

在庫管理の自動化を目指したオンラインショップシステムの研究開発

— 経営情報システムへの応用を目指して —

伊藤孝夫* 紺田温子** 恋河内敦***

A research on the on-line shop system with automation of stock control

Takao ITO* Atsuko KONTA** Atsushi KOIKAWACHI***

Abstract : Number of on-line shop has increased rapidly with the diffusion of the recent network technology. The automation of stock control is one of the important issues for on-line shops. How to develop the on-line shop system that utilizes the automation of stock control is necessary with lower cost and easy expansion for the system developer. This paper strengthens that management function such as information retrieval by using the Visual Basic language and the data base technology is an important factor in developing Data Base Management System. The automation of stock control is also realized by introducing the probability model into the data base management system in this paper. The basic technology that introduces the bar cord system has been completed and the improvement of the system efficiency has been adopted. Furthermore, the 5 digits commodity code and visual function such as graph are introduced in the on-line shop system.

Key words : automation of stock control, commodity classification, order automation

1. はじめに

近年、インターネットが普及し、実店舗よりも高速かつ低コストで業務を行うことができるなどの利点から、インターネットを経由して販売業務を展開するオンラインショップの数が増加し続けている。このようなオンラインショップの効率を向上させるためには、販売業務をはじめ、在庫管理の自動化を実現することが不可欠である。現在、多くのパッケージソフトが販売されている。オンラインショップパッケージのソフトとして、e-pos レジアプリケーショングラスプやガメディオス・イーシー (GAMEDIOS-EC) などがあげられる¹。前者は低価格という特徴を持っているが、データベースを利用していないので商品販売システムのみとなっている。後者はより豊かな機能を充実しているが、「受注案件からの再注文機能」、

「ロットによる割引機能」などの機能や在庫管理機能はそなえていない²。そのため、小規模を主流とするオンラインショップにとってかならずしも利用しやすい環境とはいえない。したがって、低コスト・高性能かつ豊かな拡張性をもつオンラインショップシステムの構築がもっとも重要な課題であることは言うまでもない。本研究は、このようなオンラインショップの問題点を考慮しながら、オンラインショップの在庫管理の自動化とシステム全体の効率向上に焦点をおき、小規模のオンラインショップが利用しやすいシステムの研究開発を目的としている。

2. オンラインショップシステムの概要

本研究では、店頭販売のスペースをもつことなく、インターネット上のホームページで商品を紹介し、購入の申し込みを受けて販売する形態のショップを、オンラインショップと呼ぶことにする。オンラインショップシステムは基本的に商品販売システムとデータベース管理システムの2つに分けることができる。

(2003年12月5日 受理)

* 宇部工業高等専門学校経営情報学科

** 宇部工業高等専門学校経営情報学科5年

*** 宇部工業高等専門学校技術室第二技術班

オンラインショップにおける販売から在庫管理までの情報は次のように8種類に分けることができる。

種類	内容
1 商品情報	商品名、単価、生産者などの情報
2 顧客情報	顧客の氏名、年齢、性別、住所などの情報
3 注文情報	注文商品名、商品数量、注文日、出荷予定日などの情報
4 発注情報	仕入先企業への発注書品名、発注数量、納品予定日などの情報
5 入荷情報	発注商品の入荷日、発注伝票番号などの情報
6 出荷情報	商品の出荷日、受注伝票番号などの情報
7 売上情報	商品の販売数量、販売金額などの情報
8 在庫情報	商品実在庫数量、在庫維持コスト、商品の調達期間などの情報

表1 オンラインショップシステムにおける情報の分類

本研究におけるオンラインショップシステムの流れは次のようになっている。まず、クライアントがサーバにアクセスすると、オンラインショ

ップのトップ画面を閲覧することができる。顧客が会員の登録作業を行うと顧客情報として、商品を注文すると注文情報として、データベースに格納することになる。担当者が注文情報を受けると、顧客に商品を出荷する。顧客への商品の受け渡しは物流会社に委託することとし、代金は商品と引き換えに受け取ることにする。セキュリティ問題を考慮していないため、銀行振り込みやクレジットカードなどの決済方法をとらず、配達による代金の引き換えシステムとする。商品の在庫数量が発注点より少なくなった場合には、仕入先企業に自動的に発注情報を送付する。仕入先企業からは入荷情報と商品情報が送信される。

顧客と商品販売システムとの間に発生するすべての情報は自動的にデータベースに格納され、必要に応じてデータベース管理システムや商品販売システムを用いて操作することができる。その全体像を図で示すと次のようになっている。

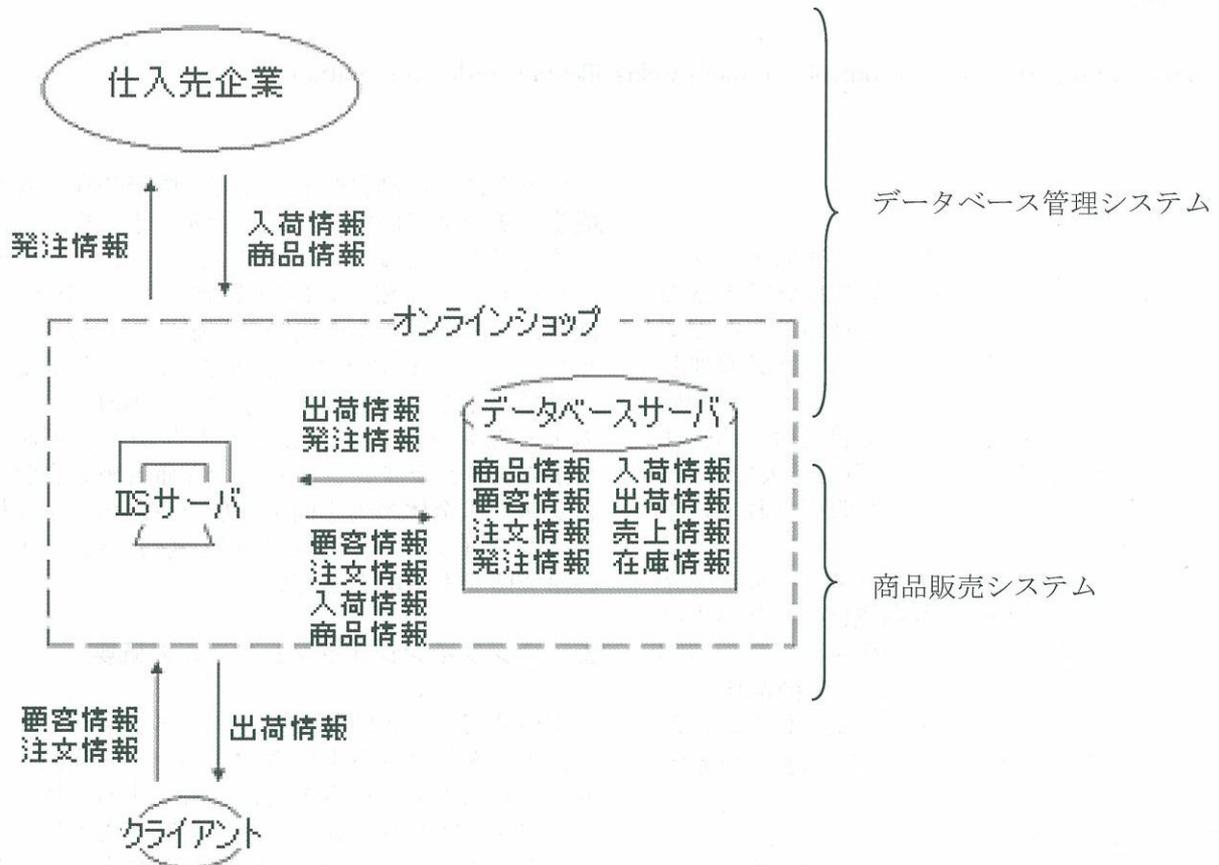


図1 オンラインショップシステムの概要

本研究では、データベース管理システムを取り上げて説明する。（商品販売システムについては、宇部工業高等専門学校研究紀要第 50 号を参照）

3. データベース管理システムの特徴

本研究におけるデータベース管理システムの特徴は次の 5 つにまとめることができる。

3-1 商品区分の拡張が可能

このシステムでは、商品区分が拡張できるように商品区分を設計している。商品区分は 2 桁表示とし、これを商品区分 ID と呼ぶことにする。商品 ID は 5 桁から構成されており、最初の 2 桁に商品区分 ID を付け加えることにより、商品 ID を見るだけで、商品区分が区別できるように設計している。

JAN コードのコード体系には分類という概念はなく、絶対番号の国のコード（2 桁）、商品メーカーコード（5 桁）、商品アイテムコード（5 桁）およびチェックデジット（1 桁）によって構成されている。そのため、本稿の商品 ID は JAN コードにも対応でき、拡張性をもつものと考えられる。

この商品区分は、新商品登録画面で追加することができる。また、新商品を登録する時に、商品 ID を自動表示できるように設計している。

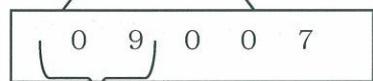
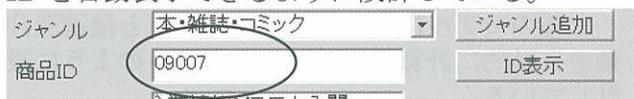


