

Visual Basic for Applications
Excel 「V B A」
 & M y S Q L

Academic Records

成績処理にみる

システム構築のしくみ

	A	B	C	D	E	F	G	H	I
1	番号	国語	社会	数学	理科	英語	合計	平均	順位
2	1	50	80	60	50	40			
3	2	80	70	60	90	90			
4	3	90	80	100	100	80			
5	4			50	100	100			
6	合計								
7	平均								
8	最高								
9	最低								

```
Option Explicit
Sub CommandButton1_Click()
    Dim row As Integer
    Dim col As Integer
    For row = 2 To 5
        Cells(row, 7) = 0
        For col = 2 To 6
            Cells(row, 7) = Cells(row, 7) + Cells(row, col)
        Next col
    Next row
End Sub
```

県立宮崎東高等学校
 武石秀男
 2010年9月7日 初版
 10月23日 訂正

プロローグ

人は太古の昔より、「壁」に夢・ロマンを描き続けている……………。

HTXW 「Hyper Text Wall」 (ハイパー・テキスト・ウォール)

「壁に落書き」……。それは少年の日の思い出。頑固じいさんに消される運命であっても、「落書き」は情熱の表現であり、「壁」はこころ青き時代の「キャンバス」であった。

「壁」は、その向こう側に空き地という「ユートピア」を隠した夢多き存在であり、「壁」をすり抜ける時のときめきは時空を超えようとする少年のロマンの原点でもあった。

「HTXW」の上で「XDIR」や「Moving」を (月刊 I/O 1993年1月号 「HTXW & XDIR」より)

手作り「プログラミング」のすすめ

(物) (人) (金)

モノ・ヒト・カネよりも、「情報」が価値あるものとされている時代は「ほんもの」でないのかもしれませんが。

しかし今の時代、この無色無形なる「情報」は一人一人の作品であり、かけがえのない宝物だということは否定できないでしょう。

この「宝物」である「情報」は文字・画像・音楽などの「データ」やプログラムと呼ばれる「コード」からなっています。とくに「プログラム」は、多くの「宝物」に光を当てている「縁の下の力持ち」的な存在で、いわば「宝の中の宝」ということになります。

「プログラム」を作ることを「プログラミング」といいますが、「プログラミング」には「壁に落書き」に似た「ときめき」を感じます。思いどおりのものができたときの喜びは何ものにも代え難いものです。自分の思いを自分なりに表現したプログラムこそが自分のホンモノの宝物と言えるでしょう。

「情報が何モノなのか」なかなかの難問ですが、「技術継承の危機」これもまた、深刻な問題の1つです。

「手作りプログラムのススメ」、これが解決の一助となることを願って、推進していきたいと思っています。

ヒトは「知を愛する」という「哲学心」を生まれ持っています。「中身が見えて中身が分かる」ものは純なる素朴心に快感を与えるし、「何でそうなるの?」といった童心を満足させてくれます。

手作り「プログラミング」にも、かつての大人達が「ゼンマイ時計」や「真空管ラジオ」制作に打ち興じていた頃の純なる感動があります。これらの「純なる感動」こそが、高度ブラックボックス社会の今、情報を学ぶ推進力となり、未来永劫続くであろう「技術継承」の原動力となるのではないのでしょうか。

「哲学するしくみ」をつくる

「哲学する」とは、「知ることを楽しむ」と、自分なりに定義しています。

おもしろきなき世(システム構築)をおもしろく。

それは、自分の中に、そして世の中に、(システム構築を)「哲学するしくみ」をもつことです。

目次

1 はじめに	1
1-1 「ホンモノ」への道	1
1-1-1 あえて系統をたどる	1
1-1-1-1 村の鍛冶屋になって	1
1-1-1-2 「 <u>まこと</u> の <u>な</u> 」は古典の中に	1
1-2 古典にみる「基礎基本」	2
1-2-1 はじめに「プログラム」ありき	2
1-2-1-1 データ形式	2
1-2-1-2 データ処理	2
1-2-2 はじめに「データ」ありき	2
1-2-2-1 オブジェクト指向	2
1-2-2-2 データベース	2
1-2-3 身近にあるデータ管理ソフト	3
1-3 「しくみ」ありて	4
1-3-1 しくみを自分で創る	4
1-3-2 しくみ「RAD」の活用	4
2 「Excel」活用のさらなる一歩	5
2-1 VBA事始め	5
2-1-1 VBAを組み込む	5
2-1-2 マクロ使用の許可	5
2-1-3 セル指定	6
2-1-4 マクロの入力	6
2-1-5 セルの指定練習	7
2-2 簡易の成績処理プログラムをつくる	8
2-2-1 統計の基本	8
2-2-2 準備スキル	8
2-2-3 「簡易成績処理」のプログラム例	9
2-2-4 「並べ替え」(ソート)のプログラム例	10
2-3 データの「読み込みと書き出し」	11
2-3-1 ファイルからのデータ読み込み	11
2-3-2 ファイルのデータ書き出しの例	11

2-3-3	ファイル群の配布.....	12
2-3-4	ファイル群からのデータ収集.....	13
3	データベース「MySQL」によるシステム.....	15
3-1	概要.....	15
3-2	構築のための準備.....	15
3-3	環境の構築.....	16
3-3-1	「Java」の設定.....	16
3-3-2	開発統合環境「Eclipse」の設定.....	16
3-3-3	「Tomcat」の設定.....	16
3-3-4	「MySql」の設定.....	17
3-3-5	連携のための「ODBC」設定.....	17
3-3-5-1	「Excel」との連携.....	18
3-3-5-2	「桐9」との連携.....	18
3-4	「MySQL」システム構築の基礎.....	19
3-4-1	ユーザの登録.....	19
3-4-1-1	全ての権限を持つユーザの追加.....	19
3-4-1-2	制限付きのユーザ追加.....	19
3-4-2	パスワードの設定.....	19
3-4-3	ユーザの一覧表示.....	20
3-4-4	データベースとテーブルの作成.....	21
3-4-5	Excel表からMySQLテーブルへの変換.....	22
3-4-6	MySqlテーブルからExcel表へ一覧表示.....	23
3-4-7	MySqlテーブルの更新と削除.....	24
3-5	構築と運用.....	25
3-5-1	構築・管理のためのエクセル表.....	25
3-5-2	構築の手順.....	27
3-5-2-1	ユーザの管理.....	27
3-5-2-2	表の定義.....	27
3-5-2-3	表の内容作成.....	27
3-5-3	運用のためのエクセル表.....	28
3-5-3-1	フォーム.....	28
3-5-3-2	シート.....	29
3-5-4	実用モジュールの作成.....	30
3-5-4-1	アプリケーションの割り込み制御.....	30
3-5-4-2	MySqlコネクションオブジェクトを返す関数.....	31
3-5-4-3	データベース名の取得.....	31
3-5-4-4	テーブルの作成.....	32
3-5-4-5	テーブルの内容表示.....	33
3-5-4-6	テキストファイルからテーブルへの変換.....	34

3-5-4-7	テーブルからテキストファイルへの変換.....	36
3-5-4-8	定義されたテーブルの一覧表示.....	37
3-5-4-9	エクセル表で定義されたテーブル群の作成.....	39
3-5-4-10	テーブルのフィールド追加.....	40
3-5-4-11	テーブルのフィールド削除.....	41
3-6	運用マニュアル.....	42
3-6-1	環境設定.....	42
3-6-1-1	データベース「MySQL」の設定.....	42
3-6-1-2	必要なファイルのコピー.....	42
3-6-2	「MySsr」の起動.....	42
3-6-3	「MySsr」メニュー.....	43
3-6-4	入力シート.....	43
3-7	「MySQL」をPHPで操作する.....	44
3-7-1	ユーザの登録.....	44
3-7-1-1	全ての権限を持つユーザの追加.....	44
3-7-1-2	制限付きのユーザ追加.....	44
3-7-1-3	パスワードの設定.....	44
3-7-2	ユーザ&パスワードの認証.....	44
4	システム管理.....	45
4-1	環境の構築.....	45
4-1-1	リモート管理の設定.....	45
4-1-1-1	ターミナルサービスクライアント.....	45
5	研修.....	46
5-1	実習内容.....	46
5-1-1	Excel「VBA」の設定とマクロ作成.....	46
5-1-2	MySQLの環境設定と「VBA」からの操作.....	46
5-2	制作課題と目標.....	46
5-2-1	(課題1) 40人学級の9科目の成績処理システムの作成.....	46
5-2-2	(課題2).....	46
5-2-3	(課題3).....	46

1 はじめに

1-1 「ホンモノ」への道

何ごととも「個体発生は系統発生を繰り返す」です。

我々はおなかの中で 30 数億年の進化の過程を経て、人として生まれてきます。またそれが「ホンモノになるためのしくみ」だと考えます。手抜きするとホンモノになれません。不自由さ・不便さを一つ一つ楽しみながらホンモノへと成長していくのです。

全世界的な効率主義と標準化が今まで創り上げてきた多くの「系統発生のしくみ」を壊し、ダメにしています。だんだんとお手本が消えていく現代はスムーズな個体発生の難しい時代でもあります。

「情報」が海から陸に上がって半世紀ほどしか経っていませんが、地球の歴史 46 億年にも劣らない進化内容だと思います。「系統発生のしくみ」がグローバル化の荒波で消え去る前に、一人一人が「系統発生」を辿る、これはホンモノの情報スキルを身につける最良の方法であり、「スキルアップのしくみ」を学ぶ機会でもあります。とにかくもこれが私の推奨する情報スキルアップの「自律訓練法」です。

1-1-1 あえて系統をたどる

まだ、コンピュータの処理能力が低く記憶容量も小さかった頃に「どうやって短時間で統計処理が済ませられるか」この難問を解決するために必死になってプログラムと睨めっこしたものです。

何せ、限られたプログラムステップ数の中、しかも、30 メモリーしかないのに 1 クラス 40 人の成績順位をどうやって付けるか、超遅き外付けメモリーで 1 学年 320 人の順位を 1 晩で付けられるのか等々、スペック貧しき中でのやりくりは豊かな今からすると想像を絶するものでした。

でも、不便さ・貧困さの時期にこそ生涯最高のスキルアップがなされたのだと思います。飽食の時代にあっても「古きを体験する」中にこそ「スキルアップのしくみ」が見つけれられるものと信じています。

1-1-1-1 村の鍛冶屋になって

今の価値観の中心は「効率」です。「標準化」「グローバル化」がその方法論として社会構造を大きく変えてきました。その結果多くのスキルが失われてきています。

江戸時代の村の鍛冶屋よろしく一人一人が一つ一つの部品を手作りしながら、そして楽しみながら、全体の「しくみ」「からくり」を自分のものとしていきます。一人一人の工夫により多様化が計られ、結果として歴史の激変にも折れない骨太のスキルが創り上げられるのだと思います。

