

Kansai Perl Mongers

Perl Mongers is a not-for-profit organization whose mission is to establish Perl user groups. In the legal sense, Perl Mongers is simply a legal entity completely separate from any Perl user group. Perl Mongers is organized under the laws of the state of New York.



Perlによる Web+Database活用事例

関西Perl Mongers
Akinobu Yamamoto

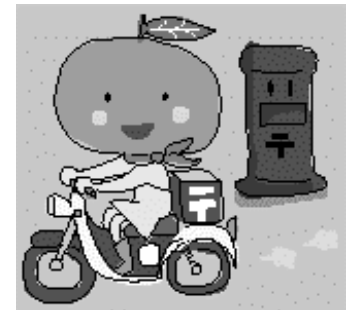



講演を行う前に...

この資料は1999年11月13日に秋葉原の
イナハライベントホールで行われた
「オープンソースまつり」で
株式会社クレイフィッシュの小山さんが
“Perlではじめるデータベースプログラミング”
として講演した資料を
私山本が加筆および修正したものです。


講演の内容

- ★ Perlによるデータベースプログラミング
- ★ DBI アプリケーションの実例
(なっちゃんの郵便屋さんの紹介)





Perlで始める データベースプログラミング



なぜデータベースが必要か？

- ★ ランダムアクセスと挿入
- ★ トランザクション
- ★ 移植性

これらに関わるコーディング工数の問題と、
イバリが効くというメリット



Perlでデータベースを扱うメリット

- ★ プログラミング言語として習得が容易
- ★ 作成と変更が容易
- ★ 多彩なモジュールが無料で利用できる
- ★ Internet との親和性
- ★ マルチプラットフォーム
- ★ 迅速、良質なサポート



DBI – DataBase Interface

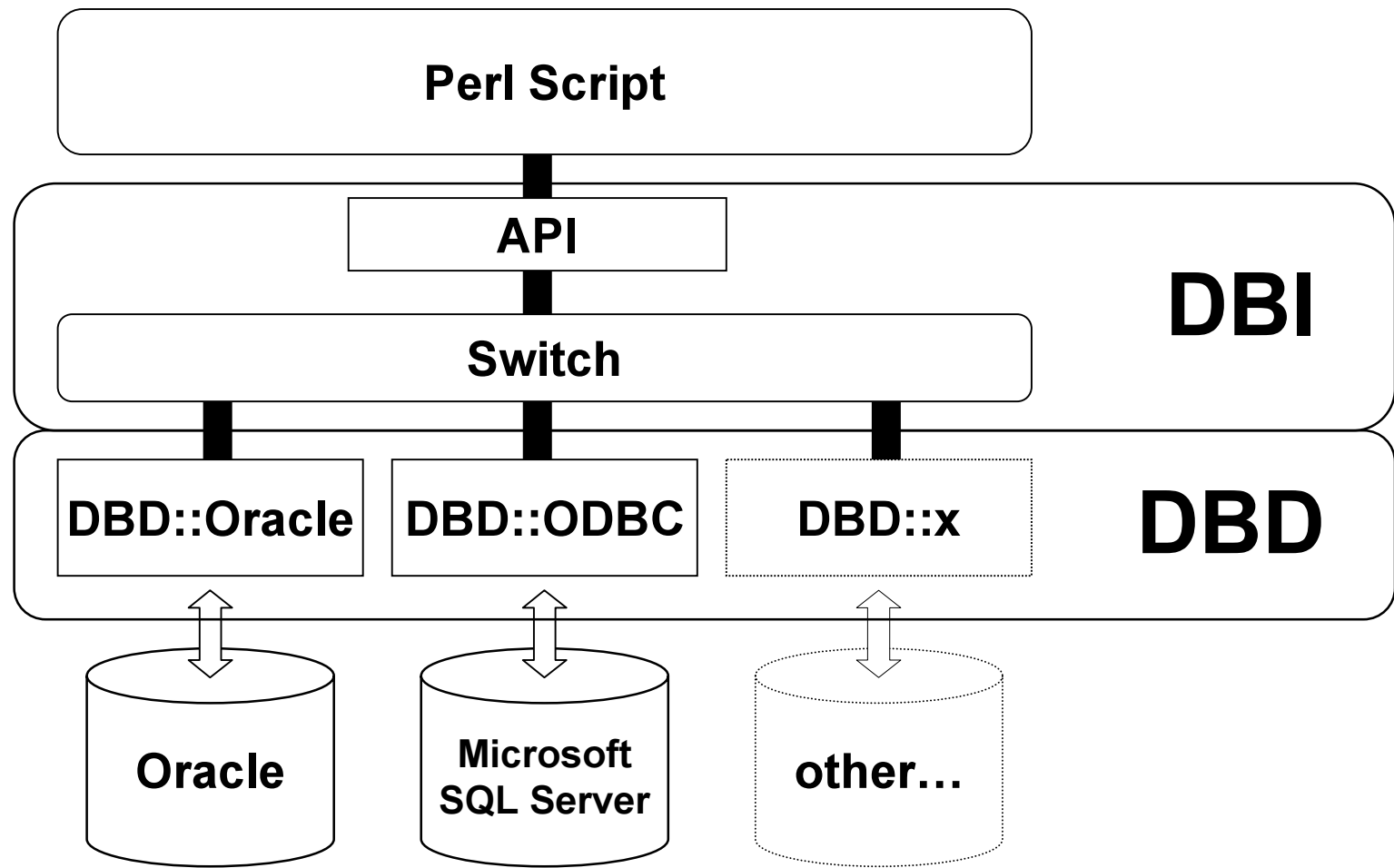
- ★ Perl のためのデータベースAPI
- ★ 異なるデータベースにも一貫したI/Fでアクセス可能