図 2 商品 ID 表示画面

3-2 発注作業の自動化

発注作業をするフォームは、新商品の場合とそうでない場合の 2 つに分けて設計している。図 3 のように発注伝票を作り、【伝票プレビュー】ボタンをクリックすると、発注伝票の内容を確認するためにプレビュー画面が図 7 のように表示される。ここで保存することにより、メール送信システムで添付ファイルとして利用することができる。【伝票送付】ボタンを押すと、メール送信システムが開き、仕入先企業に発注伝票を添付ファイルとして送ることができる。

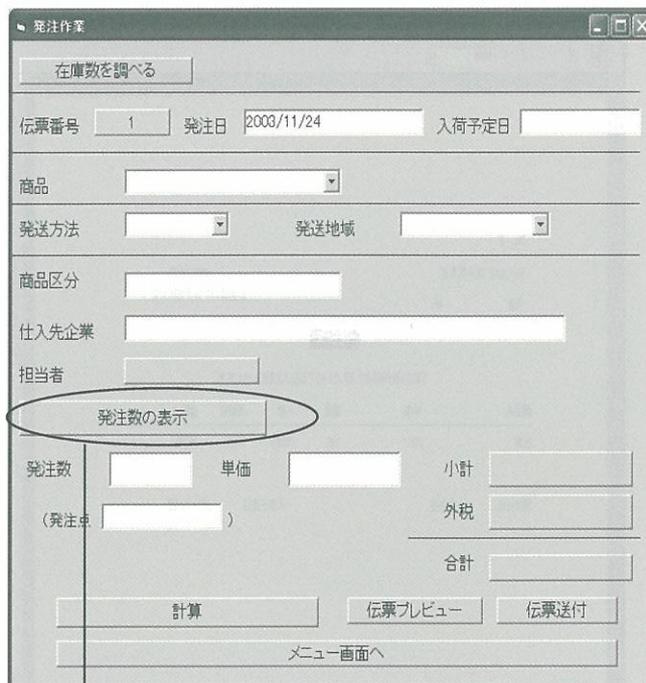


図 3 発注伝票（新商品の場合）

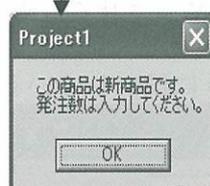


図 4 発注数確認画面

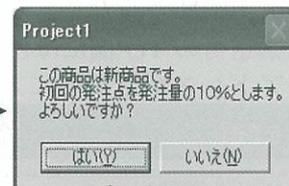


図 5 発注量確認画面



図 6 発注確認画面

発注作業を行う際に問題点の 1 つとして、発注のタイミングが考えられる。本研究では発注するタイミングは次のように設計している。

- ① 商品を登録したとき
- ② (実在庫数 - 予約数 + 予約発注数) が発注点より少なくなったとき
- ③ (1 日の売上数 + 予約数) が (実在庫数 - 予約数 + 予約発注数) より少なくなった時

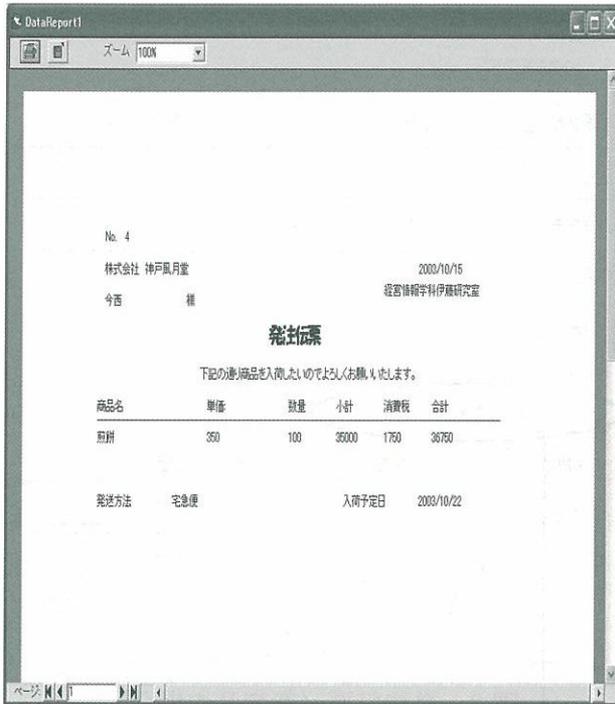


図7 発注伝票プレビュー

入力時間を少なくし、ミスを減少させるために、伝票送付の TO 欄には仕入先テーブルの企業メールアドレスが自動的に入るように設計している。

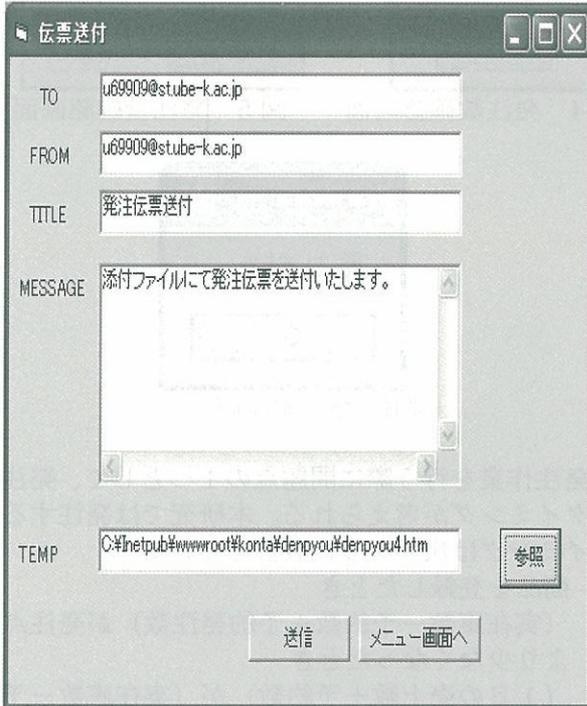


図8 発注伝票送付画面 (メール送信システム)

3-3 在庫管理モデルの導入

本研究では、いろいろな種類の商品の扱いを考えているので、それぞれの調達期間が異なるものであると考えられる。したがって、最適発注点と最適発注量を求めるために調達期間が一定でな

い場合の双関在庫方式を利用する。これは、ストックが連続的に観察され、ある在庫水準 R (発注点) に達したとき商品が発注されるような購入・販売在庫の確率モデルである。

商品の在庫管理は需要や商品調達期間を一定とみることはできず、これらが確率的に変化することを前提とする必要がある。以上より、この双関在庫方式モデルを使用している。

その計算式は次のとおりである。

$$R^* = T\mu + \sqrt{T}\varepsilon(\varepsilon)\sigma \quad (1)$$

$$y^* = \sqrt{\frac{2YK}{h}} \quad (2)$$

式の中各パラメータの意味は以下のとおりである。

R^* : 最適発注点

T : 調達期間

μ : 平均

$\varepsilon(\varepsilon)$: 在庫不足が生じる確率 (5%とする = 1.96)

σ : 調達期間の需要量の標準偏差

y^* : 最適発注量

Y : 年間気体需要量

K : 発注コスト (1回当たり)

h : 在庫維持費用 (1単位当たり)

上述のモデルに従って、最適発注点と最適発注量を自動的に計算するフォームを図9のように設計している。

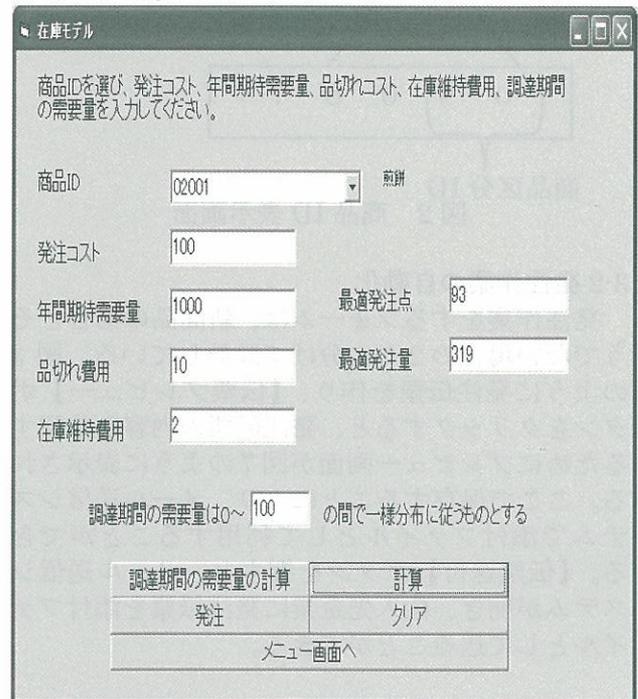


図9 在庫モデル

商品 ID をコンボボックスから選択すると、OR 在庫テーブルに格納されている、発注コスト、年間期待需要量、品切れ費用、在庫維持費用がそれぞれのテキストボックスに表示される。調達期間の需要量は【調達期間の需要量の計算】ボタンをクリックすると、自動的に計算されテキストボックスに表示される。これらの作業を終えて、【計算】ボタンをクリックすると、最適発注量と最適発注点が自動的に計算され、テキストボックスに表示される。そして、【発注】ボタンをクリックすると図 3 のような発注作業の画面が開かれる。