成績処理に関するスキルアップの極意も、アンチグローバル化です。一人一人が計算式などの便利なくみ使わないで、工夫しながらスキルの自分のものとしていきます。

何ごととも「急がば回れ」です。グローバル化・標準化ではなく、ローカル化・多様化にのちこそ、人は「生きる意味・真の喜び」を見出すのだと思います。

1-1-1-2 「^{まこと}真^なの名」は古典の中に

「古典」は国語に始まるいろんなジャンルに顔を出します。、堅苦しい響きを持ち、無意味とも思える代物。ところが最近、その「古典」が気になり出す。大事に思える。これは、むかし言葉が妙に深淵に感じられる年になったということもありますが、それよりも何よりも、全ての物ごとの「真の名」が「古典」の中に隠されていると気づいたからです。

「真の名」を知れば何でもできるようになる。といいます。(宮崎駿のアニメを見れば分かります。) それゆえに、ホンモノになるため、「成績処理」の「真の名」を知るためにも、「古典」を紐解くことが大事な勉強の 1 つなのです。

情報に関する身近な「古典の写し」といえば「情報処理試験の解説書」です。今東西のあらゆる歴史が濃縮されています。検定試験を受けるために、もしくは受けるつもりでこれらの書物を必ず一度は紐解いてみましょう。

1-2 古典にみる「基礎基本」

1-2-1 はじめに「プログラム」ありき

情報処理のはじめは、主役がプログラムで、プログラムが「しくみ」を創っていました。データはその中で操作（作成・更新・削除など）されます。この時代のスキルも大切な要素です。

1-2-1-1 データ形式

- テキスト形式（シーケンシャル・ファイル、CSV）
- バイナリ形式（ランダム・アクセス・ファイル）

1-2-1-2 データ処理

- 構造化プログラミング
「GOTO 文」の永久追放
(IF THEN) (FOR NEXT) (WHILE WEND) (DO LOOP) ……………
- 抽出と並べ替え（9つのソート）
- 合計・平均・最高・最低・順位付け・偏差値

1-2-2 はじめに「データ」ありき

今は「情報」が最も価値あるモノとされる時代です。経済的な情報戦略とともに、主役が「プログラム」から「データ」へと、パラダイムシフトしました。

1-2-2-1 オブジェクト指向

データは「オブジェクト」と呼ばれ、それ自体が「しくみ」を持っています。しくみは、属性（プロパティ）手続き（メソッド）割り込み（イベント）などを有します。

プログラムではオブジェクトを「クラス」として定義しますが、プログラミングの手法も「関数呼び出し」から「オブジェクト間通信」という概念へ移行しました。

1-2-2-2 データベース

かつて、コンピュータ言語の古典となろうとしている「COBOL」「FORTRAN」は手続きがやけに回りくどく感ぜられ、その厳格さに嫌気がさしてどうしても好きになれなかった経験があります。ところが、情報の中心となっている「データベース」を学んでみると、それら厳格さの必然的な意味がわかってきます。

今の時代は SIS「Strategic Information System(戦略情報システム)」に象徴されるように情報が産業の中心にあり、データが世界を動かしているといっても過言ではありません。それゆえに、データという「オブジェクト」は、それ自体を独立させ、論理性・信頼性・機密性・保守性を十分考慮したうえで、限りなく完全体に近いものとしなくてはなりません。そのしくみを提供してくれるのが「データベース」なのです。

データベースのモデルは、関係データベース・階層型データベース・網型データベースなどいくつかありますが、現時点では理論の簡単明瞭な「関係データベース」がほとんどです。

「関係データベース」は「リレーショナルデータベース」とも呼ばれ、以下のデータベース管理システム (DBMS) がサポートしています。

オラクル(Oracle Database)、Microsoft SQL Server、MySQL、PostgreSQL

S Q L (Structured Query Language)

1-2-3 身近にあるデータ管理ソフト

学校事務の「標準化」「効率化」「グローバル化」がどの段階まで必要なのか、スキル継承のしくみをどうして構築するのか、考えなくてはならない問題が多々あります。

データを管理するソフトはいろんな経緯もあってか各学校バラバラようです。でも、軟弱な理由からか「MS オフィス」へと移行している学校が多いようです。

結局の所、エクセルはデータ操作の要ともいえるでしょう。成績処理に限らずマクロVBAを使う機会も多くなるでしょうから、マクロを自分のものとしておきましょう。

① エクセル

- Windows (OS) とセットの「Office」で提供されている。
- 敷居が低く、気軽に導入できる。
- 計算式などが充実している。
- マクロ (VBA) で多彩な処理ができる。
- データベースとしてのしくみが乏しい。
(貧弱な部分はマクロで補う)

② アクセス

- 敷居が比較的に高い
- データベースとしてのしくみが用意されている。
- WindowOS セットの高価バージョンが必要である。

③ 桐(管理工学研究所:日本製データベース)

- 日本人好みに設計されている。(罫線の太さなど、きめ細かな設定が可能)
- 表の「一覧表」「レポート (印刷) などに多彩なしくみが用意されている。
- 一括処理 (マクロ) で多彩な処理ができる。
- データベースとしてのしくみがそれなりに用意されている。
- システム価格が高い。

1-3 「しくみ」ありて

高度情報社会の世になって、学校現場でも高度な情報処理が要求されるようになりました。より高度なことをより効率的にやろうとすれば「しくみ」が必要となります。

コンピュータ言語と言われるものも機械語（16進数）をわかりやすく表現するために創り出された「しくみ」なのです。

今までにいろんな言語が誕生しましたが「アセンブラ」（ニーモニック・マクロ）「COBOL」「BASIC」「FORTRAN」「C」などが今なお健在です。「JavaScript」「Java」「PHP」などは「C」がインターネット環境で進化したもので言語体系は同じです。

「BASIC」（≒VBAのマクロ）は長い歴史もあり、いろんなシステムに組み込まれている言語なので「しくみ」としては必須といえましょう。

ICT急進化の今は、しくみも大がかりに、そしてより複雑に階層的になってきています。出来合いの「しくみ」に満足できない人は、「オリジナルなしくみ」を、そうでない人は「出来合いのしくみ」を大いに活用してみましょう。今は、フリーで完成度の高いしくみが数多く出回っています。

OSが「MS-DOS」の時代には提供される「しくみ」が貧弱でしたが、「Windows」になってからは豊富になりました。その豊富さには嫌気がさすほどで、我々の出番を残す意味でも「しくみ」は「腹六分目～八分目」程度にしてもらいたいものです。

1-3-1 しくみを自分で創る

今どき「しくみ」のないものはありません。

結局、しくみの中で「しくみ」を創ることになります。さらに、その中に「しくみ」を、と階層的に創っていきます。「しくみ」を創り始めると「情報観」が変わっていきます。

- 高いスキルを必要とする。
- 思いどおりのシステムが構築できる。

1-3-2 しくみ「RAD」の活用

Rapid Application Development

（すばやく 実務処理プログラム 開発）

- あまりスキルを必要としない。
- それなりに費用がかかる
今ではフリーソフトが充実しています。

2 「Excel」活用のさらなる一歩

「Excel」は、昔からある「計算ソフト」に「関係データベース」「オブジェクト指向」を組み込んだ多機能なプログラムで、活用すればするほど奥の深さを感じる「アプリケーションソフト」です。「セル」と呼ばれる「小箱」（細胞）に数字や文字などを入れて、統計などの数値計算や表・グラフなど、多彩な処理をすることができます。

そして、さらに「VBA」と呼ばれるBASICプログラムで高度な処理が可能となります。

2-1 VBA事始め

VBAは「Visual Basic for Applications」の略称です。

マイクロソフトの（というか、ビルゲイツの）隠れ戦略の1つが「BASICを再び陽の当たる場所へ」と憶測しています。（青春時代の「BASIC」よ蘇れ！）

2-1-1 VBAを組み込む

VBAのツールが表示されていない場合は、以下の操作で、VBA環境を表示します。

- 「Excel2003」の場合
 - ① ダウンメニュー「表示 (V)」をクリックして、メニューを開く。
 - ② 「ツールバー」の上にカーソルを持って行き、サブメニューを表示する。
 - ③ Visual Basic にチェックを入れる。
- 「Excel2007」の場合
 - ① 「Office ボタン」をクリックして、メニューを開く。
 - ② 最下行の「Excel のオプション」をクリックし、オプション設定メニューを表示。
 - ③ （「開発」タブをリボンに表示する）にチェックを入れる。
 - ④ 「メニューバー」に「開発」が表示されるのでクリックする。

2-1-2 マクロ使用の許可

最近はウィルス対策の1つとして、初期値では「マクロ」使用が厳しく制限されています。自分の作ったマクロは問題ないでしょうから、以下の手順で設定変更します。

- 「Excel2003」の場合
 - ① ダウンメニュー「ツール (T)」の「オプション (O)」設定画面を開く。
 - ② 「セキュリティ」タブをクリックし、階層下設定画面を開く。
 - ③ その中の「マクロ セキュリティ」ボタンをクリックし、階層下画面を開く。
 - ④ 「セキュリティレベル」を「低」または「中」に設定する。
- 「Excel2007」の場合
 - ① 「Office ボタン」をクリックして、メニューを開く。
 - ② 最下行の「Excel のオプション」をクリックし、オプション設定メニューを表示。
 - ③ 左項目「セキュリティセンター」をクリックし、「セキュリティセンターの設定」を開く。
 - ④ 左項目「マクロの設定」をクリックし、「マクロの設定」の最下段項目を選択する。
または
 - ① 「開発タブ」をクリックして、「マクロのセキュリティ」をクリックする。
 - ② 左項目「マクロの設定」をクリックし、「マクロの設定」の最下段項目を選択する。

2-1-3 セル指定

「セル」には1次元配列として「セル番号」が付いています。左から右・上から下への連番で「1～65536」の範囲で指定できます。

Cells(1)～Cells(65536)

ただし、最新バージョンでは列数が 16784 (2¹⁴)、行数が1048576 (2²⁰) 行へと拡張されています。

この指定では平面上の位置関係が分かりにくいので、2次元配列としても指定できるようにしてあります。また、この指定の方が一般的です。

ただし、数学の配列と行と列の指定順序が逆になっているので注意が必要です。

Cells(行、列)