DBD – データベースドライバ

- ✱ DBD::Oracle
 - ✱ DBD::ODBC
 - ✱ DBD::Sybase
 - ✱ DBD::Informix
 - ✱ DBD::Pg(PostgreSQL)
 - ✱ DBD::mSQL
 - ✱ DBD::xBase
- etc...

DBI&DBD のアーキテクチャ





簡単な使用例

- ✳ Microsoft Access sample Northwindデータベースに接続
- ✳ 在庫切れの商品一覧を取り出し表示する



商品テーブル

商品コード
フリガナ
商品名
仕入先コード
梱包単位
単価
在庫

在庫切れの商品一覧がほしい...

sample1.pl

```
use DBI;
my $dbh = DBI->connect('dbi:ODBC:Northwind','', '');
my $sth = $dbh->prepare(q{
    SELECT 商品名, 単価 FROM 商品 where 在庫 = 0
});
$sth->execute;
while (my $cols = $sth->fetchrow_arrayref) {
    printf "%s = %d¥n", $cols->[0], $cols->[1];
}
$sth->finish;
$dbh->disconnect;
exit;
```

>perl sample1.pl

C:\¥>perl sample1.pl

バナナキャンディー = 160

ロッキーチーズ = 380

極上マカロニ = 190

コーンフレークチョコ = 220

アメリカンポーク = 180

うす味ウィンナー = 160

もめんどうふ特上 = 120

きぬごしどうふ特上 = 120

特選味のり = 1800

大陸サーモン = 2800

なまからし = 200



データベース操作の流れ #参照

- データソースに接続 - connect
- 要求を準備 - prepare
- 実行 - execute
- 結果の取得 - fetchrow_arrayref
- 処理の終了 - finish
- データソースから切断 - disconnect



データベース操作の流れ #変更

- データソースに接続 - connect
- 要求を準備 - prepare
- 実行 - execute
- 処理の終了 - finish
- 変更を確定,破棄 - commit, rollback
- データソースから切断 - disconnect

DBI->connect() #接続する

```
DBI->connect($data_souse, $user, $password,  
    ¥%attr);
```

要求されたデータソースへ接続
データベースハンドルを返す
接続に失敗すると undef を返す

ex..

```
$dbh = DBI->connect(  
    $data_souse, $user, $pass  
) or die $DBI::errstr;
```