3-4 売上状況表示機能

データベースの情報を用いて、顧客別の嗜好性、時刻別・客層別の売れ行き分布、品目別売れ行き分布など次の経営計画に生かせる戦略および戦術情報を継続的に収集・分析する必要がある。

商品の売れ行きの傾向などを知るために売上状況をグラフで見ることができるよう設計している。グラフの種類は棒グラフで、図 10 の総売上商品個数の月間累計、図 11 の総売上金額別の月間累計を表すグラフのほかに、月別の商品別売上個数の月間累計、商品別売上金額の月間累計を表すグラフ、商品区分別売上個数の月間累計、商品区分別売上金額の月間累計のグラフを設計している。

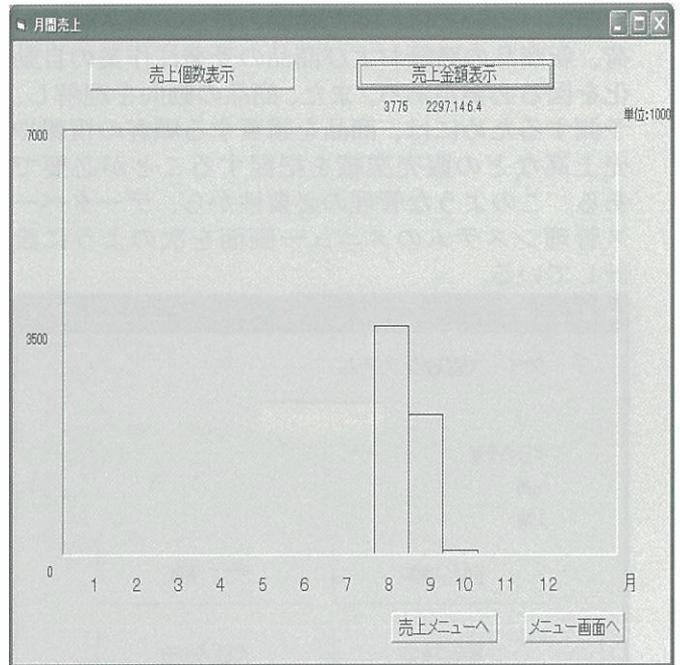


図 11 総売上金額の月間累計表示

3-5 更新、削除履歴テーブル

商品情報や顧客情報が変わったり、削除されたりした場合、完全に前のデータを削除するのではなく、別のテーブルに履歴を残すことにしている。このテーブルを履歴テーブルと呼ぶ。こうすることにより、削除、更新後のデータと、もとのデータを比較することができるようになった。

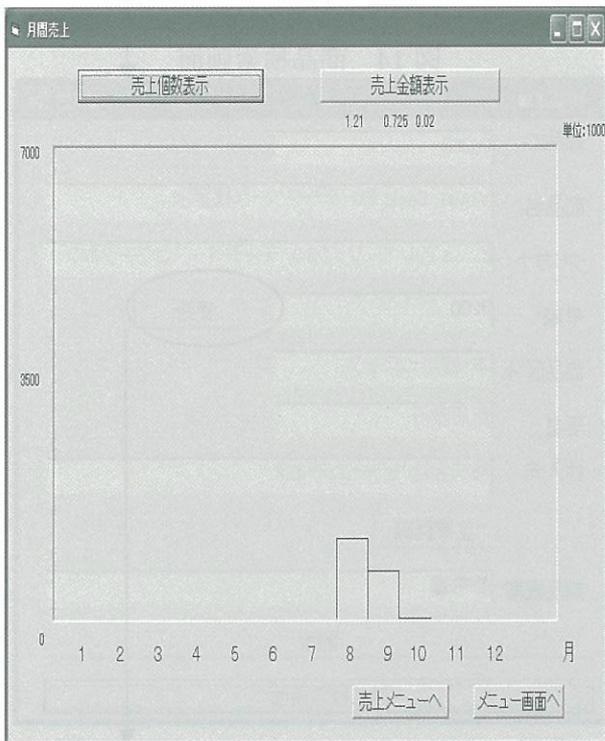


図 10 総売上商品個数の月間累計表示

履歴ID	発生日	テーブル名	フィールド名	レコードID	更新前	更新後
1	2003/01/07	商品マスター	単価	09002	250	2500

図 12 履歴テーブル

4. データベース管理システム

データベース管理システムとは、商品販売システムによりネットワークを経由して登録され、データベースに格納された顧客情報、商品情報、注文情報、発注情報、入荷情報、出荷情報、売上情報、在庫情報を管理するシステムである。

企業は在庫管理をする際には、在庫の品切れのほかに、在庫維持費用の合理化、新商品の調達に伴う登録問題などに直面する。本研究では、これらの問題を解決するために、商品、顧客および受発注の側面からデータベース管理システムの設計を行った。

商品については、商品検索のほか、商品の出荷、新商品の登録および商品の受発注作業の自動化を図る必要がある。また、商品の動向を理解し、予測するためには、商品を購入する顧客の情報や売上高などの販売業績を把握することが必要である。このような管理の必要性から、データベース管理システムのメニュー画面を次のように設計している。

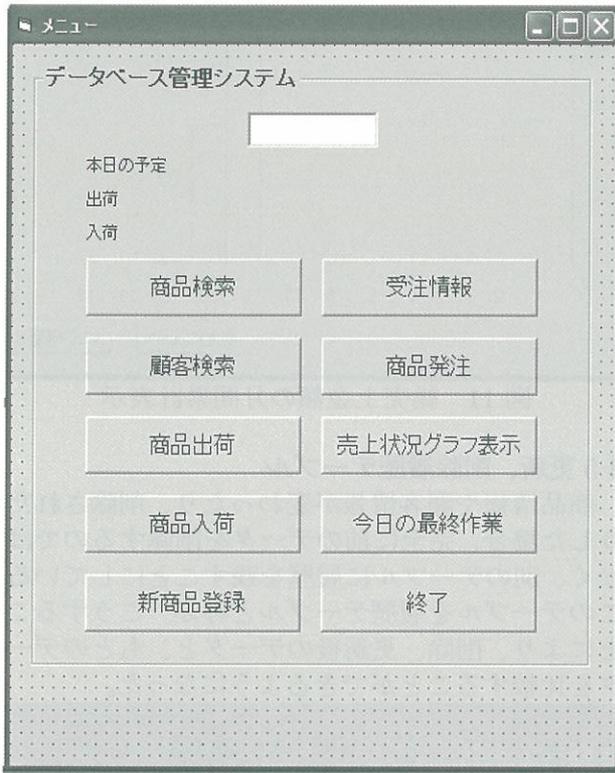


図13 データベースのメニュー画面

4-1 商品検索機能

商品情報を管理し、確認するために商品を検索できる機能が必要である。メニュー画面の【商品検索】ボタンをクリックすると、商品名の一覧を表示する画面を図14のように設計している。これは、ActiveX Data Objects データコントロール (ADODC)、データグリッドコントロールを使用し、表示している。商品名の一覧の中から、検索したい商品名を選び、【商品詳細画面へ】ボタンをクリックすると、商品詳細画面が開かれ、その商品の詳細を見ることができる。

図15の商品詳細画面からわかるように、商品情報の更新、削除機能が設計されている。商品の価格は経営情勢の変化とともに変化するので、いつでも対応できるように更新機能を追加した。価格を更新すると、その変化の過程を把握するために、更新に関する情報が自動的に履歴テーブルに格納される。商品がなくなった場合は、その商品

を選択して【削除】ボタンをクリックすればよい。そうすると、商品マスターの削除フィールドにチェックがつき、その商品は削除されたと見なされることになる。また、仕入先企業の詳細を確認したいときは、【企業詳細】ボタンをクリックすると、仕入先企業詳細画面が図16のように開かれる。仕入先企業情報は、住所や担当者、URLなど変更する可能性があることから、【更新】ボタンを追加している。それをクリックすることにより、更新に関する情報が商品詳細画面と同じように履歴テーブルに格納される。