	A	B	C	D
1	Cells(1, 1)	Cells(1, 2)	Cells(1, 3)	Cells(1, 4)
2	Cells(2, 1)	Cells(2, 2)	Cells(2, 3)	Cells(2, 4)
3	Cells(3, 1)	Cells(3, 2)	Cells(3, 3)	Cells(3, 4)
4	Cells(4, 1)	Cells(4, 2)	Cells(4, 3)	Cells(4, 4)

2-1-4 マクロの入力

- (1) 「デザインモード」に変更する
 - 「Excel2003」の場合
 - 「デザインモード」アイコンをクリックし、「コントロールツールボックス」を表示する。
 - 画面に表示されていない場合は
 - 「ダウンメニュー」の「表示」「メニューバー」「Visual Basic」にチェックを付ける。
 - 「Excel2007」の場合
 - 「挿入」アイコンをクリックする。
- (2) 「コマンドボタン」を貼り付ける。
 - ① 「コントロールツールボックス」の中の
 - 「ActiveX コントロール」(下半分)の方の「コマンドボタン」をクリックする。
 - ② 画面上の適当な位置でドラッグ(左ボタンを押したまま移動する)して、「コマンドボタン」を貼り付ける。「コマンドボタン」ダブルクリックして、プログラム編集モードにする。
- (3) カーソル位置に、プログラムを打ち込む。
 - 例として、以下を入力。

$$\text{Cells}(3,4) = \text{"VBA のはじまり"}$$
- (4) ワークシートに戻る
 - タスクバーに表示された「ワークシート」をクリックしたする。
- (5) 「デザインモード」アイコンをクリックし、「デザインモード」を終了する。
- (6) 実行と画面の再設定
 - 「コマンドボタン」をクリックして、VBA(マクロ)の動作を確認する。

2-1-5 セルの指定練習

(1) 直接指定によるセルへの代入

```
Cells(2, 1).Value = 3      ' (セル「A2」に数字の「3」が入力される)
Cells(2, 2) = 4           ' 「.Value」部分は省略できます。
```

(2) 変数による間接指定によるセルへの代入

```
n = 3                     ' (変数「n」に「3」を代入する)
Cells(n, 2) = 4           ' (セル「B3」に数字の「4」が入力される)

n = 3
m = 4
Cells(n, m) = 5           ' (セル「D3」に数字の「5」が入力される)
```

(3) 繰り返しによるセルへの代入

```
For n = 1 To 9             ' (変数「n」を1から9までステップ1で変化させる)
    Cells(n, 2) = n        ' (n = 1, 2, 3, 4, 5, 6, 7, 8, 9)
Next n
```

(4) 条件分岐によるセルへの代入

```
n = Cells(2, 1)           ' A2のセルの値を取り出す。
IF n < 6 Then              ' もし、変数nが6よりも小さかったならば次の行を実行
    Cells(2, 2) = 460
Else                        ' そうでなかったならば次の行を実行
    Cells(2, 2) = 1000
End If                     ' IF文の終了
```

2-2 簡易の成績処理プログラムをつくる

2-2-1 統計の基本

- (1) 横の合計・平均
- (2) 縦の合計・平均・最高・最低
- (3) 順位づけ

	A	B	C	D	E	F	G	H	I
1	番号	国語	社会	数学	理科	英語	合計	平均	順位
2	1	50	80	60	50	40			
3	2	80	70	60	90	90			
4	3	80	70	60	90	90			
5	4	90	80	100	100	80			
6	合計								
7	平均								
8	最高								
9	最低								

2-2-2 準備スキル

得点を直接入力するのは面倒な場合は、乱数を使って得点を埋めます。

```
Option Explicit ' 変数定義を明示 (変数定義がないとエラーとなるように設定する)
Private Sub CommandButton1_Click()
    Dim row As Integer ' 行変数
    Dim col As Integer ' 列変数
    For row = 2 To 5
        For col = 2 To 6
            Cells(row, col) = Int(Rnd * 100) + 1 ' 乱数の代入 (1 から 100 までの整数)
        Next col
    Next row
End Sub
```


2-2-3 「簡易成績処理」のプログラム例

```

Option Explicit
Private Sub ComBtnCalc_Click()
    Dim row, col, ttl, num, cnt, max, min, dat, rank, dmx As Integer

    ' -----横の合計・平均
    num = 0 ' 人数
    For row = 2 To 5
        ttl = 0 ' 横合計
        cnt = 0 ' 科目数
        For col = 2 To 6
            ttl = ttl + Cells(row, col).Value
            cnt = cnt + 1
        Next col
        Cells(row, 7).Value = ttl
        Cells(row, 8).Value = ttl / cnt
        num = num + 1
    Next row

    ' -----縦の合計・平均・最高・最低
    For col = 2 To 6
        ttl = 0 ' 縦合計
        max = -1 ' 初期最大値
        min = 999 ' 初期最小値
        For row = 2 To 5
            dat = Cells(row, col).Value
            ttl = ttl + dat
            If max < dat Then max = dat
            If min > dat Then min = dat
        Next row
        Cells(6, col).Value = ttl
        Cells(7, col).Value = ttl / num
        Cells(8, col).Value = max
        Cells(9, col).Value = min
    Next col

    ' ----- 順位付け (例)
    max = 999 ' 順位付け対象上限値
    rank = 1
    Do ' 「Loop」までの条件内繰り返し
        dmx = -1 ' 順位付け合計値
        For row = 2 To 5
            dat = Cells(row, 7).Value
            If dat < max Then
                If dat > dmx Then
                    Cells(row, 9) = rank
                    dmx = dat ' 暫定的な最大値の更新
                    cnt = 1
                Else
                    If dat = dmx Then
                        Cells(row, 9) = rank
                        cnt = cnt + 1 ' 同順位数をカウント
                    End If
                End If
            End If
        Next row
        rank = rank + cnt ' 次の順位設定 (同順位付け数を加算)
        max = dmx ' 順位付け対象上限値を検索最大値に再設定
    Loop While rank <= num
End Sub

```

2-2-4 「並べ替え」(ソート) のプログラム例

Option Explicit

Private Sub ComBtnSort_Click() ' ソート (隣接交換法) セル上での並べ替え

Dim row, flg, dt1, dt2 As Integer

For row = 1 To 1000

Cells(row, 1) = Int(Rnd * 1000) + 1 ' 1 から 1000 までの数値を縦に'

Next row

Do

flg = 0

For row = 1 To 999

dt1 = Cells(row + 0, 1)

dt2 = Cells(row + 1, 1)

If dt1 < dt2 Then

Cells(row + 0, 1) = dt2

Cells(row + 1, 1) = dt1

flg = 1

End If

Next row

Loop While flg = 1

End Sub

'----- (高速版) 上の方法と比較してみよう -----'

Private Sub ComBtnDimSort_Click() ' ソート (隣接交換法) 配列に格納してからの並べ替え

Dim row, flg, dt1, dt2 As Integer

Dim dat(1000) As Integer

For row = 1 To 1000

Cells(row, 1) = Int(Rnd * 1000) + 1 ' 1 から 1000 までの数値を縦に表示

dat(row) = Cells(row, 1) ' 配列に格納

Next row

Do

flg = 0

For row = 1 To 999

dt1 = dat(row + 0)

dt2 = dat(row + 1)

If dt1 < dt2 Then

dat(row + 0) = dt2

dat(row + 1) = dt1

flg = 1

End If

Next row

Loop While flg = 1

For row = 1 To 1000

If Cells(row, 1) = "" Then Exit For

Cells(row, 1) = dat(row)

Next row

Cells(1, 1).Select

End Sub

2-3 データの「読み込みと書き出し」

江戸の鍛冶職人は、西洋の利器を一目見ただけで同じものを、または、それ以上のものが作れたといいます。天平時代の仏師もそうですが、そのスキルの高さには驚きです。プログラムを読み解き、それを思いのままに作り替える、これも情報スキルの大事な要素です。成績処理にはそのための特有なスキル（データの読み・書き・配布・収集）が必要とされます。

2-3-1 ファイルからのデータ読み込み

```
Option Explicit
Private Sub CommandButton1_Click()
    Dim buff As String          ' 1行読み込み文字列
    Dim temp As Variant        ' 読み込み文字列の配列化
    Dim row As Long           ' 行変数
    Dim col As Long           ' 列変数

    Open ThisWorkbook.Path & "\sample.txt" For Input As #1
    row = 2
    Do Until EOF(1)
        Line Input #1, buff
        temp = Split(buff, ",")    ' 1行の文字列をカンマで切り分ける
        For col = 0 To UBound(temp) ' 配列の最後まで繰り返す
            Cells(row, col + 2) = temp(col)
        Next col
        row = row + 1
    Loop
    Close #1
End Sub
```

2-3-2 ファイルのデータ書き出しの例

```
Private Sub CommandButton2_Click()
    Dim buff As String          ' 1行書き込み文字列
    Dim row As Integer          ' 行変数
    Dim col As Integer          ' 列変数

    Open ThisWorkbook.Path & "\sample2.txt" For Output As #1
    For row = 2 To 5
        buff = ""
        For col = 2 To 6
            If col > 2 Then buff = buff & ","
            buff = buff & Cells(row, col)
        Next col
        Print #1, buff
    Next row
    Close #1
End Sub
```

2-3-3 ファイル群の配布

```

Private Sub CommandButton5_Click()
    Dim n As Integer          ' 科目のカウンント
    Dim bookna As Variant    ' 科目名を格納する配列 (科目名は必要な数だけ並べる)
    Dim subj As String       ' 科目名
    Dim bkna As String
    Dim bkph As String
    Dim rps As Integer
    Dim row As Integer
    Dim col As Integer
    Dim tFile As Workbook
    Dim mybtn As Integer

    mybtn = MsgBox("科目の単票を書き出しますか?", vbYesNo + vbQuestion, "科目単票の書き出し")
    If mybtn <> vbYes Then End

    bookna = Array("国語総合", "現代社会", "数学 I", "理科総合 A", "体育", "保健", "音楽 I", "英語 I")
    For n = 0 To UBound(bookna)
        subj = bookna(n)
        bkna = "単票" & subj & ".xls"
        bkph = ThisWorkbook.Path & "\¥" & bkna ' 科目のフルパス名
        If Dir(bkph) = "" Then ' ファイルがなかったら作成する
            Set tFile = Workbooks.Add
            tFile.SaveAs Filename:=bkph
            tFile.Close
        End If
        Workbooks.Open Filename:=bkph
        With Workbooks(bkna).Worksheets(1)
            .Range("B2:W4000").ClearContents
            rps = 2
            For row = 2 To 20000
                If Cells(row, 1) = subj Then ' 1 列目に教科名が一致したら
                    For col = 1 To 20 ' コピー部分は適当に
                        .Cells(rps, col) = Cells(row, col)
                    Next col
                    rps = rps + 1
                End If
            Next row
        End With
        Workbooks(bkna).Save ' 科目ファイルを保存
        Workbooks(bkna).Close ' 科目ファイルを閉じる
    Next n
End Sub