データソース

- ★ 接続するデータソースを指定

dbi:ドライバ名:データベース名

ODBC *dbi:ODBC:DSN*

Oracle *dbi:Oracle:S/D*

データソースによってはホスト名やポート番号なども指定する



接続オプション

ハッシュのリファレンスで設定

✱ PrintError

エラーが発生したらエラーコードと警告メッセージ warn() を表示する。

=> デフォルトは ON (1)

✱ RaiseError

エラーが発生したら例外 die() を発生する。

=> デフォルトは OFF (0)

✱ AutoCommit

データベースの変更を自動的に保存 (コミット) する。

=> デフォルトは ON (1)

オプションの例

```
$dbh = DBI->connect(  
    'dbi:ODBC:Northwind', $username, $pass, {  
        PrintError => 0,  
        RaiseError => 1,  
        AutoCommit => 0  
    }  
);
```

警告非表示、エラーは例外を発生、自動でコミットしない



`$dbh->prepare()` #要求の準備

```
$dbh->prepare( $sql_statement );
```

SQL文をデータベースが実行できるように準備し、ステートメントハンドルを返す。

実行や結果の取得にはステートメントハンドルを使用する。

ex..

```
$sth = $dbh->prepare(q{  
    SELECT col_name, col_name FROM table_name  
});
```



`$sth->execute()` # 要求の実行

`$sth->execute();`

`prepare()`でセットされたSQLステートメント
を実行する。

失敗すると `undef` が返される。

SELECT 文以外を `execute()` すると

=> 影響を受けた(追加、更新された)行の数を返す

0件の時は `0E0` => perl は 0 として扱うが 真と解釈する

SELECT 文を `execute()` すると

=> データベースエンジンのステータスを返す

行数はまだ判らない



`$sth->finish` **# 処理の終了**

`$sth->finish;`

prepare()でセットされたSQLステートメント
の処理を終了する。

プレースホルダとバインド値

- ★ prepareされたSQLステートメントへ値を挿入できる
- ★ 動的な検索条件 or 登録データの変更

ex..

```
$sth = $dbh->prepare(q{  
    INSERT INTO users (name, mail, age)  
    VALUES (?, ?, ?)  
});  
$sth->execute($name, $mail, $age);
```


パフォーマンス

- ★ プレースホルダを使わない場合、繰り返し prepare する必要がある

ex..

```
$sth = $dbh->prepare(q{  
    INSERT INTO users (name, mail, age) VALUES (?, ?, ?)  
});  
while (<CSV>) {  
    chomp;  
    my ($name, $mail, $age) = split /,/;  
    $sth->execute($name, $mail, $age) or die $dbh->errstr;  
}  
$dbh->finish;  
$dbh->commit;
```



`$sth->fetchrow_arrayref` #結果の取得

```
$array_ref = $sth->fetchrow_arrayref;
```

データベースから返されたデータを一行取り出し、配列のリファレンスとして返す。

データの終端に達すると undef を返す。

ex...

```
while (my $cols = $sth->fetchrow_arrayref) {  
    print $cols->[0], $cols->[1], $cols->[2], "¥n";  
}
```



fetch のバリエーション

- ✳ fetchrow_arrayの別名

```
$array_ref = $sth->fetch;
```

- ✳ リファレンスではなく配列で返す

```
@array = $sth->fetchrow_array;
```

- ✳ カラムの名前をハッシュのキーとして返す

```
$hash_ref = $sth->fetchrow_hashref( $name );  
print $hash_ref->{NAME}, $hash_ref->{MAIL};
```



\$dbh->disconnect; # 切断する

- ✳ データベースハンドルをデータベースから切断する
- ✳ disconnect時のトランザクションの動作はデータベースにより異なる。
 - > commit するか rollback するかはまちまち
- ✳ 明示的に切断しないままハンドルが使用されなくなったり、スクリプトが終了した場合は自動的に rollback してから切断される。



```
$dbh->commit; # 変更の確定  
$dbh->rollback; # 変更の破棄
```