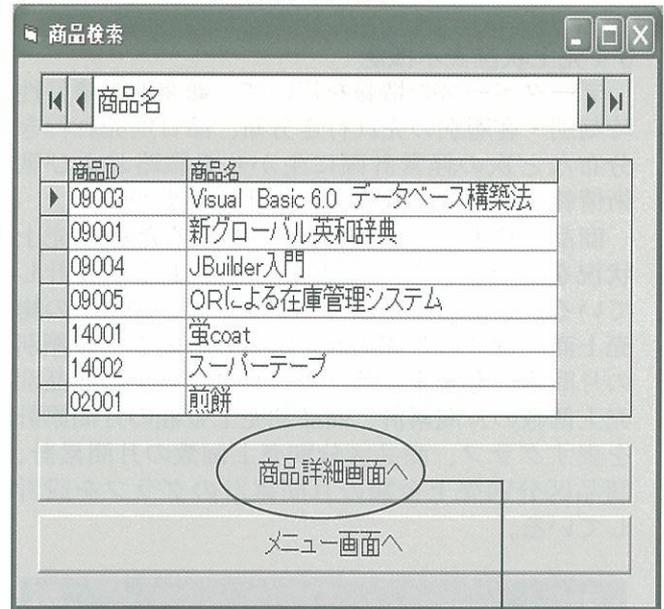


図14 商品検索画面

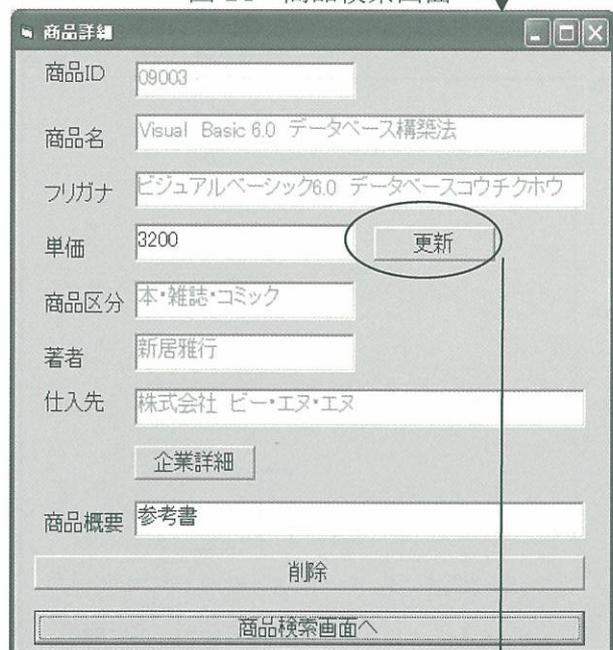


図15 商品詳細画面

図16へ

図 15 から

16 企業詳細画面

4-2 顧客検索機能

ネットワークを経由して登録された顧客情報を管理し、確認するために顧客を検索できる機能が必要である。そこで、この画面を図 17 のように設計している。商品検索画面と同様に、ADODC、データグリッドコントロールを使用し、顧客の一覧をデータグリッドに表示する。データグリッドコントロールの顧客名の中から、検索したい顧客名を選び、【顧客詳細画面へ】ボタンをクリックすると、図 18 の顧客詳細画面が開かれる。顧客情報は、顧客名、フリガナ、郵便番号、住所、ビル・マンション名、電話番号、携帯番号、性別、誕生日、メールアドレスを取り上げている。名前や住所などは、顧客に連絡する時や、商品を発注する時に必要最低情報となるので必ず得なければならない。そして、生年月日を得ることにより、年齢を知ることができ、その年齢と性別は、顧客嗜好を知るための情報として利用することができる。Eメールアドレスは、企業が何らかの情報を送るときに手軽に伝えることができるため、記入してもらうように設計している。

図 17 顧客検索画面 クリックする

図 18 顧客詳細画面

4-3 商品出荷機能

図 19 は、商品を顧客に出荷する時に使用するフォームである。あらかじめ注文を受けた商品に対しての処理を行う。受注伝票 ID を選択すると、顧客 ID、商品 ID、注文数が自動的に表示される。

【出荷】ボタンをクリックすると、その商品の OR 在庫テーブルの「実在庫数」が注文数だけ減少する。出荷に関する情報は自動的に出荷テーブルに格納される。

図 19 商品出荷画面

4-4 商品入荷機能

図 20 は、商品の入荷を処理するとき使用するフォームである。あらかじめ発注しておいた商品に対しての処理を行う。発注伝票 ID を選択すると、商品 ID と入荷数が自動的に表示される。

【入荷】ボタンをクリックすると、その商品の OR 在庫テーブルの「実在庫数」が注文数だけ増える。入荷に関するデータは入荷テーブルに格納される。

図 20 商品入荷画面

4-5 発注伝票の入力機能

図 21 は、在庫がなくなり仕入先企業に注文するとき必要とするフォームである。在庫数は【在庫数を調べる】ボタンをクリックすると自動的に確認され、発注点に達している商品を知らせ

てくれるよう設計している。必要項目をそれぞれのテキストボックスに入力し、【計算】ボタンをクリックすると、小計、外税、合計を自動的に計算することができる。例えば、発注が必要な商品が3件ある場合、商品 ID を選ぶコンボボックスには3件のデータが入り、1件を選択する。そして、金額を計算し発注作業が終了すると、コンボボックスのデータは2件になる。コンボボックスにデータがなくなった状態で発注作業を終了する。

発注数	100	単価	3000	小計	300000
(発注点)	10			外税	15000
				合計	315000

図 21 発注作業画面

図 22 発注確認画面

図 23 発注数確認画面

図 24 へ

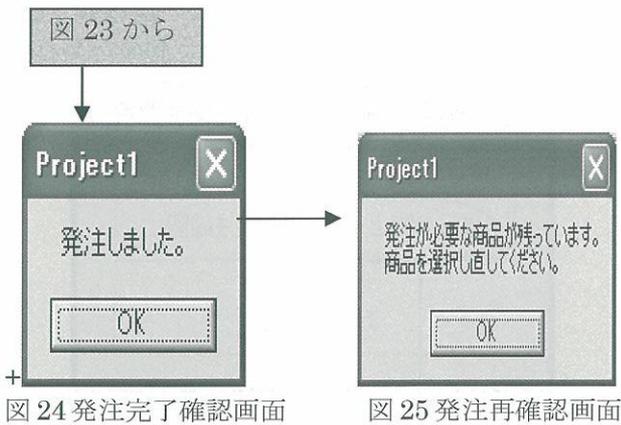


図 24 発注完了確認画面

図 25 発注再確認画面

4-6 新商品登録機能

図 26 は、新商品を登録するためのフォームである。登録情報には、商品の名前、価格などが含まれる。これらの情報は、【新商品登録】ボタンをクリックするとデータベースに登録される。新商品を登録した後は、その商品を入荷する必要があるため、発注作業画面が自動的に開くように設計している。

本研究は、商品区分の拡張が可能のため、商品区分を追加する機能が必要となる。また、本の著者や仕入先企業も追加する機能が必要である。そこで、それぞれに対して追加ボタンを追加し、図 27、図 28、図 29 の画面で追加することができるように設計している。

図 26 新商品登録画面

図 29 へ

図 27 へ

図 28 へ

図 27 仕入先企業追加画面

図 28 商品区分追加画面

図 29 著者追加画面

4-7 受注情報表示機能

オンラインで登録された受注情報を確認するためのフォームである。注文を受けて、まだ顧客に商品を出荷していない伝票のみ確認できる。受注の伝票番号を選択すると、その伝票の情報が出てくる。そして、【受注詳細画面へ】ボタンをクリックすると受注伝票で表示される伝票番号の

詳細な情報を図31のように確認することができる。

図30 受注伝票画面

図31 受注詳細画面

4-8 売上状況グラフ表示機能

月ごとの売上状況の変化をグラフで見るときのフォームである。その月の総売上数、商品別売り上数と商品区分別売上数の売上状況を見ることができる。その詳細については、第3節のシステムの設計特徴で説明してある。

4-9 本日の最終作業

図32は、毎日の最終作業メニュー画面である。1日の終わりにその日の売上を集計する必要があるため、売上状況を見るフォームを図33のように設計している。このフォームでは、その日に販売した商品の情報が表示される。この商品情報は【データベースへ】ボタンをクリックすると自

動的に売上テーブルに格納される。そして、【在庫確認】ボタンをクリックすると、自動的に在庫確認が行われ、発注点に達している商品があれば、図35のように知らせてくれるよう設計している。

図32 最終作業メニュー画面

クリックする

図33 売上状況表示画面
クリックする

図34 売上状況登録確認画面

図35 在庫確認画面

5. おわりに

今回の「在庫管理の自動化を目指したオンラインショップシステムの構築」では、発注伝票レス化と在庫モデルを導入することにより発注作業の自動化作業を実現することに成功した。しかしながら、商品と顧客のキーワード検索機能やセキュリティなどの問題は解決されておらず、今後の課題として引き続き研究していく必要がある。

謝辞：今回の研究は伊藤研究室の4期生幸山孝明の「オンラインショッピング構築の研究」、6期生吉倉聖子の「バーチャルモールシステムの研究開発」、7期生坪井志津と藤岡香織の「オンラインショップシステムの構築について」を参考して

作成したものである。

【参考文献】

- [1] Ryan K. Stephens, Ronald R Plew, Bryan Morgan, Jeff Perkins : Teach Yourself SQL in 21 Days Second Edition, Sams Publishing, 1997 木村博文監修 : 『SQL プログラミング入門』 ソフトバンク株式会社 (1998)
- [2] 新居雅行 : 『Visual Basic 6.0 データベース構築法』 株式会社エクシードプレス (1999)
- [3] 北原貞輔・児玉正憲 : 『OR による在庫管理システム』 九州大学出版会 (1982)
- [4] 流通システム開発センター : 『EDI の知識』 日本経済新聞社 (1997)

注釈

- 1 前者のパッケージソフトは有限会社アウイッシュによって開発され、その価格は 39,800 円であり、サーバー使用料として月額 2,000 円となっている。詳しくは e-pos レジアプリケーションングラspb (<http://www.grasp.jp/index.html>) を参照。後者は凸版印刷株式会社によって開発され、初期費用を 550 万円、保守費用を約 120 万円必要としている。
- 2 B to B 用機能としての「受注案件からの再注文機能」、「ロットによる割引機能」、「取引先単位の価格設定」、「在庫／納期回答機能」や (オプション設定の) B to B to C 用機能としての「取引先が上代を設定し、店舗の集合体での EC 機能」などはこれから搭載予定である。詳しくは凸版印刷株式会社 (<http://ecp.gamedios.com/gamediou-ec/>) を参照。