```

2-3-4 ファイル群からのデータ収集

```

Private Sub CommandButton3_Click()
    Dim xname As String          ' 検索ファイル群
    Dim bkna As String          ' 検索されたファイル（ブック）名
    Dim bkph As String          ' 検索されたフルパス名
    Dim row As Integer          ' 行
    Dim col As Integer          ' 列
    Dim rps As Integer

    xname = ThisWorkbook.Path & "¥単票*.xls" ' 「*」はワイルドカード
    rps = 2                      ' 「単票+科目名」のエクセルファイルのみ検索
    bkna = Dir(xname, vbNormal)  ' 2行目からコピーを開始
    Do While bkna <> ""          ' 1番目のファイル名を検索
        Cells(1, 1) = bkna      ' ファイル名の検索が尽きるまで
        bkph = ThisWorkbook.Path & "¥" & bkna ' 1行目にファイル名を表示(確認の意味)
        Workbooks.Open Filename:=bkph ' 科目のフルパス名
        With Workbooks(bkna).Worksheets(1)
            For row = 2 To 1500
                If .Cells(row, 1) = "" Then Exit For
                For col = 1 To 10
                    Cells(rps, col) = .Cells(row, col) ' データをコピーする
                Next col
                rps = rps + 1
            Next row
        End With
        Workbooks(bkna).Close ' 検索ファイルを閉じる
        bkna = Dir()          ' 次のファイル名を検索
    Loop
End Sub

```


3 データベース「MySQL」によるシステム

3-1 概要

「インターネット」や「イントラネット」のしくみを理解することは必須といえます。この数年、社会的業務は、「インターネット」をインフラとした「イントラネット」へと移行してきて、その中心軸となる「データベースサーバ」として「MySQL」が活用されています。

成績処理では、ファイル共有・同時アクセス・排他処理などの難問を解決しなくてはなりません。それらを問題としないのが「データベース」といえます。使ってみれば、すぐにその爽快さを体感できます。

3-2 構築のための準備

サーバ	サービス内容
Apache	Web サーバ。 HTTP などのプロトコルを用いメッセージなどの送受信を行う。 Web の URL (DNS) の応答サービス
Tomcat	Java サブレット・JSP を処理するアプリケーションサーバ。 Tomcat は単独で Web サーバとして動作することも可能ですが、Apache や IIS のプラグインと動作できるようになっており、実際にはプラグインとしての利用が主流です。
Java	狭義ではオブジェクト指向プログラミング言語 Java で、広義ではプログラミング言語 Java のプログラムの実行環境および開発環境をいいます。
MySQL	無償で入手できる RDBMS(リレーショナルデータベース管理システム)です。 PostgreSQL と並ぶオープンソース RDBMS の代表格で、欧州で特に人気があります。最大の特徴は検索の高速性に重点を絞った設計になっていることで、商用 RDBMS に比べ、処理を遅くする機構を大幅に省いた反面、大量のデータを高速に検索することができます。UNIX 系各種 OS や Windows など多くのプラットフォームに対応しているのも利点です。
ODBC	Open DataBase Connectivity Microsoft 社によって提唱された、データベースにアクセスするためのソフトウェアの標準仕様。各データベースの違いは ODBC ドライバによって吸収されるため、ユーザは ODBC に定められた手順に従ってプログラムを書けば、接続先のデータベースがどのようなデータベース管理システムに管理されているか意識することなくアクセスできるようになります。

3-3 環境の構築

以後のインストールファイルはサーバのフォルダ「App¥Java」にダウンロード済みです。

3-3-1 「Java」の設定

この設定は「Java」サーバを構築するためのものです。

- (1) 「JDK」(Java Development Kit)のインストールファイルをダウンロードし、実行する。

「jdk-6u17-windows-i586.exe」

インストーラファイルをダブルクリックしてインストールを実行

- (2) 環境変数「JAVA_HOME」を新規作成

(例)「JAVA_HOME」 = 「C:¥Program Files¥Java¥jdk1.6.0_17」

- (3) 環境変数「Path」に「Java」のパスを追加する

「%JAVA_HOME%¥bin」(デリミッタ「」を付けること)

- (4) 環境変数「CLASSPATH」に以下を追記

「.%JAVA_HOME%¥lib¥dt.jar %JAVA_HOME%¥lib¥tools.jar」

3-3-2 開発統合環境「Eclipse」の設定

この設定は「Java プログラム開発のための統合環境」を提供するためのものです。

- (1) 「Eclipse3.5」(ガリレオ)のインストールファイルをダウンロード。

「eclipse-jee-galileo-win32.zip」

- (2) ZIP形式の圧縮ファイルを展開して「C:work」フォルダに移動する

- (3) 日本語化ソフトウェアをダウンロードし、ZIP形式の圧縮ファイルを展開。

「pleiades.zip」

- (4) 「plugins」フォルダ内の全ファイルを「Eclipse」の「plugins」フォルダにコピー。

- (5) 「features」フォルダ内の全ファイルを「Eclipse」の「features」フォルダにコピー。

- (6) 環境設定ファイル「eclipse.ini」の末尾に、以下のプラグイン項目を追記する。

「-javaagent:plugins/jp.sourceforge.mergedoc.pleiades/pleiades.jar」

3-3-3 「Tomcat」の設定

この設定は「サーバサイドJavaを動かすための環境」を提供するためのものです。

- (1) サーバソフトインストールファイルをダウンロードし、実行。

「apache-tomcat-6.0.20.exe」

インストーラファイルをダブルクリックしてインストールを実行

3-3-4 「MySQL」の設定

この設定は「データベースサーバ」を提供するためのものです。

(1) 「Xampp」のインストールファイルをダウンロードし、実行。

「 xampp-win32-1.7.3.exe 」

これは、「Apache」と「MySQL」の設定をセットで行ってくれる便利なツールです。

文字化け対策として、システム全体の文字コードを「UTF-8」に統一する。

「MySQL」の環境設定ファイル「my.ini」の中を以下のように追加変更する。

C:\xampp\mysql\bin (フォルダ例)

[client]

default-character-set=utf8
skip-character-set-client-handshake

[mysqld]

default-character-set = utf8
skip-character-set-client-handshake
character-set-server = utf8
collation-server = utf8_general_ci
init-connect = SET NAMES utf8

- (2) ディスクトップ上の「Xampp Control Panel」をクリックして設定画面を表示する。
- (3) 「MySQL」の「Start」ボタンをクリックして、起動する。
次の設定確認のために起動しておきます。

3-3-5 連携のための「ODBC」設定

この設定は「Excel」「access」「桐9」などと「MySQL」とを連携するためのものです。

(1) 「MyODBC」のインストールファイルをダウンロードし、実行。

「mysql-connector-odbc-5.1.6-win32.msi」

インストーラファイルをダブルクリックしてインストールを実行

(2) 「データ ソース (ODBC)」の設定

- ① 「コンピュータ管理」の「データ ソース (ODBC)」を開く。
「コントロールパネル→パフォーマンスとメンテナンス→管理ツール」で辿り着く
- ② 「システムD S N」タブをクリック。
- ③ 「追加」ボタンをクリックして、「MySQL ODBC 5.1 Driver」を選択し、完了。
- ④ データソース (Data Source Name) 名を「MySQL5.1」(例)とする。
- ⑤ テストのために以下を入力する。

入力項目	入力値	通信制の場合
Server	localhost	sssr
User	root	自分のトリエスID
Password	なし	自分のトリエスパスワード
DataDase	test	db_ssr22

- ⑥ 「Test」ボタンをクリックして「MySQL」との接続テストを行う。
「Conection Successful」のメッセージが表示されたら設定完了。

3-3-5-1 「Excel」との連携

- (1) 「ワークシート」でコネクト（接続）する場合
 - 「Excel2003」の場合
 - ① 「データ」メニュー「外部データの取り込み」の「データ取り込み」
 - ② 「新しいデータソースへの接続」を開いて、「ODBC DSN」を選択して次へ
 - ③ 「MySQL5.1」を選択する。
 - ④ 「データベース名」と「テーブル名」を指定する。
 - 「Excel2007」の場合
 - ① 「データ」メニューの「その他のデータソース」
 - ② 「データ接続ウィザード」を選択して、「ODBC DSN」を選択する。
 - ③ 「MySQL5.1」を選択する。
 - ④ 「データベース名」と「テーブル名」を指定する。
- (2) マクロでコネクトする場合

「VBA」画面の「ツール」メニュー「参照設定」で以下の項目にチェックを入れる。
「Microsoft ActivexX Data Objects ×.× Library」（「×.×は」最新版）

3-3-5-2 「桐9」との連携

- ① 外部DBタブを選択
- ② 新規作成でファイル名を入力
- ③ 外部データベースに接続メニューで「MySQL5.1」を選択し、「接続」ボタンをクリック
- ④ 「ユーザ名」とパスワードを入力。
- ⑤ 「結合対象表」で「表」にチェックを入れ、「表の名前」を選択する。
MySQLが「UTF-8」なので全角文字があると読み込めない（2010年2月1日現在）

3-4 「MySQL」システム構築の基礎

3-4-1 ユーザの登録

3-4-1-1 全ての権限を持つユーザの追加

```
Private Sub ComBtnUser_Click()
    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String
    Dim idStr As String
    Dim pwStr As String

    conStr = "Driver={MySQL ODBC 5.1 DRIVER};" & " SERVER=localhost;" & " DATABASE=;"
        & " USER=root;" & " PASSWORD=;"

    Set con = New ADODB.Connection
    On Error GoTo Err
    con.Open conStr

    idStr = "rio"           ' ユーザ名
    pwStr = "abcd"         ' パスワード
    sqlStr = "grant all privileges on *.* to " & idStr & "@'" & pwStr & "';"
    Set rs = con.Execute(sqlStr)   'SQL 文実行
    con.Close
    Set rs = Nothing
    Set con = Nothing
    Exit Sub

Err:
    Set rs = Nothing
    Set con = Nothing
    MsgBox (Err.Description)
End Sub
```

3-4-1-2 制限付きのユーザ追加

上記プログラムの下線部分を以下のように書き換える

```
sqlStr = "grant select insert update delete on *.* to " & idStr & "@'" & pwStr & "';"
```