- ✳ データベースに対する一連の変更をコミット(確定)する or ロールバック(破棄)
- ✳ データベースがトランザクションをサポートしている場合に使用可能。
- ✳ AutoCommit を有効にしてきるときに使用すると警告が発生する。



トランザクション

- ★ データベースへの一連の作業を一つの単位として保証することで、エラーやデータの不整合などデータベース障害から保護する。
- ★ AutoCommit は無効OFF(0)にする必要がある。



Perl script でのトランザクション

★ RaiseError を有効ON(1)に設定し、一連の処理を `eval { ... }` で囲む

ex..

```
eval {  
    # データの挿入や 更新  
    # その他いろいろな処理  
};  
if ($@) {  
    $dbh->rollback;  
    # この後に他の後片付けを追加するもよし  
} else {  
    $dbh->commit;  
}
```



Perl,DBI の情報源 # Internet

- ★ DBI オフィシャル Web site

<http://www.symbolstone.org/technology/perl/DBI/>

- ★ DBIドキュメント日本語訳 – 河馬屋2千年堂

<http://member.nifty.ne.jp/hippo2000/>

- ★ DBI メーリングリスト(英語)

<http://www.isc.org/services/public/lists/dbi-lists.html>

- ★ DBI-japan メーリングリスト

http://member.nifty.ne.jp/hippo2000/dbi/dbi_japan.htm

- ★ CPAN

<http://www.perl.com/CPAN/>



Perl,DBI の情報源 # 書籍

- ★ Perlリソースキット

ISBN: 4-900900-65-6

- ★ 実用Perlプログラミング

ISBN: 4-900900-82-6

- ★ Perlモジュール活用ガイド

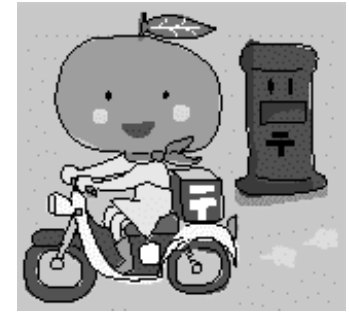
ISBN: 4-88135-668-2

- ★ Perl DBI(英語) Amazon.comにて発売中

DBI アプリケーションの実例 (なっちゃんの郵便屋さんの紹介)



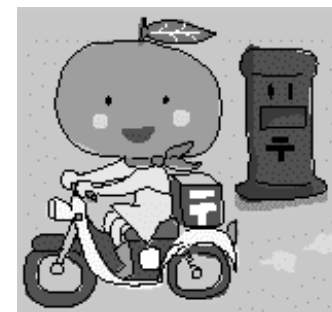
なっちゃんの郵便屋さん とは？



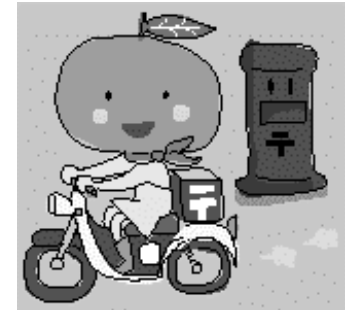
- ★ メールアドレス非公開でメールのコミュニケーションを行う出会いサイト
- ★ 登録は無料
- ★ 会員間のメールの他に掲示板やチャット
- ★ 1日あたりの送信者の送信数および受信者の受信数の制約が可能
- ★ 会員数5347名(2000/03/13現在)
- ★ 1日20～50名の新規入会者

紹介された雑誌/サイト

- ✳ INTERNET ST@RT1月号
- ✳ Mac Peaple2月1日号
- ✳ komuro.com(ネットダイブよりリンク)



History



1999年04月---3Webサーバにてスタート

(このときはWeb+Databaseではない)

1999年05月---Web+Databaseシステムに移行するに伴い
ホスティングサーバを検討

1999年06月---“natsumikan.com”ドメインを取得
同時にサーバ移転

1999年10月---掲示板を設置

2000年01月---試験的にJavaチャットを設置
“natsumikan.net”ドメインを取得

2000年02月---アカウントを忘れた会員のために
アカウント検索システムを設置

システム構成

サーバ : Sun Solaris(SunOS5.6)