付 録 (プログラミングリスト)

【発注伝票画面】

```
Private cn As ADODB.Connection
Private rs, rs2, rs3, rs4, rs5 As ADODB.Recordset
Private mySQL, mySQL2, amount, amount2 As String
```

```
Private Sub Form_Load()
```

```
    'データベースに接続
    Set cn = New ADODB.Connection
    cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & "Data
    Source=C:\inetpub\wwwroot\konta.mdb;"

    '発注伝票 ID を自動的に表示する
    mySQL = "SELECT 発注伝票 ID FROM 発注伝票テーブル
    ORDER BY 発注伝票 ID DESC"
    Set rs = cn.Execute(mySQL, , adCmdText)
    If rs.EOF Then
        Command1.Caption = 1
    Else
        Command1.Caption = rs!発注伝票 ID + 1
    End If

    denpyouID = Command1.Caption

    '今日の日付を自動表示する
    Text1.Text = Date

    '発送方法をコンボボックスに入れる
    mySQL = "SELECT 発送方法 ID, 発送方法 FROM 発送
    方法テーブル"
    Set rs3 = cn.Execute(mySQL, , adCmdText)
    rs3.MoveFirst
    While Not (rs3.EOF)
        Combo2.AddItem rs3!発送方法
        rs3.MoveNext
    Wend

    '発送地域をコンボボックスに入れる
    mySQL = "SELECT * FROM 発送テーブル"
    Set rs4 = cn.Execute(mySQL, , adCmdText)
    rs4.MoveFirst
    While Not (rs4.EOF)
        Combo3.AddItem rs4!地域名
        rs4.MoveNext
    Wend

    '商品の単価
    If shouhinID <> "" Then
        mySQL = "SELECT * FROM 商品マスター WHERE
        商品 ID = " & shouhinID & ""
        Set rs5 = cn.Execute(mySQL, , adCmdText)
        Combo4.Text = shouhinID
        Text7.Text = rs5!単価
    End If
End Sub

Private Sub Combo4_Click() '商品を選択したとき
    mySQL = "SELECT * FROM 商品マスター WHERE 商品
    ID = " & Left(Combo4, 5) & ""
    Set rs5 = cn.Execute(mySQL, , adCmdText)

    If Combo4.Text <> "" Then

        '仕入単価をコマンドボタンに表示する
        Text7.Text = rs5!仕入単価

        '入荷予定日をテキストボックスに表示する
        Text2.Text = rs5!調達期間 + Date

        '仕入先企業を表示する
        mySQL2 = "SELECT * FROM 仕入先テーブル
        WHERE 企業 ID = " & rs5!企業 ID & ""
        Set rs5 = cn.Execute(mySQL2, , adCmdText)
        Text3.Text = rs5!企業 ID & " " & rs5!企業名
```

```

Command3.Caption = rs5!担当者

'商品区分を表示する
Text6.Text = Left(Combo4.Text, 2)

rs5.MoveNext
End If
End Sub

Private Sub Command2_Click() '発注数の表示
mySQL = "SELECT * FROM OR 在庫テーブル WHERE
商品 ID = " & Left(Combo4.Text, 5) & ""
Set rs3 = cn.Execute(mySQL, , adCmdText)

While Not (rs3.EOF)
If rs3!商品 ID = Left(Combo4.Text, 5) Then
Text4.Text = rs3!発注数
Text5.Text = rs3!発注点
rs3.MoveNext
Exit Sub
Else
rs3.MoveNext
End If
Wend

MsgBox "この商品は新商品です。" & vbCrLf & "発注数は入
力してください。"
End Sub

Private Sub Command8_Click() 'E メールシステムを開く
denpyou.Show
Unload Me
End Sub

Private Sub command9_Click() '小計、消費税、合計の計算
Dim shoukei, zei As Integer

If Text4.Text = "" Then
MsgBox "注文数を入力してください。"
Else
'小計
amount = Text4.Text
Command6.Caption = amount * Text7.Text
shoukei = Command6.Caption
'外税
Command5.Caption = Command6.Caption * 0.05
zei = Command5.Caption

zei = Val(zei)
shoukei = Val(shoukei)
'合計
goukei = zei + shoukei
Command7.Caption = goukei
End If
End Sub

Private Sub Command10_Click() 'メニュー画面へ
menu.Show
Unload Me
End Sub

Private Sub Command11_Click() '発注ボタンをクリックした時
Dim name As String

'未入力の項目があった時
If Combo2.Text = "" Or Combo3.Text = "" Or Combo4.Text
= "" Or Text4.Text = "" Then
MsgBox "未入力項目があります。すべて記入してくだ
さい。"
Else
If Text4.Text = "" Then
MsgBox "発注数を入力してください。"
Else '発注数が入力されていた時
mySQL = "SELECT 商品 ID FROM OR 在庫テー
ブル"

Set rs = cn.Execute(mySQL, , adCmdText)

'発送方法 ID の取り出し
mySQL = "SELECT 発送方法 ID FROM 発送方法
テーブル WHERE 発送方法 =" &
Combo2.Text & ""
Set rs3 = cn.Execute(mySQL, , adCmdText)

'運送 ID の取り出し
mySQL2 = "SELECT 輸送 ID FROM 発送テーブ
ル WHERE 地域名 =" &
Combo3.Text & ""
Set rs4 = cn.Execute(mySQL2, , adCmdText)

'データベースへの登録
mySQL = "INSERT INTO 発注伝票テーブル(発注
伝票 ID,発注日,仕入先企業 ID,発送方法
ID, 運送 ID, 小計, 消費税, 合計)
VALUES(" & Command1.Caption &
"," & Text1.Text & "," &
Left(Text3.Text, 2) & "," & rs3!発送方
法 ID & "," & rs4!輸送 ID & "," &
Command6.Caption & "," &
Command5.Caption & "," &
Command7.Caption & ")"
Set rs = cn.Execute(mySQL, , adCmdText)

mySQL = "INSERT INTO 発注明細テーブル(商品
区分 ID,商品 ID,数量,単価,入荷予定日,
発注伝票 ID) VALUES(" & Text6.Text
& "," & Left(Combo4.Text, 5) & "," &
Text4.Text & "," & Text7.Text & "," &
Text2.Text & "," &
Command1.Caption & ")"
Set rs = cn.Execute(mySQL, , adCmdText)

If vbYes = (MsgBox("発注数は" & Text4.Text & "で
す。" & vbCrLf & "よろしいですか?", vbYesNo))
Then
'データベースへの登録
mySQL = "UPDATE OR 在庫テーブル SET
予約発注数=" & Text4.Text & "
WHERE 商品 ID=" &
Left(Combo4.Text, 5) & ""
Set rs = cn.Execute(mySQL, , adCmdText)

MsgBox "発注しました。"
MsgBox "発注が必要な商品が残っています。"
& vbCrLf & "商品を選択し直してく
ださい。"
Combo4.RemoveItem (0)
Command1.Caption = Command1.Caption +
1

If Combo4.ListCount <> 0 Then
Else
MsgBox "発注する商品はありません"
Exit Sub
End If
Else

```

```

MsgBox "発注数を入力してください。"
amount2 = Text4.Text

'OR 在庫テーブルの更新
mySQL = "UPDATE OR 在庫テーブル SET
    発注数=" & Text4.Text & "
    WHERE 商品 ID=" &
    Left(Combo4.Text, 5) & ""
Set rs = cn.Execute(mySQL, adCmdText)
End If
End If
End If

'発注伝票プレビュー表示
DataReport1.Show
End Sub

Private Sub Command12_Click() '在庫件数を調べる
Dim cnt As Integer '在庫件数カウント用

'OR 在庫テーブルからデータの取り出し
mySQL = "SELECT * FROM OR 在庫テーブル"
Set rs = cn.Execute(mySQL, adCmdText)

'商品マスターからデータの取り出し
mySQL2 = "SELECT * FROM 商品マスター WHERE 商
    品 ID =" & rs!商品 ID & ""
Set rs2 = cn.Execute(mySQL2, adCmdText)

rs.MoveFirst
rs2.MoveFirst

'在庫件数を調べる
cnt = 0
While Not (rs.EOF)
    If Val(rs!実在庫数) - rs!予約数 + rs!予約発注数 <
        Val(rs!発注点) Then
        cnt = cnt + 1
        Combo4.AddItem rs!商品 ID
        rs.MoveNext
    Else
        rs.MoveNext
    End If
Wend

MsgBox "在庫数の計算を終了しました。" & vbCrLf & cnt &
    "件の商品を発注してください。"

End Sub

```

【在庫管理モデル】

```

Private cn As ADODB.Connection
Private rs, rs2, rs3, rs4, rs5, rs6, drs As ADODB.Recordset
Private mySQL, mySQL2, mySQL3, mySQL4, mySQL5,
mySQL6, dateSQL As String
Private hatto As Long
Private dash As Long
Private num As Integer
Public lead As Integer

Private Sub Form_Load()
'データベースに接続
Set cn = New ADODB.Connection
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & "Data
Source=C:\inetpub\wwwroot\konta.mdb"

'商品名をコンボボックスに入れる