3-4-2 パスワードの設定

上記プログラムの下線部分を以下のように書き換える

```
sqlStr = "set password for " & idStr & "=password(" & pwStr & ");"
```

3-4-3 ユーザの一覧表示

```

Private Sub CommandButton3_Click()
    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String

    Dim row , col As Integer
    Dim item As Variant

    conStr = "Driver={MySQL ODBC 5.1 DRIVER};" & " SERVER=localhost;" & "
            DATABASE=mysql;" & " USER=root;" &
            " PASSWORD="

    Set con = New ADODB.Connection
    On Error GoTo Err
    con.Open conStr
    sqlStr = "select * from mysql.user;"
    Set rs = con.Execute(sqlStr)

    For col = 0 To 2                                     ' rs.Fields.Count - 1 (全てなら)
        Cells(2, 2 + col).Value = rs.Fields(col).Name   ' 項目名の表示
    Next col

    row = 3
    Do While rs.EOF = False
        col = 2
        For Each item In rs.Fields
            Cells(row, col).Value = item.Value
            col = col + 1
            If col > 4 Then Exit For
        Next
        row = row + 1
        rs.MoveNext
    Loop

    con.Close
    Set rs = Nothing
    Set con = Nothing
Exit Sub

Err:
    Set rs = Nothing
    Set con = Nothing
    MsgBox (Err.Description)
End Sub

```

下線部は下記でもよい。

```

Do While rs.EOF = False
    For col = 0 To 2
        Cells(rowNo, 2 + col).Value = rs.Fields.item(col)
    Next col

```

3-4-4 データベースとテーブルの作成

```

Private Sub CommandButton1_Click()
    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String

    conStr = "Driver={MySQL ODBC 5.1 DRIVER};" & " SERVER=localhost;" & "
            DATABASE=;" & " USER=root;" & " PASSWORD=;"

    Set con = New ADODB.Connection
    On Error GoTo Err          ' エラーの場合の処理先を指定
    con.Open conStr          ' データベースに接続します。

    sqlStr = "create database sample"      ' データベースの作成
    Set rs = con.Execute(sqlStr)          ' SQL 文実行
    sqlStr = "use sample"                 ' データベースの指定
    Set rs = con.Execute(sqlStr)          ' SQL 文実行

    sqlStr = "create table tests("        ' テーブル作成用 SQL 文
    sqlStr = sqlStr & "no integer,"
    sqlStr = sqlStr & "name char(32),"
    sqlStr = sqlStr & "kokugo integer,"
    sqlStr = sqlStr & "suugaku integer,"
    sqlStr = sqlStr & "english integer);"
    Set rs = con.Execute(sqlStr)          ' SQL 文実行

    sqlStr = "insert into tests value(1,'宮崎東 太郎',100,90,80);"
    Set rs = con.Execute(sqlStr)          ' SQL 文実行 (1行分のデータ挿入)
    sqlStr = "insert into tests value(2,'通信制 花子',80,70,90);"
    Set rs = con.Execute(sqlStr)          ' SQL 文実行

    con.Close
    Set rs = Nothing
    Set con = Nothing
    Exit Sub

Err:
    Set rs = Nothing
    Set con = Nothing
    MsgBox (Err.Description)
End Sub

```

3-4-5 Excel 表から MySQL テーブルへの変換

```

Private Sub CommandButton3_Click()
    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String

    Dim row, col As Integer

    conStr = "Driver={MySQL ODBC 5.1 DRIVER};" & " SERVER=localhost;" & "
            DATABASE=sample;" & " USER=root;" & " PASSWORD=;"

    Set con = New ADODB.Connection
    On Error GoTo Err
    con.Open conStr

    sqlStr = "delete from tests;"           ' 全データの削除
    Set rs = con.Execute(sqlStr)           ' SQL 文実行
    For row = 3 To 4
        sqlStr = "insert into tests value(" & str(Cells(row, 2)) & "," & Cells(row, 3) & "," ' 番号と氏名
        For col = 4 To 6                     ' データ (点数) の結合
            If col > 4 Then sqlStr = sqlStr & ","
            sqlStr = sqlStr & str(Cells(row, col))
        Next col
        sqlStr = sqlStr & ");"
        Set rs = con.Execute(sqlStr)         ' SQL 文実行
    Next row

    con.Close
    Set rs = Nothing
    Set con = Nothing
    Exit Sub

Err:
    Set rs = Nothing
    Set con = Nothing
    MsgBox (Err.Description)
End Sub

```

3-4-6 MySql テーブルから Excel 表へ一覧表示

```

Private Sub CommandButton1_Click()
    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String

    Dim col As Integer
    Dim rowNo As Integer
    Dim colNo As Integer
    Dim item As Variant

    conStr = "Driver={MySQL ODBC 5.1 DRIVER};" & " SERVER=localhost;" & "
            DATABASE=sample;" & " USER=root;" & " PASSWORD=;"
    Set con = New ADODB.Connection
    On Error GoTo Err
    con.Open conStr
    sqlStr = "select * from tests;"
    Set rs = con.Execute(sqlStr)

    For col = 0 To rs.Fields.Count - 1
        Cells(2, 2 + col).Value = rs.Fields(col).Name
    Next col

    rowNo = 3
    colNo = 2
    Do While rs.EOF = False
        For Each item In rs.Fields
            Cells(rowNo, colNo).Value = item.Value
            colNo = colNo + 1
        Next
        colNo = 2
        rowNo = rowNo + 1
        rs.MoveNext
    Loop

    con.Close
    Set rs = Nothing
    Set con = Nothing
Exit Sub

Err:
    Set rs = Nothing
    Set con = Nothing
    MsgBox (Err.Description)
End Sub

```

’項目名（フィールド名）の表示

’項目名をセルにコピー

’最終行まで繰り返す

’全列データを取り出す

’次の行へ移動する。

3-4-7 MySQL テーブルの更新と削除

- (1) 出席番号 2 番の生徒の国語の成績を 90 点に変更する。

```
Option Explicit                                     ' 変数定義を明示化する
Private Sub CommandButton1_Click()
    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String

    Dim sno As Integer
    Dim dat As Integer

    conStr = "Driver={MySQL ODBC 5.1 DRIVER};" & " SERVER=localhost;" & "
            DATABASE=sample;" & " USER=root;" & " PASSWORD=;"
    Set con = New ADODB.Connection
    On Error GoTo Err
    con.Open conStr

    sno = 2
    dat = 90
    sqlStr = "update tests set kokufoe=" & str(dat) & " where no=" & str(sno) & " ;"
    Set rs = con.Execute(sqlStr)

    con.Close
    Set rs = Nothing
    Set con = Nothing
Exit Sub

Err:
    Set rs = Nothing
    Set con = Nothing
    MsgBox (Err.Description)
End Sub
```

- (2) 出席番号 2 番の生徒のデータを全削除する。

上記下線部分を以下のように書き換える

```
sno = 2
sqlStr = "delete from tests where no=" & str(sno) & " ;"
```


3-5 構築と運用

エクセル版システム全般は「M:¥Excel」フォルダにある。ちなみに「M:」には、コンピュータサーバ「¥¥Sssr」のフォルダ「D:¥MySssr¥MySssr」がネットワークドライブとして割り当てられている。

「MySQL」データは「¥¥Sssr」の「C: ¥xampp¥mysql¥data」フォルダに構築される。データベース名をそのままフォルダ名とし、その中に各テーブルが3対のファイル名として作成される。

3-5-1 構築・管理のためのエクセル表

「M:¥Excel¥make」フォルダの「MakeSssr.xls」に以下のシートを作成各処理には、以下のようなワンタッチボタンが用意されている。

- (1) 「ユーザ管理」シート
 - 「ユーザ登録一覧表」: 「Host」「user」「password」(暗号化)を一覧表示
 - 「ユーザ登録追加」: ユーザの追加
 - 「ユーザの削除」: ユーザを削除
 - ログイン情報 (システム全体に共通な情報)
 - 「サーバ名」「データベース名」「ユーザ名」「パスワード」
 - とくに、「データベース名」は構築・運用 (変更) の際に慎重十分注意する。
- (2) 「表定義」シート
 - 「定義されたテーブルの一覧表示」: 全テーブルとその項目・型を表示
 - 「表定義の実行」: 以降の行に記述された「表・項目・型」を新たに定義
 - 「定義不一致のチェック」: 一覧表示内容と定義された内容の差異をチェック
- (3) 「項目変更」シート
 - 「フィールドの追加」: 以降に表示されたテーブルにフィールドを追加
 - 「フィールドの削除」: 以降に表示されたテーブルにフィールドを削除
 - 「フィールドの変更」: 以降に表示されたテーブルにフィールドを変更
- (4) 「表の変換」シート
 - 「選択メニュー」: シート「表群」に表示された表名より選択する
 - 「テーブルの読み込み」: 項目・内容を一覧表示する。
 - 「テキストからテーブルへの変換」: シート「表群」に表示されたフォルダから併合。
 - 「テーブルからテキストへの変換」: シート「表群」に表示されたフォルダに書き出す。
- (5) 「表群」シート
 - 「MySssr テーブルの一覧表」: 「dbSssrs」テーブルの内容を一覧表示する。
 - 「テキストから全テーブルへの変換 (○印のみ)」: シート「表群」に表示されたフォルダから併合。○印がついたものだけが適用される。
 - 「テーブルから全テキストへの変換 (○印のみ)」: シート「表群」に表示されたフォルダに書き出す。○印がついたものだけが適用される。
- (6) 「構成」シート
 - 基礎データの管理シート
 - メニュー選択項目のデータを管理する
 - ログイン時のデータベース管理「セル O4:O7」を行う。

3-5-2 構築の手順

3-5-2-1 ユーザの管理

- (1) 「ユーザ管理」シートで行う
- (2) 「ユーザ登録一覧表」ボタンをクリックして、一覧でユーザの確認を行う。
- (3) 追加登録は、セル「F3」以降にユーザ情報を記入し、「ユーザ登録追加」をクリックする。
- (4) ユーザ削除は、セル「J3」以降にユーザ情報を記入し、「ユーザの削除」をクリックする。

3-5-2-2 表の定義

- (1) 「構成シート」の管理セル「O4:O7」にデータベース名などの MySQL 情報を入力する。
- (2) 「表定義」シートでテーブル群を作成する。
セル「I2」以降にテーブル名・フィールド名・フィールドタイプを記入し、「表定義の実行」ボタンをクリックする。
「定義されたテーブルの一覧表示」ボタンをクリックし、正しく定義されているかを確認する。