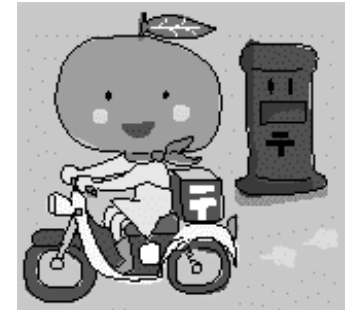
Web : Apache

開発言語 : Perl Ver5.005_03

Javascript

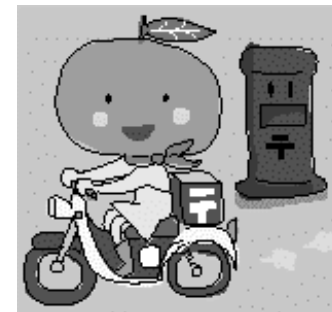
接続インターフェース : DBI&DBD(mysql)

Database : MySQL Ver3.22.29

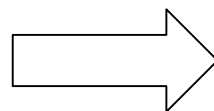


システムの流れ1

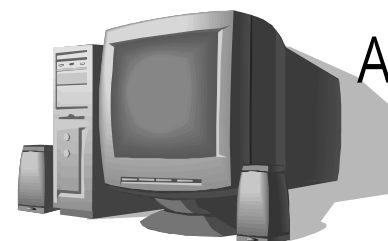
(アカウントの登録)



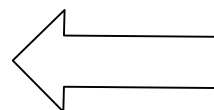
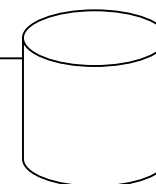
アカウントの登録



Natsumikan.com



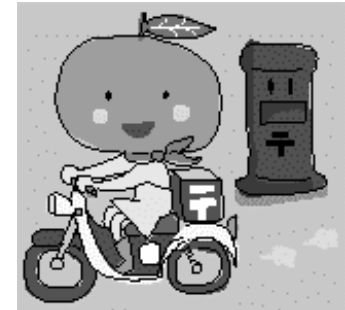
Account Table



メールでアカウント
を知らせる

- アカウントの作成
- メール送信

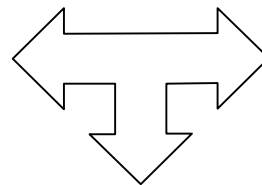
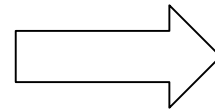
システムの流れ2 (ログイン)



問題なければ
会員ページを表示



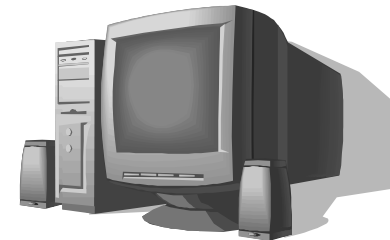
ログイン



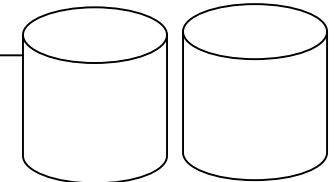
プロフィール未登録なら
プロフィール登録画面へ

アカウントが見つからない
もしくは停止会員なら
エラーメッセージ

Natsumikan.com

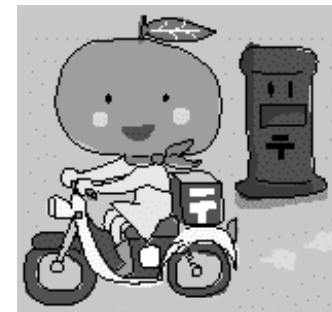


Account Table
Member Table

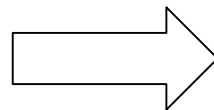


- アカウントを検索
- プロフィール登録済みチェック
- 停止会員チェック

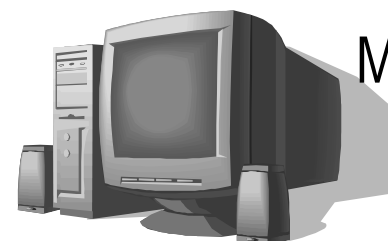
システムの流れ3 (プロフィール登録)