```

```

mySQL = "SELECT * FROM OR 在庫テーブル ORDER
    BY 商品 ID ASC"
Set rs = cn.Execute(mySQL, adCmdText)
rs.MoveFirst
While Not (rs.EOF)
    Combo2.AddItem rs!商品 ID
    rs.MoveNext
Wend
End Sub

Private Sub Combo2_Click()
'選択された商品の情報を各テキストボックスに入れる
mySQL = "SELECT * FROM OR 在庫テーブル WHERE
    商品 ID =" & Combo2.Text & ""
Set rs = cn.Execute(mySQL, adCmdText)

k.Text = rs!発注コスト
y.Text = rs!年間期待需要量
p.Text = rs!品切れコスト
h.Text = rs!在庫維持コスト
Label1.Caption = rs!調達期間の需要量

mySQL2 = "SELECT * FROM 商品マスター WHERE 商
    品 ID =" & Combo2.Text & ""
Set rs2 = cn.Execute(mySQL2, adCmdText)

Label2.Caption = rs2!商品名
End Sub

Private Sub Command1_Click() '計算
Dim heikin As Double
Dim hensa As Double
Dim kukan As Double
Dim sum As Double

kukan = 1.96
heikin = Label1.Caption / lead

sum = 0

rs6.MoveFirst

While Not (rs6.EOF)
    sum = sum + ((rs6!数量 * heikin) ^ 2)
    rs6.MoveNext
Wend

hensa = Sqr(sum / Label1.Caption)

If k.Text <> "" And y.Text <> "" And p.Text <> "" And
    h.Text <> "" And Label1.Caption <> "" Then
'問題が実行可能解をもつかどうかチェックする
hatto = Sqr(2 * y.Text * (Label1.Caption / 2) / 2)
dash = p.Text * y.Text / 2
If dash >= hatto Then
    Y1 = Sqr(2 * y.Text * k.Text / h.Text)
    r1 = lead * heikin + Sqr(lead) * kukan * hensa
    s = r1 ^ 2 / (2 * Label1.Caption) - r1 +
        (Label1.Caption / 2)
    amount.Text = Int(Y1)
    pnt.Text = Int(r1)

mySQL = "UPDATE OR 在庫テーブル set 発注コ
    スト =" & k.Text & ",年間期待需要量="
    & y.Text & ",品切れコスト=" & p.Text &
    ",在庫維持コスト=" & h.Text & ",調達期
    間の需要量=" & Label1.Caption & ",発

```

```

        注数=" & amount.Text & ",発注点=" &
        pnt.Text & " WHERE 商品 ID=" &
        Combo2.Text & ""
        cn.Execute mySQL, , adCmdText
    Else
        MsgBox "解は存在しません。データを入れなおし
        てください。"
    End If
Else
    MsgBox "商品 ID を選んでください。"
End If
End Sub

Private Sub Command2_Click() 'メニュー画面を開く
    menu.Show
    Unload Me
End Sub

Private Sub Command3_Click() '調達期間の需要量の計算
    Dim hacchu, nyuka, shukka As Date
    Dim i As Integer
    Dim dtv As Variant

    mySQL = "SELECT * FROM 発注伝票テーブル"
    Set rs = cn.Execute(mySQL, , adCmdText)

    mySQL4 = "SELECT * FROM 発注明細テーブル WHERE
    発注伝票 ID = " & rs!発注伝票 ID
    Set rs4 = cn.Execute(mySQL4, , adCmdText)

    mySQL2 = "SELECT * FROM 出荷テーブル WHERE 商
    品 ID =" & Combo2.Text & ""
    Set rs2 = cn.Execute(mySQL2, , adCmdText)

    mySQL3 = "SELECT * FROM 入荷テーブル WHERE 商
    品 ID =" & Combo2.Text & ""
    Set rs3 = cn.Execute(mySQL3, , adCmdText)

    mySQL5 = "SELECT * FROM 受注伝票テーブル"
    Set rs5 = cn.Execute(mySQL5, , adCmdText)

    If rs.EOF Or rs2.EOF Or rs3.EOF Or rs4.EOF Or
    rs5.EOF Then
        MsgBox "この商品は調達期間の間に需要はありません
        でした。"
    Exit Sub
    Else
        rs.MoveFirst
        rs2.MoveFirst
        rs3.MoveFirst
        rs4.MoveFirst
        rs5.MoveFirst
    End If

    '調達期間の需要量を求める
    shukka = rs2!出荷日
    hacchu = rs!発注日
    nyuka = rs3!入荷日
    '調達期間の計算
    lead = nyuka - shukka

    '受注伝票の日付けが発注日から入荷日までのデータを取り
    出す
    dateSQL = "SELECT * FROM 受注伝票テーブル
    WHERE 受注日 BETWEEN #" & rs!発注日 &
    "# And #" & rs3!入荷日 & "#"
    Set drs = cn.Execute(dateSQL, , adCmdText)

```

```

mySQL6 = "SELECT * FROM 受注明細テーブル WHERE
    受注伝票 ID = " & drs!受注伝票 ID
Set rs6 = cn.Execute(mySQL6, , adCmdText)

drs.MoveFirst
rs6.MoveFirst

num = 0
While Not (rs6.EOF)
    num = num + rs6!数量
    rs6.MoveNext
    drs.MoveNext
Wend

Label1.Caption = num
End Sub

Private Sub Command4_Click() 'クリア
    Combo2.Text = ""
    k.Text = ""
    y.Text = ""
    p.Text = ""
    h.Text = ""
    Label1.Caption = ""
    amount.Text = ""
    pnt.Text = ""
End Sub

Private Sub Command5_Click() '発注画面を開く
    shouhinID = Combo2.Text
    hacchu.Show
    Unload Me
End Sub

【メール送信画面】
Option Explicit

Public Base0 As String
Public Base1 As String
Public Base2 As String
Public Base3 As String
Public base As String
Public MessageBody As String
Public MessageHeader As String
Dim flg As Integer

Private cn As ADODB.Connection
Private rs, rs2 As ADODB.Recordset
Private mySQL, mySQL2 As String

'送信
Private Sub Kansuu()
    Const Boundary = "**** boundary ****"

    flg = 0
    MessageBody = ""

    CommonDialog1.FileName = Text5.Text

    'メールヘッダー
    MessageHeader = ""
    MessageHeader = MessageHeader & "Reply-To: " &
    Text1.Text & vbCrLf
    MessageHeader = MessageHeader & "X-Mailer:
    vb_smtptest" & vbCrLf
    MessageHeader = MessageHeader & "MIME-Version: 1.0"
    & vbCrLf

```

```

MessageHeader = MessageHeader & "X-Mozilla-Status:
                0001" & vbCrLf

Winsock1.Connect

While flg = 0
    DoEvents
Wend

'SMTP へ接続の開始
If flg = 1 Then '220
    Winsock1.SendData "HELO leo.tg.ube-k.ac.jp" &
vbCrLf
    flg = 0
Else
    MsgBox "接続 error エラー"
End If

While flg = 0
    DoEvents
Wend

'mail 処理の開始
If flg = 1 Then '250
    Winsock1.SendData "MAIL FROM:<" & Text1.Text &
">" & vbCrLf
    flg = 0
Else
    MsgBox "メールエラー"
End If

While flg = 0
    DoEvents
Wend

'受信者の設定
If flg = 2 Then '250
    Winsock1.SendData "RCPT TO:<" & Text2.Text & ">"
& vbCrLf
    flg = 0
Else
    MsgBox "受信者設定エラー"
End If

    While flg = 0
        DoEvents
    Wend

'テキストデータ送信準備
If flg = 2 Then '250
    Winsock1.SendData "DATA" & vbCrLf
    flg = 0
Else
    MsgBox "送信準備エラー"
End If

While flg = 0
    DoEvents
Wend

'題名と本文の送信
CommonDialog1.FileName = Text5.Text

If flg = 3 Then '354
    If Text5.Text = "" Then
        '添付ファイルなし
        MessageBody = MessageBody & "Content-Type:
text/plain;"

```

```

                charset=Shift-Jis" &
                vbCrLf
        MessageBody = MessageBody &
"Content-Transfer-Encoding:
7bit" & vbCrLf
        MessageBody = "Subject:" & JisConv(Text3.Text)
& vbCrLf & vbCrLf &
Text4.Text & vbCrLf & "." &
vbCrLf
        Winsock1.SendData MessageHeader &
MessageBody

        flg = 0
    Else
        '添付ファイルあり
        MessageBody = "Subject: " & JisConv(Text3.Text)
& vbCrLf
        MessageBody = MessageBody & "Content-Type:
multipart/mixed;
boundary="" & Boundary &
"" & vbCrLf & vbCrLf

        MessageBody = MessageBody & "This is a
multi-part message in MIME
format." & vbCrLf & vbCrLf

                MessageBody =
                MessageBody & "--" & Boundary &
vbCrLf
        MessageBody = MessageBody & "Content-Type:
text/plain;
charset=Shift-JIS" &
vbCrLf
        MessageBody = MessageBody &
"Content-Transfer-Encoding:
7bit" & vbCrLf & vbCrLf

        MessageBody = MessageBody & Text4.Text &
vbCrLf & vbCrLf

        MessageBody = MessageBody & "--" & Boundary
& vbCrLf
        MessageBody = MessageBody & "Content-Type: "
&
        GetMime(CommonDialog1.File
Title) & "; name="" &
CommonDialog1.FileTitle & """"
& vbCrLf
        MessageBody = MessageBody &
"Content-Transfer-Encoding:
base64" & vbCrLf
        MessageBody = MessageBody &
"Content-Disposition: inline;
filename="" &
JisConv(CommonDialog1.FileTi
tle) & """" & vbCrLf & vbCrLf

        MessageBody = MessageBody &
Encod_Base64(Text5.Text)

        MessageBody = MessageBody & "--" & Boundary
& "--" & vbCrLf & "." & vbCrLf

        Winsock1.SendData MessageHeader &
MessageBody

        flg = 0
    End If