3-5-2-3 表の内容作成

- (1) 「桐 9」や「エクセル」でテーブル作成用の基本データ CSV ファイルを作成する。
「桐 9」のテーブル群は「M:¥Sys¥systbl」（平成 22 年度）に、
CSV 形式のテキストファイル群は「M:¥Sys」（平成 22 年度）にある。
桐のテーブル名とテキストファイル名は原則、拡張子部分を除いて同名としている。
MySQL テーブル名との対応表は「表群」シートの列「I」「J」を参照。
- (2) 作成したファイル群を「表群」シートで指定されたフォルダに保存する。
「M:¥Excel¥make」フォルダの階層下「sys」「txt」にある。
- (3) 「表群」シートの「テキストからテーブルへの変換」ボタンでデータを併合する。
変換するもののセル（M 列）に「○」を付ける。変換されると「◎」に変わる。
レポート「subjrep2divs」・スクーリング「subjlsn2divs」・放送視聴「subjext2divs」などの学習記録は定義のみ（空欄）とし、この変換は使用しない。
- (4) 変更する場合は「表の変換シート」で行う。
 - 追加する場合
それぞれのテーブルに対応する CSV ファイルに追加するデータのみを保存し、「テキストからテーブルへの追加」ボタンをクリックする。
 - 削除する場合
それぞれのテーブルに対応する CSV ファイルに削除するデータのみを保存し、「テキスト一致データのテーブル削除」ボタンをクリックする。

3-5-3 運用のためのエクセル表

「M:¥Excel」フォルダに以下の4つのアプリケーションソフトがあるので各自のクライアントコンピュータの適当なフォルダにコピーし、「mySssr.xls」または「MySssr.vbs」を起動する。

- mySssr.xls：成績処理プログラムの本体で、フォームやシート群から構成。
- MySssr.vbs：起動時に Excel シートが見えるのを避けるための「ぼろ隠しプログラム」内部で「MySssr.xls」を読み込み、シートを隠してから起動しているスクリプト
- 掲示板表示用のウェブ HTML プログラム
- 成績構築.doc：構築運用全般のマニュアル

3-5-3-1 フォーム

「MySssr」メニューフォーム

起動時に表示されるメニューフォーム。「ログイン」「業務シート選択」などの全般を管理する。各業務シートの「メニュー」ボタンで表示される。また、マクロキー「Ctrl+M」でも表示される。

- ログオン
「ユーザ名」(ID) とパスワードを入力してログインする
「ホスト名」は「sssr」とし、平成22年度のデータベース名を「db_sssr22」とする。
- 業務処理ボタンをクリック
ログインが完了すると、「各業務選択」ボタンがアクティブになる。
- 「描画表示」
各シートの画面作成のときに逐次画面更新のための再描画を繰り返すと時間がかかってしまう。そこで、ボタンをクリックすると「描画非表示」に変わり、作成過程は見られなくなるが作成時間が大幅に短縮される。
もう一度クリックすると「描画表示」に戻る。初期値は「非表示」
- 「ツールバー表示」
各データ入力の際に上部のツールバーが表示されていると画面が狭く感じられる。
「ツールバー表示」をクリックすると、「ツールバー」が非表示になり入力部が広がる。
このバー部分は2007バージョン以降でリボンと呼ばれ、この操作はこのバージョンでのみ有効。これ以前のバージョンは、エクセルメニューの「表示」で設定できる。
クリックするたびに表示・非表示を繰り返す。初期値は「表示」
各業務シートのマクロキー「CTR+U」でも同様の処理ができる。
- 「プロファイル表示」
生徒の顔写真を表示。
初期値は「非表示」

3-5-3-2 シート

- (1) 「MySssr」シート
以前のメニューシート、現在は裏シートとして使用
 - 「ユーザ名」「パスワード」「ホスト名」(sssr)「データベース名」を入力し、ログイン。
 - ログインで認証されたら、「レポート」などの処理ボタンがアクティブになる。
独立して「メニューフォーム」を作成したのでこのシートは「メニュー」としては使用しない。ただし、基礎データの保存用「裏シート」として使用する。
- (2) 「レポート」シート
レポートの評価の入力
必ず最新のボタンをクリックしてから入力を始めること。以下のシート同様のこと。
入力ときは、割り込みによって各種の手助けをしてくれる「MySssr 動作中」にしておく。
「最新」ボタンを押すと自動的にこの状態にしてくれる。
この状態では「カーソル移動」や「セルの変更」などを感知し割り込み処理を行ってくれる。
通常の「エクセル」の状態にしたい場合はこのボタンを押して「MySssr 動作へ」にする。
- (3) 「スクリーニング」シート
面接時間の入力
- (4) 「放送視聴」シート
放送視聴や特別テストなど定期行事以外の活動記録
- (5) 「受講指導」シート
単位管理と来季受講予定科目の記録
- (6) 「レポ受付」シート
レポート受付のバーコード読み込み
- (7) 「記録簿」シート
学習記録簿の作成と印刷
- (8) 「科目単票」シート
科目の提出用単票の作成・印刷
- (9) 「成績一覧表」シート
クラスの成績・単位修得一覧表の作成・印刷
- (10) 「単位累計一覧表」シート
クラスの単位累計一覧表の作成・印刷
- (11) 「其他単位」シート
本校のカリキュラムにない教科の単位を管理
- (12) 「バーコード」シート
レポート提出用「登録番号」のバーコード作成と印刷
- (13) 「構成」(裏シート)
各シート選択メニューの基礎データ保存として使用
- (14) 「Welcome」
起動時の表示シートで、無用の用シート

3-5-4 実用モジュールの作成

3-5-4-1 アプリケーションの割り込み制御

(1) エクセルシートの非表示

```
Application.Visible = False
```

このシート非表示状態では、エクセル自身が終了しているのか、非表示状態で常駐しているのか判別できないので注意。タスクマネージャ（「CTRL」+「Alt」+「Delete」で起動）で「プロセス」を表示し、その中に「EXCEL.EXE」があったら、まだ常駐中。

(2) ツールバーの非表示

```
Application.ExecuteExcel4Macro "SHOW.TOOLBAR("""Ribbon""",false)"
```

リボンと呼ばれる「ツールバー」の部分を表示しないで画面を広く使うことができる。必要なときは「false」を「true」に変更する。

(3) 画面再描画の停止

```
Application.ScreenUpdating = False
```

画面のリフレッシュ表示がないので負荷が大幅に削減されて高速処理になる。

ただし、この状態では現在処理中の内容が確認されないので処理が進行中なのか機能停止の状態かは判別できない。

(4) 割り込みの許可

割り込みにはシステム設定とユーザ設定とがある。

① システム設定（ユーザが設定できない）

(1) シートがアクティブになったとき

```
Private Sub Worksheet_Activate()
```

② ユーザ設定（ユーザが制御できる）

```
Application.EnableEvents = True （開始）
```

```
Application.EnableEvents = False （終了）
```

(2) セルが変更されたとき

```
Private Sub Worksheet_Change(ByVal Target As Range)
```

(3) セルの選択位置が変更されたとき

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

3-5-4-2 MySQL コネクションオブジェクトを返す関数

「MySQL」サーバを使う場合は、接続のための手続きをしなくてはならない。各プログラムにおいて共通なので「サブルーチン」化しておくとう便利である。

```
Public Function MySQLConect(dbmode As Integer) As ADODB.Connection
    Dim con As ADODB.Connection
    Dim conStr As String
    Dim dbName, svr, usr, pw As String

    With Sheets("構成") ' 構成シートに「サーバ名、データベース名、ユーザ名、パスワード」設定
        svr = .Cells(3, 15)
        usr = .Cells(5, 15)
        pw = .Cells(6, 15)
        If dbmode = 2 Then
            dbName = "mysql"
        Else
            If dbmode = 1 Then
                dbName = .Cells(4, 15)
            Else
                dbName = ""
            End If
        End If
        conStr = "Driver={MySQL ODBC 5.1 DRIVER}; SERVER=" & svr & "; DATABASE=" & dbName
        conStr = conStr & "; USER=" & .usr & "; PASSWORD=" & .pw & ";"
    End With
    On Error GoTo Err
    Set con = New ADODB.Connection
    con.Open conStr

    Set MySQLConect = con

Exit Function

Err:
    Set con = Nothing
    Set MySQLConect = con
    MsgBox (Err.Description)

End Function
```

3-5-4-3 データベース名の取得

```
Public Function GetDBname() As String

    Dim dbName As String

    With Sheets("構成")
        dbName = .Cells(4, 15)
    End With
    GetDBname = dbName

End Function
```

3-5-4-4 テーブルの作成

(引数にテーブル名、警告メッセージ、作成クエリー文字列)

```
Public Sub CreateTable(tblName As String, msg As String, CreateSql As String)
```

```
    Dim row As Long
```

```
        ' 行変数
```

```
    Dim con As ADODB.Connection
```

```
    Dim rs As ADODB.Recordset
```

```
    Dim conStr As String
```

```
    Dim sqlStr As String
```

```
    Dim dbName As String
```

```
    Dim mybtn As Integer
```

```
    mybtn = MsgBox(msg, vbYesNo + vbQuestion, "作成")
```

```
    If mybtn <> vbYes Then End
```

```
    On Error GoTo Err
```

```
    Set con = MySQLConnect(1)
```

```
    sqlStr = "drop table if exist " & tblName & ";"
```

```
    Set rs = con.Execute(sqlStr)
```

```
        ' もし存在したらテーブルを削除する  
        ' SQL 文実行
```

```
    Set rs = con.Execute(CreateSql)
```

```
        ' 作成クエリーの実行
```

```
    sqlStr = "select * from " & tblName & ";"
```

```
    Set rs = con.Execute(sqlStr)
```

```
    For row = 0 To rs.Fields.Count - 1
```

```
        Cells(3 + row, 2).Value = rs.Fields(row).Name
```

```
    Next row
```

```
        ' 作成した項目を表示
```

```
    con.Close
```

```
    Set rs = Nothing
```

```
    Set con = Nothing
```

```
Exit Sub
```

```
Err:
```

```
    Set rs = Nothing
```

```
    Set con = Nothing
```

```
    MsgBox (Err.Description)
```

```
End Sub
```


