われがちですが、われわれは、プログラマーを目指しているわけではないので、深く知る必要はありませんし、概念を理解する必要もありません。時と場合に応じた、必要な書き方だけ、少し覚えるだけでいいのです。変数の概念とか、美しいプログラムの記述がどうのこうの・・・とかは、一切関係ありません。構文を少し覚えるだけです。それも使われている状況を知っておけば、その状況の時に、コピーしてきたモジュールを少し変更して使えばいいのです。

#### ●プロシージャ

モジュールはプロシージャの集まりです。プロシージャとは、ひとつひとつに私たちが、名前を付けた関数やコマンドです。サンプルでどのように使われているかを見て、真似をして使えるようになればそれでいいと思います。

プロシージャはオブジェクトの中の、モジュールに作る場合と、フォームの中に作る場合があります。 ここでは、ほとんどフォームの中にプロシージャを入れるようにしています。いろんなフォームから呼 び出して使うようなプロシージャだけ、モジュールに入れています。それと、プログラム全体で参照す る変数の定義もモジュールで設定しています。

プロシージャには 2 種類あります。Function プロシージャと Sub プロシージャです。違いは、Function プロシージャは、引数を渡して、値を返すことができまが、Sub プロシージャではそれができません。

ここでは、その違いは知らなくてもいいです。サンプルでも値を返すような使い方はしていません。フォームの中で使うのが Sub プロシージャで、モジュールの中で使うのが Function プロシージャと思っていただいてもいいと思います。サンプルでも、モジュールの中に書いてあるプロシージャは、Function プロシージャにしています。

### ●変数

変数とは、プロシージャの中で数値や文字を一時的に覚えさせておく入れ物です。プロシージャの中で、使いますよという、宣言をします。その変数がどんな型であるかにより違います。

a という変数を数値(小数点を使わないもの)で使う場合

Dim a As Long

b という変数を文字で使う場合

Dim b As String

このあたりの使い方は、プログラムを作り始めてから、サンプルをコピーして使っているうちに、覚えられますから、この段階で解らなくてもいいです。

## ●プロシージャでよく使う構文

よく使う構文がありますので、紹介しておきます。これらはサンプルでもよく出てきます。

If ···· Then

Xの処理

Else

Yの処理

EndIf

これは、ある条件を満たした場合はXの処理を実行し、そうでない場合はYの処理をするものです。

Select Case · · ·

Case a

Xの処理

Case b

Yの処理

Case c

Zの処理

### End Select

これは、条件の結果がいくつもある場合に使います。a の場合はXの処理、b の場合はYの処理、c の場合はZの処理をするものです。

Do Until · · ·

Xの処理

#### Loop

これは、Xの処理を繰り返して実行します。レコードを順番に調査するときによく使います。

For i = 1 To 10

Xの処理

#### Next

これは、変数 i を 1 から 10 に、 1 つずつ増やしながら、 X の処理を実行するものです。 同じ処理を 10 回繰り返すことができます。 いろんな場合に使います。

このように、変数、構文等、いろいろありますが、プログラムを作る前にこれらを覚える必要はありません。自分の処理したい場面で、サンプルから似たようなものを探してコピーして使ってください、慣れてくると、いろいろ応用することにより、理解が深まると思います。

プログラマーが勉強する本に記載されているサンプルは、見ても何が書いてあるのか解らないことだらけです。(それが正しい書き方なのでしょうけど)

ここで使ってるサンプルは、なるべく解りやすく書くようにしています。なるべく複雑な処理をしないようにしています。サンプルで使えそうな部分をコピーして使ってください。

# 終わりに

# ●早くプログラムを完成させる

早くプログラムを完成させるには、とにかくサンプルプログラムの中の使えそうな部分をコピーして使うことです。そこに理解しにくいプロシージャがあったとしても、期待通りに動いていれば、それについてあれこれ考えないことです。いろいろ手直ししてると、そのうち理解できるようになりますし、解らなくても使えれば、それでいいではないですか。

また、作りたいプログラムが、かなりボリュームのあるものなら、すべてを一度に完成させようとしては駄目です。一部分でもいいからデータベースにして、仕事で使えるような状態にしましょう。それが出来上がったら次の段階に進みましょう。最終地点があまりにも遠いと、疲れてしまいます。

これを読まれた方が、自分の目指すデータベースが完成し、社内で尊敬され、会社になくてはならない人材になっていただけることをお祈りいたしております。

最後まで読んでいただき、ありがとうございました。