```

```

Else
  MsgBox "送信 error..."
End If

While flg = 0
  DoEvents
Wend

If flg = 2 Then '250
  Winsock1.SendData "QUIT" & vbCrLf
  flg = 0
Else
  MsgBox "終了 error..."
End If

While flg = 0
  DoEvents
Wend

'送信完了
If flg = 4 Then '221
  MsgBox "送信完了しました", vbOKOnly
  flg = 0
Else
  MsgBox "送信できませんでした"
End If
End Sub

'MIME タイプを指定
Public Function GetMime(FileName As String) As String
  Select Case LCase(Mid(FileName, InStr(1, FileName, ".")
+ 1, 3))
    Case "jpg": GetMime = "image/jpeg"
    Case "jpeg": GetMime = "image/jpeg"
    Case "gif": GetMime = "image/gif"
    Case "xls": GetMime = "application/vnd.ms-excel"
    Case "doc": GetMime = "application/msword"
    Case "txt": GetMime = "text/plain"
    Case Else: GetMime = "application/octet-stream"
  End Select
End Function

'シフト JIS 文を JIS 文に変換
Function JisConv(post_data As String) As String
  Dim pos As Long

  Dim aaa As String
  Dim xxx As String
  Dim yyy As String

  Dim wf1 As Integer '漢字スタート 1 : シフトイン 0 :
シフトアウト
  Dim wf2 As Integer 'カタカナスタート 1 : シフトイン
0 : シフトアウト

  wf1 = 0
  wf2 = 0
  pos = 1
  aaa = ""

  Do
    If pos > Len(post_data) Then
      Exit Do
    End If

    xxx = Hex(Asc(Mid(post_data, pos, 1)))
    If ("&H81" <= "&H" & xxx And "&H" & xxx <=
"&H9F") Or ("&HE0" <= "&H" & xxx And "&H" &

```

```

xxx <= "&HFC") Then
  '漢字
  If wf2 = 1 Then
    wf2 = 0
  End If
  If wf1 = 0 Then
    wf1 = 1
    aaa = aaa & Chr(&H1B) & Chr(&H24) &
Chr(&H42)
  End If

  yyy = SJISTOJIS(Hex(Asc(Mid(post_data, pos,
1))))
  aaa = aaa & Chr("&H" & Mid(yyy, 1, 2)) &
Chr("&H" & Mid(yyy, 3, 2))
  pos = pos + 1
Elseif &HA0 <= Val("&H" & xxx) And Val("&H" &
xxx) <= &HDF Then
  'カタカナ
  If wf1 = 1 Then
    wf1 = 0
  End If
  If wf2 = 0 Then
    wf2 = 1
    aaa = aaa & Chr(&H1B) & Chr(&H28) &
Chr(&H49)
  End If

  aaa = aaa & Chr(Asc(Mid(post_data, pos, 1)) -
128)
  pos = pos + 1
Else
  '半角英数字
  If wf1 = 1 Or wf2 = 1 Then
    wf1 = 0
    wf2 = 0
    aaa = aaa & Chr(&H1B) & Chr(&H28) &
Chr(&H4A)
  End If
  aaa = aaa & Chr(Asc(Mid(post_data, pos, 1)))
  pos = pos + 1
End If
Loop

  JisConv = aaa
End Function

'S-JIS -> JIS 変換
Public Function SJISTOJIS(SJIS As String) As String
  Dim H1 As Double
  Dim LO As Double

  H1 = Val("&H" & Mid(SJIS, 1, 2))
  LO = Val("&H" & Mid(SJIS, 3, 2))

  If H1 <= &H9F Then
    H1 = H1 - &H71
  Else
    H1 = H1 - &HB1
  End If

  H1 = H1 * 2 + 1

  If LO > &H7F Then
    LO = LO - 1
  End If

  If LO >= &H9E Then
    LO = LO - &H7D

```

```

    HI = HI + 1
Else
    LO = LO - &H1F
End If

SJISTOJIS = Hex(Val("&H" & Right("00" & Hex(HI), 2) &
    Right("00" & Hex(LO), 2)))
End Function

```

'Base64 データをエンコード

```
Public Function Encod_Base64(Attch_Name As String) As String
```

```

    Dim FileNo As Long
    Dim buf() As Byte
    Dim i As Integer
    Dim pos As Integer
    Dim aaa As String
    Dim bbb As String
    Dim PosM1 As Long
    Dim PosM2 As Integer

```

'添付ファイル読み込み

```

FileNo = FreeFile
Open Attch_Name For Binary As #FileNo

```

```
ReDim Preserve buf(LOF(FileNo)) As Byte
```

```
PosM1 = LOF(FileNo)
```

```

For i = 1 To LOF(FileNo)
    Get #FileNo, , buf(i)
Next i
Close #1

```

```

aaa = ""
pos = 1

```

```
PosM2 = PosM1 - (PosM1 Mod 3)
```

```

For pos = 1 To PosM2 Step 3
    aaa = aaa & E_Base64(buf(pos) And &HFC) / 4
    aaa = aaa & E_Base64((buf(pos) And &H3) * 16 +
        (buf(pos + 1) And &HF0) / 16)
    aaa = aaa & E_Base64((buf(pos + 1) And &HF) * 4 +
        (buf(pos + 2) And &HC0) / 64)
    aaa = aaa & E_Base64(buf(pos + 2) And &H3F)
Next

```

```

If (PosM1 Mod 3) = 2 Then
    aaa = aaa & E_Base64(buf(pos) And &HFC) / 4
    aaa = aaa & E_Base64((buf(pos) And &H3) * 16 +
        (buf(pos + 1) And &HF0) / 16)
    aaa = aaa & E_Base64((buf(pos + 1) And &HF) * 4)
ElseIf (PosM1 Mod 3) = 1 Then
    aaa = aaa & E_Base64(buf(pos) And &HFC) / 4
    aaa = aaa & E_Base64((buf(pos) And &H3) * 16)
End If

```

```

If (Len(aaa) Mod 3) = 2 Then
    aaa = aaa & "="
ElseIf (Len(aaa) Mod 3) = 1 Then
    aaa = aaa & "=="
End If

```

```

bbb = ""
pos = 1

```

```

Do
    If pos > Len(aaa) Then

```

```

        Exit Do
    End If

```

```

        bbb = bbb & Mid(aaa, pos, 72) & vbCrLf
        pos = pos + 72

```

```
Loop
```

```
Encod_Base64 = bbb
```

```
End Function
```

'Base64 データをエンコード

```
Public Function E_Base64(B_data As Integer) As String
```

```

    Select Case B_data
        Case 0: E_Base64 = "A"
        Case 1: E_Base64 = "B"
        Case 2: E_Base64 = "C"
        Case 3: E_Base64 = "D"
        Case 4: E_Base64 = "E"
        Case 5: E_Base64 = "F"
        Case 6: E_Base64 = "G"
        Case 7: E_Base64 = "H"
        Case 8: E_Base64 = "I"
        Case 9: E_Base64 = "J"
        Case 10: E_Base64 = "K"
        Case 11: E_Base64 = "L"
        Case 12: E_Base64 = "M"
        Case 13: E_Base64 = "N"
        Case 14: E_Base64 = "O"
        Case 15: E_Base64 = "P"
        Case 16: E_Base64 = "Q"
        Case 17: E_Base64 = "R"
        Case 18: E_Base64 = "S"
        Case 19: E_Base64 = "T"
        Case 20: E_Base64 = "U"
        Case 21: E_Base64 = "V"
        Case 22: E_Base64 = "W"
        Case 23: E_Base64 = "X"
        Case 24: E_Base64 = "Y"
        Case 25: E_Base64 = "Z"
        Case 26: E_Base64 = "a"
        Case 27: E_Base64 = "b"
        Case 28: E_Base64 = "c"
        Case 29: E_Base64 = "d"
        Case 30: E_Base64 = "e"
        Case 31: E_Base64 = "f"
        Case 32: E_Base64 = "g"
        Case 33: E_Base64 = "h"
        Case 34: E_Base64 = "i"
        Case 35: E_Base64 = "j"
        Case 36: E_Base64 = "k"
        Case 37: E_Base64 = "l"
        Case 38: E_Base64 = "m"
        Case 39: E_Base64 = "n"
        Case 40: E_Base64 = "o"
        Case 41: E_Base64 = "p"
        Case 42: E_Base64 = "q"
        Case 43: E_Base64 = "r"
        Case 44: E_Base64 = "s"
        Case 45: E_Base64 = "t"
        Case 46: E_Base64 = "u"
        Case 47: E_Base64 = "v"
        Case 48: E_Base64 = "w"
        Case 49: E_Base64 = "x"
        Case 50: E_Base64 = "y"
        Case 51: E_Base64 = "z"
        Case 52: E_Base64 = "0"
        Case 53: E_Base64 = "1"
        Case 54: E_Base64 = "2"