3-5-4-5 テーブルの内容表示

項目名、その内容「200行以上は確認メッセージを表示」)

```

Public Sub TableDisp(tblName As String)

    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String
    Dim dbName As String

    Dim mybtn As Integer
    Dim num As Long
    Dim row As Long
    Dim col As Integer
    Dim item As Variant

    Range(Cells(2, 2), Cells(1000, 100)).ClearContents

    On Error GoTo Err
    Set con = MySQLConect(1)

    sqlStr = "select count(*) from " & tblName & ";"
    Set rs = con.Execute(sqlStr)
    num = rs.Fields(0)
    Cells(2, 2) = num
    If num > 200 Then
        mybtn = MsgBox("<" & num & "件全て読み込みますか?", vbYesNo + vbQuestion, "読み込み")
        If mybtn <> vbYes Then
            num = 200
        End If
    End If
    sqlStr = "select * from " & tblName & ";"
    Set rs = con.Execute(sqlStr)

    For row = 0 To rs.Fields.Count - 1
        Cells(3 + row, 2).Value = rs.Fields(row).Name
    Next row
    For col = 0 To rs.Fields.Count - 1
        Cells(2, 4 + col).Value = rs.Fields(col).Name
    Next col

    row = 3
    Do While rs.EOF = False
        If row > num + 10 Then Exit Do
        col = 4
        For Each item In rs.Fields
            Cells(row, col).Value = item.Value
            col = col + 1
        Next
        row = row + 1
        rs.MoveNext
    Loop

    con.Close
    Set rs = Nothing
    Set con = Nothing
Exit Sub

Err:
    Set rs = Nothing
    Set con = Nothing
    MsgBox (Err.Description)
End Sub

```

3-5-4-6 テキストファイルからテーブルへの変換

```
Public Function Text2Table(tblName As String, txtName As String, msg As String) As Integer
```

```

    Dim buff As String          ' 1行読み込み文字列
    Dim temp As Variant        ' 読み込み文字列の配列化
    Dim row As Long           ' 行変数
    Dim col As Long           ' 列変数
    Dim icnt, dcnt As Integer
    Dim typ As Integer
    Dim rcnt As Long          ' 行数

    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String
    Dim dbName As String

    Dim itype(100) As Integer
    Dim mybtn As Integer

    If msg <> "" Then
        mybtn = MsgBox(msg, vbYesNo + vbQuestion, "再構築")
        If mybtn <> vbYes Then End
    End If

    On Error GoTo Err
    Set con = MySQLConect(1)

    sqlStr = "select count(*) from " & tblName & ";"
    Set rs = con.Execute(sqlStr)
    rcnt = rs.Fields(0)
    If rcnt < 1 Then
        sqlStr = "select * from " & tblName & ";"
    Else
        sqlStr = "select * from " & tblName & " limit 1;"
    End If
    Set rs = con.Execute(sqlStr)
    icnt = rs.Fields.Count
    For col = 0 To rs.Fields.Count - 1
        typ = rs.Fields(col).Type
        Select Case typ
            Case 203
                itype(col) = 1
            Case 205
                itype(col) = 1
            Case 130
                itype(col) = 1
            Case Else
                itype(col) = 0
        End Select
    Next col

    sqlStr = "delete from " & tblName & ";"          ' 全データの削除
    Set rs = con.Execute(sqlStr)                    ' SQL 文実行

```

```

row = 3
Open ThisWorkbook.Path & txtName For Input As #1
Do Until EOF(1)
  Line Input #1, buff
  sqlStr = "insert into " & tblName & " value("
  temp = Split(buff, ",")          '1 行の文字列をカンマで切り分ける
  dcnt = UBound(temp) + 1
  For col = 0 To icnt - 1          ' 配列の最後まで繰り返す
    If col > 0 Then sqlStr = sqlStr & ","
    If col < dcnt Then
      sqlStr = sqlStr & "" & temp(col) & ""
    Else
      If itype(col) = 1 Then
        sqlStr = sqlStr & ""
      Else
        sqlStr = sqlStr & "-1"
      End If
    End If
  Next col
  sqlStr = sqlStr & ");"
  If row < 1000 Then
    Cells(row, 101) = sqlStr
  End If
  row = row + 1
  Set rs = con.Execute(sqlStr)    'SQL 文実行
Loop
Close #1

con.Close
Set rs = Nothing
Set con = Nothing

Text2Table = 0
Exit Function

Err:
Close #1
Set rs = Nothing
Set con = Nothing
MsgBox (Err.Description)

Text2Table = -1
End Function

```

3-5-4-7 テーブルからテキストファイルへの変換

```
Public Sub Table2Text(tblName As String, txtName As String)
```

```
    Dim con As ADODB.Connection  
    Dim rs As ADODB.Recordset  
    Dim conStr As String  
    Dim sqlStr As String  
    Dim dbName As String
```

```
    Dim mybtn As Integer  
    Dim num As Integer  
    Dim col As Integer  
    Dim row As Integer  
    Dim item As Variant  
    Dim buff As String
```

```
    On Error GoTo Err  
    con = MySQLConect(1)
```

```
    sqlStr = "select * from " & tblName & ";"  
    Set rs = con.Execute(sqlStr)
```

```
    Open ThisWorkbook.Path & txtName For Output As #1  
    buff = ""
```

```
    For col = 0 To rs.Fields.Count - 1  
        If buff <> "" Then buff = buff & ","  
        buff = buff & rs.Fields(col).Name
```

```
    Next col  
    Print #1, buff
```

```
    Do While rs.EOF = False  
        buff = ""  
        For Each item In rs.Fields  
            If buff <> "" Then buff = buff & ","  
            buff = buff & item.Value
```

```
        Next  
        Print #1, buff  
        rs.MoveNext
```

```
    Loop  
    Close #1  
    con.Close  
    Set rs = Nothing  
    Set con = Nothing
```

```
Exit Sub
```

```
Err:  
    Close #1  
    Set rs = Nothing  
    Set con = Nothing  
    MsgBox (Err.Description)
```

```
End Sub
```

3-5-4-8 定義されたテーブルの一覧表示

```
Private Sub DispTables_Click()
```

```
Dim con As ADODB.Connection  
Dim rs As ADODB.Recordset  
Dim conStr As String  
Dim sqlStr As String  
Dim dbName As String  
Dim tblName As String
```

```
Dim TblNas(100) As String
```

```
Dim t As Integer  
Dim col As Integer  
Dim row As Integer  
Dim rct As Integer  
Dim tent As Integer  
Dim typ As Integer
```

```
Range(Cells(2, 2), Cells(1500, 7)).Clear
```

```
On Error GoTo Err  
Set con = MySQLConect(1)  
dbName = GetDBname()
```

```
sqlStr = "show tables from " & dbName & ";"  
Set rs = con.Execute(sqlStr)
```

```
tent = 0  
Do While rs.EOF = False  
    TblNas(tent) = rs.Fields(0)  
    tent = tent + 1  
    rs.MoveNext
```

```
Loop
```

```
row = 2  
For t = 0 To tent - 1  
    tblName = TblNas(t)  
    Cells(row, 2) = 1 + t  
    Cells(row, 3) = tblName  
    sqlStr = "select * from " & tblName & " limit 1;"  
    Set rs = con.Execute(sqlStr)  
    For rct = 0 To rs.Fields.Count - 1  
        Cells(1 + row + rct, 3).Value = rct  
        Cells(1 + row + rct, 4).Value = rs.Fields(rct).Name  
        Cells(1 + row + rct, 6).Value = rs.Fields(rct).Type  
        typ = rs.Fields(rct).Type  
        Cells(1 + row + rct, 7).Value = rs.Fields(rct).DefinedSize  
        Select Case typ  
            Case 3  
                Cells(1 + row + rct, 5).Value = "integer"  
            Case 4  
                Cells(1 + row + rct, 5).Value = "float"  
            Case 20  
                Cells(1 + row + rct, 5).Value = "bigint"  
            Case 135  
                Cells(1 + row + rct, 5).Value = "datetime"  
            Case 203  
                Cells(1 + row + rct, 5).Value = "text"  
            Case 205  
                Cells(1 + row + rct, 5).Value = "blob"
```

システム構築の手引き

```
Case 130
    Cells(1 + row + rct, 5).Value = "char(" & rs.Fields(rct).DefinedSize & ")"
Case Else
    Cells(1 + row + rct, 5).Value = "不明? ?"
End Select
Next rct
With Range(Cells(row, 2), Cells(row, 7)).Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .Weight = xlMedium
End With
With Range(Cells(row, 2), Cells(row, 7)).Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .Weight = xlMedium
End With
With Range(Cells(row, 2), Cells(row + rct, 2)).Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
With Range(Cells(row, 2), Cells(row + rct, 2)).Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlHairline
End With
With Range(Cells(row, 7), Cells(row + rct, 7)).Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
With Range(Cells(row + rct, 2), Cells(row + rct, 7)).Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .Weight = xlMedium
End With
row = row + rct + 2
Next t

con.Close
Set rs = Nothing
Set con = Nothing
Exit Sub

Err:
Set rs = Nothing
Set con = Nothing
MsgBox (Err.Description)

End Sub
```

3-5-4-9 エクセル表で定義されたテーブル群の作成

```
Private Sub MakeTables_Click()
```

```
    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String
    Dim dbName, tblName As String
    Dim TblNas(100) As String
    Dim item As String
    Dim row, rw2, icnt As Integer
    Dim mybtn As Integer
```

```
    Range(Cells(2, 16), Cells(1500, 16)).ClearContents
    mybtn = MsgBox("全テーブルの作成をしますか?", vbYesNo + vbQuestion, "全作成")
    If mybtn <> vbYes Then End
```

```
    On Error GoTo Err
    Set con = MySQLConect(0)
    dbName = GetDBname()
```

```
    sqlStr = "create database if not exists " & dbName & ";"
    Set rs = con.Execute(sqlStr)          'SQL 文実行
```

```
    sqlStr = "use " & dbName & ";"
    Set rs = con.Execute(sqlStr)        'SQL 文実行
```

```
    For row = 2 To 1000
        tblName = Cells(row, 10)
        If tblName <> "" Then
            sqlStr = "drop table if exists " & tblName & ";"
            Set rs = con.Execute(sqlStr)    'SQL 文実行
```

```
            sqlStr = "create table " & tblName & "("
            icnt = 0
            For rw2 = row + 1 To 1000
                item = Cells(rw2, 11)
                If item = "" Then Exit For
                If icnt > 0 Then sqlStr = sqlStr & ","
                sqlStr = sqlStr & item & " " & Cells(rw2, 12)
                icnt = icnt + 1
            Next rw2
            sqlStr = sqlStr & ");"
            Cells(row, 16) = sqlStr
            Set rs = con.Execute(sqlStr)
            row = rw2 - 1
```

```
        End If
    Next row
```

```
    con.Close
    Set rs = Nothing
    Set con = Nothing
```

```
Exit Sub
```

```
Err:
    Set rs = Nothing
    Set con = Nothing
    MsgBox (Err.Description)
End Sub
```

3-5-4-10 テーブルのフィールド追加

```

Private Sub CbFieldAdd_Click()
    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim conStr As String
    Dim sqlStr As String

    Dim row As Integer
    Dim tblName As String
    Dim fieldName As String
    Dim dataType As String
    Dim value As String

    On Error GoTo Err
    Set con = MySQLConect(1)

    For row = 3 To 100
        tblName = Cells(row, 16)           ' テーブル名
        If tblName = "" Then Exit For
        fieldName = Cells(row, 17)        ' フィールド"
        dataType = Cells(row, 18)        ' データ型
        sqlStr = "alter table " & tblName & " add " & fieldName & " " & dataType & ";"
        Cells(row, 20) = sqlStr
        Set rs = con.Execute(sqlStr)      'SQL 文実行
        value = Cells(row, 19)           ' データ
        If value <> "" Then
            ' If dataType = "integer" Or dataType = "bigint" Or dataType = "float" Then
            '     sqlStr = "update " & tblName & " set " & fieldName & "=" & value & ";"
            ' Else
            '     sqlStr = "update " & tblName & " set " & fieldName & "=" & value & ";"
            ' End If
            Set rs = con.Execute(sqlStr)  'SQL 文実行
        End If
    Next row

    con.Close
    Set rs = Nothing
    Set con = Nothing
    Exit Sub

Err:
    Set rs = Nothing
    Set con = Nothing
    MsgBox (Err.Description)

End Sub

```


3-5-4-1 1 テーブルのフィールド削除

```
Private Sub CbFieldDrop_Click()  
    Dim con As ADODB.Connection  
    Dim rs As ADODB.Recordset  
    Dim conStr As String  
    Dim sqlStr As String  
  
    Dim row As Integer  
    Dim tblName As String  
    Dim fieldName As String  
  
    On Error GoTo Err  
    Set con = MySQLConect(1)  
  
    For row = 3 To 100  
        tblName = Cells(row, 16)           ' テーブル名  
        If tblName = "" Then Exit For  
        fieldName = Cells(row, 17)        ' フィールド"  
        sqlStr = "alter table " & tblName & " drop " & fieldName & ";"  
        Cells(row, 20) = sqlStr  
        Set rs = con.Execute(sqlStr)      'SQL 文実行  
    Next row  
  
    con.Close  
    Set rs = Nothing  
    Set con = Nothing  
    Exit Sub  
  
Err:  
    Set rs = Nothing  
    Set con = Nothing  
    MsgBox (Err.Description)  
End Sub
```

3-6 運用マニュアル

3-6-1 環境設定

3-6-1-1 データベース「MySQL」の設定

- (1) フォルダ「M:¥mysql」の中の実行ファイル「mysql-connector-odbc-5.1.6-win32.ms」をダブルクリックして、プログラムを起動する。
- (2) 「次へ」(Next) のボタンを押し続け、最後に「終了」(Finish) のボタンを押す。

3-6-1-2 必要なファイルのコピー

- (1) 各自のコンピュータの中にフォルダを作成
(例) ドライブ「D :」にフォルダ「MySssr」を作成
- (2) フォルダ「M:¥Excel」の中の3つのファイルを作成したフォルダの中にコピーする。
 - 「mySssr.xls」(本体)
 - 「MySssr.vbs」(実行時に Excel シートが表示されるのを嫌う者はこれを使用)
 - 「bbs.htm」(掲示板表示のための Web ページ)
- (3) 実行ファイルの「ショートカット」を作成
 - コピーした「mySssr.xls」または「MySssr.vbs」ファイルの上で右クリックする。
 - メニューで「送る」→「デスクトップ (ショートカットを作成)」を選択する。

3-6-2 「MySssr」の起動

- (1) 作成したショートカットをダブルクリックする。
システムが大きく重たいので、起動にちょっと時間がかかる。何度もダブルクリックして多重起動にならないように注意する。確実な起動の仕方は、ショートカットの上で右クリックし、ポップメニューの一番上の項目「開く(O)」を選ぶ。
- (2) 「MySssr.vbs」の方を起動したとき、デスクトップにエクスプローラなどを表示していると裏に隠れて画面が見えない場合がある。

また、Excel ワークシートを隠した状態の起動なので何かのトラブルがあった場合に Excel が終了しているのか、まだ常駐しているのか判別ができないので注意。

(トラブルシューティング)

このようなとき、「タスクマネージャ」で確かめることができる。

「Ctrl+Alt+Delete」のキーを同時に押し、「タスクマネージャ」を起動。「プロセス」タブを選び、リストの中に「EXCEL.EXE」があったら、まだ終了していないことになる。その場合は「EXCEL.EXE」プロセスを選択し「プロセスの終了」を実行する。

3-6-3 「MySssr」メニュー

- (1) メニュー画面で、「MySssr」用の「ID」「パスワード」を入力して、ログインする。
起動画面がメニュー画面であり、各処理に活用する。
各シートの「メニュー」ボタンか、マクロキーの「Ctrl+M」で呼び出すことができる。
- (2) 「描画表示」ボタン
各シートの画面作成のときに逐次画面更新のための再描画を繰り返すと時間がかかってしまう。そこで、ボタンをクリックすると「描画非表示」に変わり、作成過程は見られなくなるが作成時間が大幅に短縮される。
もう一度クリックすると「描画表示」に戻る。初期値は「描画非表示」
- (3) 「ツールバー表示」
各データ入力の際に上部のツールバーが表示されていると画面が狭く感じられる。
「ツールバー表示」をクリックすると、「ツールバー」が非表示になり入力部が広がる。
このバー部分は2007バージョン以降でリボンと呼ばれ、この操作はこのバージョンでのみ有効。これ以前のバージョンは、エクセルメニューの「表示」で設定できる。
クリックするたびに表示・非表示を繰り返す。初期値は「表示」
各業務シートのマクロキー「Ctrl+U」でも同様の処理ができる。
- (4) 「プロフィール表示」
生徒の顔写真を表示。初期値は「非表示」

3-6-4 入力シート

メニュー上部の「レポート」などのボタンをクリックすると、入力画面が開く。
必ず「最新」のボタンをクリックしてから入力を始めること。
各シートとも同様の操作を行う。
入力ときは、割り込みによって各種の手助けをしてくれる「MySssr 動作中」にしておく。「最新」ボタンを押すと自動的にこの状態にしてくれる。この状態では「カーソル移動」や「セルの変更」などを感知し割り込み処理を行ってくれる。
通常の「エクセル」の状態にしたい場合は、「MySssr 動作中」ボタンを押して「MySssr 動作へ」に変更する。

3-7 「MySQL」をPHPで操作する

3-7-1 ユーザの登録

3-7-1-1 全ての権限を持つユーザの追加

```
<?php
    header("Content-Type: text/plain; charset=UTF-8");
    $id = "yayohi";
    $pw = "abcd";
    $db = mysql_connect('localhost','root','');
    $str_sql = "grant all privileges on *.* to '{$id}'@%' identified by '{$pw}';";
    $result = mysql_query($str_sql,$db);
    mysql_close($db);
?>
```

3-7-1-2 制限付きのユーザ追加

上記プログラムの下線部分を以下のように書き換える

```
$str_sql = "grant select insert update delete on *.* to '{$id}'@%' identified by '{$pw}';";
```

3-7-1-3 パスワードの設定

上記プログラムの下線部分を以下のように書き換える

```
$str_sql = "set password for '{$id}'=password('{$pw}');";
```

3-7-2 ユーザ&パスワードの認証

```
<?php
    header("Content-Type: text/plain; charset=UTF-8");
    $id = $_POST["id"];
    $pw = $_POST["pw"];
    $db = mysql_connect('localhost',$id,$pw);
    mysql_query($sql, $db) or die(" I D&パスワードの確認 ! ");
    mysql_close($db);
?>
```

4 システム管理

4-1 環境の構築

4-1-1 リモート管理の設定

4-1-1-1 ターミナルサービスクライアント

この設定はサーバにリモートログインするためのものです。

- (1) サーバフォルダ「APP」(Z:\Lan\Svr_term\Disk1)の「SETUP.EXE」を実行する。
- (2) 各サーバをクライアントとして設定

サーバ名	I Pアドレス	サービスの種類
Server	192.168.0.1	「PDC」「DNS」「DHCP」「ファイル」
server2	192.168.0.2	「SDC」「DNS」
sssr	192.168.0.3	「MySQL」「WWW」「J a v a」

- (3) 「MySssr」データのバックアップ
 - ① 「sssr」サーバにリモートログインする
 - ② 「MySssr へのショートカット」をダブルクリックして、保存フォルダを開く
 - ③ 「db_sssr22.」フォルダ名を変更（末尾に年月日時情報を追記）
 - ④ 「cpDB2sys.bat へのショートカット」ダブルクリックして「MySssr」データを保存。
- (4) 「MySssr」の時期ステージ変更
 - ① 「MySssr」を起動する
 - ② メニュー「環境」の「(SU) スーパーユーザへ」で、管理者権限へ移行する
 - ③ メニュー「環境」の「学期時期の変更」でステージを進める。

5 研修

5-1 実習内容

5-1-1 Excel「VBA」の設定とマクロ作成

「2-2」の成績処理プログラム作成練習

- (1) 「2-2-1」の表作成
- (2) 「2-2-2」の「乱数による点数の入力」マクロ作成
- (3) 「2-2-3」の「成績の統計処理」マクロ作成

5-1-2 MySQLの環境設定と「VBA」からの操作

今回設定するのは「MySQL」と「ODBC」です。

- (1) 「3-3-4」の「MySQL環境」設定
- (2) 「3-3-5」の「ExcelからのMySQL操作環境」設定
- (3) 「3-3-4」の「MySQL成績テーブル」作成
- (4) 「3-3-5」の「MySQLからExcel表変換」マクロ作成

5-2 制作課題と目標

5-2-1 (課題1) 40人学級の9科目の成績処理システムの作成

5科目4人の「簡易成績処理プログラム」の拡張版を作成する。

- (1) 項目
出席番号、氏名（全角7文字）
国語、社会、数学、理科、保健、体育、芸術、英語、家庭、（9科目）
合計、平均、順位
- (2) 氏名は「生徒台帳.tbl」より適宜40人分コピーする
- (3) 点数は「0~100」の整数を乱数で発生させる。
- (4) 縦の合計・平均・最高・最低を求める。

5-2-2 (課題2)

「課題1」の成績一覧表を「テキストファイル」として保存する。
「保存ボタン」と「読み込みボタン」を作成する

5-2-3 (課題3)

「課題1」の成績一覧表を「MySQLデータベース」として保存する。
「保存ボタン」と「読み込みボタン」を作成する