```

```

Case 55: E_B64 = "3"
Case 56: E_B64 = "4"
Case 57: E_B64 = "5"
Case 58: E_B64 = "6"
Case 59: E_B64 = "7"
Case 60: E_B64 = "8"
Case 61: E_B64 = "9"
Case 62: E_B64 = "+"
Case 63: E_B64 = "/"
End Select
End Function

Private Sub Command2_Click()
If Winsock1.State = 0 Then
If Text3.Text = "" Then
MsgBox "タイトルが入ってないけど…いいの??",
vbOKCancel
If vbOKCancel = "2" Then
Exit Sub
Else
Kansuu
End If
Else
Kansuu
End If
Else
If Winsock1.State = 1 Then
Winsock1.Close
End If
If Winsock1.State = 9 Then
Winsock1.Close
End If
End If
End Sub

Private Sub Command3_Click() 'メニュー画面を開く
menu.Show
Unload Me
End Sub

Private Sub Command4_Click()
'コモンダイアログ参照
CommonDialog1.ShowOpen
Text5.Text = CommonDialog1.FileName
End Sub

Private Sub Form_Load()
'データベースに接続
Set cn = New ADODB.Connection
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & "Data
Source=C:\¥Inetpub¥konta.mdb;"

mySQL = "SELECT * FROM 発注伝票テーブル WHERE
発注伝票 ID=" & denpyouID
'mySQL = "SELECT * FROM 発注伝票テーブル WHERE
発注伝票 ID=4"
Set rs = cn.Execute(mySQL, , adCmdText)

mySQL2 = "SELECT * FROM 仕入先テーブル WHERE
企業 ID=" & rs!仕入先企業 ID & ""
Set rs2 = cn.Execute(mySQL2, , adCmdText)

If rs2!企業 URL = "-" Then
MsgBox "この企業のメールアドレスは登録されていま
せん。" & vbCrLf & "発注伝票の印刷を行って
ください。"

DataReport1.Show
Else
Text1.Text = rs2!企業メールアドレス
Text2.Text = "kousen@st.ube-k.ac.jp"
Text3.Text = "発注伝票送付"
Text4.Text = "添付ファイルにて発注伝票を送りたいし
ます。"
End If
End Sub

Private Sub Winsock1_Close()
Winsock1.Close
End Sub

Private Sub Winsock1_Connect()
flg = 1
End Sub

'サーバからの返事に対して発生
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim st As String
Dim code As String

st = ""

Winsock1.GetData st, vbString

code = Val(Left(st, 3))

Select Case code
Case 220
flg = 1
Case 250
flg = 2
Case 354
flg = 3
Case 221
flg = 4
Case Else
MsgBox "error..."
End Select
End Sub

'エラー処理
Private Sub Winsock1_Error(ByVal Number As Integer,
Description As String, ByVal Scode As Long, ByVal Source As
String, ByVal HelpFile As String, ByVal HelpContext As Long,
CancelDisplay As Boolean)
If Err.Number <> 0 Then
Msg = "エラー番号 " & Str(Err.Number) & Err.Source
& " でエラーが発生しました。" & Chr(13) &
Err.Description
MsgBox Msg, , " エ ラ ー ", Err.HelpFile,
Err.HelpContext
End If
End Sub

【月間総売上金額/個数グラフ表示】
Private cn As ADODB.Connection
Private rs As ADODB.Recordset
Private mySQL As String
Private month As Variant

Private Sub Command1_Click()
menu.Show
Unload Me
End Sub

Private Sub Command2_Click()
graph.Show

```

```

Unload Me
End Sub

Private Sub Command3_Click()
    Dim amount(11) As Long

    Label18.Caption = "単位 : 1000"

    Picture1.Cls

    mySQL2 = "SELECT * FROM 売上累計テーブル"
    Set rs2 = cn.Execute(mySQL2, , adCmdText)

    While Not (rs2.EOF)
        '売上の月を取得
        month = Right(rs2!売上日, 5)
        month = Int(Left(month, 2))

        Select Case month
        Case 1
            amount(0) = amount(0) + rs2!売上数
        Case 2
            amount(1) = amount(1) + rs2!売上数
        Case 3
            amount(2) = amount(2) + rs2!売上数
        Case 4
            amount(3) = amount(3) + rs2!売上数
        Case 5
            amount(4) = amount(4) + rs2!売上数
        Case 6
            amount(5) = amount(5) + rs2!売上数
        Case 7
            amount(6) = amount(6) + rs2!売上数
        Case 8
            amount(7) = amount(7) + rs2!売上数
        Case 9
            amount(8) = amount(8) + rs2!売上数
        Case 10
            amount(9) = amount(9) + rs2!売上数
        Case 11
            amount(10) = amount(10) + rs2!売上数
        Case 12
            amount(11) = amount(11) + rs2!売上数
        Case Else
        End Select
        rs2.MoveNext
    Wend

    Picture1.Line (1300, amount(0))-(1950, 0), RGB(0, 0, 0),
BF
    Picture1.Line (1950, amount(1))-(2600, 0), RGB(153, 153,
153), BF
    Picture1.Line (2600, amount(2))-(3250, 0), RGB(255, 153,
0), BF
    Picture1.Line (3250, amount(3))-(3900, 0), RGB(204, 255,
0), BF
    Picture1.Line (3900, amount(4))-(4550, 0), RGB(0, 255, 0),
BF
    Picture1.Line (4550, amount(5))-(5200, 0), RGB(0, 153,
255), BF
    Picture1.Line (5200, amount(6))-(5850, 0), RGB(102, 255,
255), BF
    Picture1.Line (5850, amount(7))-(6500, 0), RGB(255, 0,
204), BF
    Picture1.Line (6500, amount(8))-(7150, 0), RGB(255, 0, 0),
BF
    Picture1.Line (7150, amount(9))-(7800, 0), RGB(255, 102,
0), BF

```

```

    Picture1.Line (7800, amount(10))-(9000, 0), RGB(255, 255,
0), BF
    Picture1.Line (9000, amount(11))-(9650, 0), RGB(255, 255,
255), BF

    '売上数の表示
    For i = 0 To 11 Step 1
        If amount(i) <> 0 Then
            Label19(i).Caption = amount(i) / 1000
        Else
            Label19(i).Caption = ""
        End If
    Next i
End Sub

Private Sub Command4_Click()
    Picture1.Cls

    Dim money(11) As Long

    Label18.Caption = "単位 : 1000"

    mySQL2 = "SELECT * FROM 売上累計テーブル"
    Set rs2 = cn.Execute(mySQL2, , adCmdText)

    While Not (rs2.EOF)
        '売上の月を取得
        month = Right(rs2!売上日, 5)
        month = Int(Left(month, 2))

        Select Case month
        Case 1
            money(0) = money(0) + rs2!売上金額
        Case 2
            money(1) = money(1) + rs2!売上金額
        Case 3
            money(2) = money(2) + rs2!売上金額
        Case 4
            money(3) = money(3) + rs2!売上金額
        Case 5
            money(4) = money(4) + rs2!売上金額
        Case 6
            money(5) = money(5) + rs2!売上金額
        Case 7
            money(6) = money(6) + rs2!売上金額
        Case 8
            money(7) = money(7) + rs2!売上金額
        Case 9
            money(8) = money(8) + rs2!売上金額
        Case 10
            money(9) = money(9) + rs2!売上金額
        Case 11
            money(10) = money(10) + rs2!売上金額
        Case 12
            money(11) = money(11) + rs2!売上金額
        Case Else
        End Select
        rs2.MoveNext
    Wend

    Picture1.Line (1300, money(0) / 1000)-(1950, 0), RGB(0, 0,
0), BF
    Picture1.Line (1950, money(1) / 1000)-(2600, 0), RGB(153,
153, 153), BF
    Picture1.Line (2600, money(2) / 1000)-(3250, 0), RGB(255,
153, 0), BF
    Picture1.Line (3250, money(3) / 1000)-(3900, 0), RGB(204,
255, 0), BF
    Picture1.Line (3900, money(4) / 1000)-(4550, 0), RGB(0,
255, 0), BF

```

```
255, 0), BF
Picture1.Line (4550, money(5) / 1000)-(5200, 0), RGB(0,
153, 255), BF
Picture1.Line (5200, money(6) / 1000)-(5850, 0), RGB(102,
255, 255), BF
Picture1.Line (5850, money(7) / 1000)-(6500, 0), RGB(255,
0, 204), BF
Picture1.Line (6500, money(8) / 1000)-(7150, 0), RGB(255,
0, 0), BF
Picture1.Line (7150, money(9) / 1000)-(7800, 0), RGB(255,
102, 0), BF
Picture1.Line (7800, money(10) / 1000)-(9000, 0),
RGB(255, 255, 0), BF
Picture1.Line (9000, money(11) / 1000)-(9650, 0), RGB(255,
255, 255), BF

'売上数の表示
For i = 0 To 11 Step 1
    If money(i) <> 0 Then

Label19(i).Caption = money(i) / 1000
Else
Label19(i).Caption = ""
End If
Next i
End Sub

Private Sub Form_Load()
    'データベースに接続
    Set cn = New ADODB.Connection
    cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & "Data
Source=C:\Inetpub\konta.mdb;"

    Picture1.Scale (0, 7000)-(10400, 0)
End Sub
```