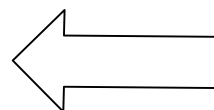
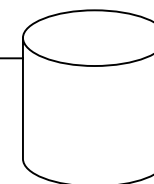
プロフィール登録



Natsumikan.com



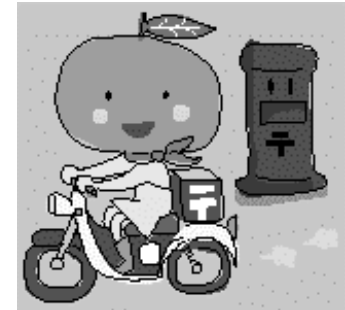
Member Table



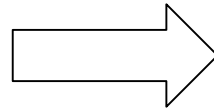
プロフィール登録
終了メッセージを
表示

・プロフィール登録

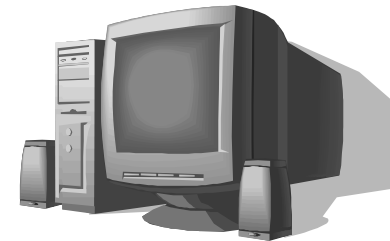
システムの流れ4 (会員検索)



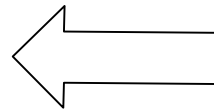
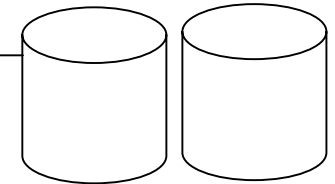
検索条件を入力



Natsumikan.com



Account Table
Member Table

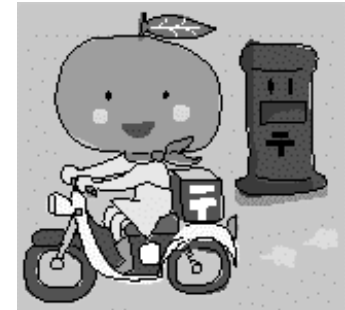


検索結果を表示

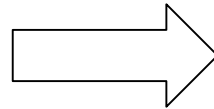


- 条件にあった会員を検索
- 停止会員チェック
- 検索件数が500件を超えるか

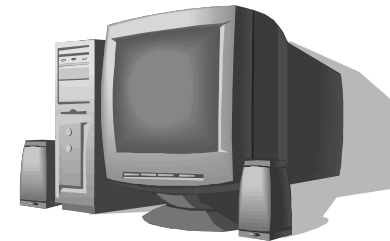
システムの流れ5 (会員プロフィール表示)



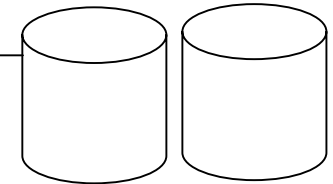
会員を選択



Natsumikan.com



Account Table
Member Table



プロフィールを表示

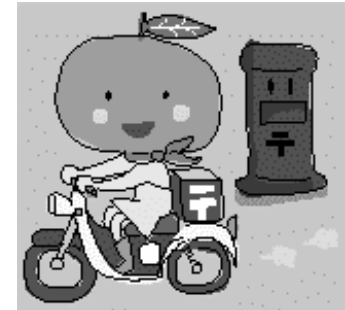


検索結果を表示

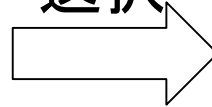
・該当会員のプロフィールを検索

システムの流れ6

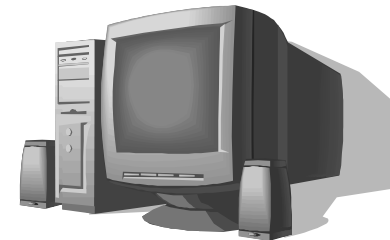
(受信会員選択)



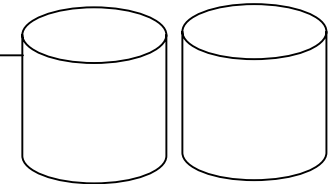
送信先会員を
選択



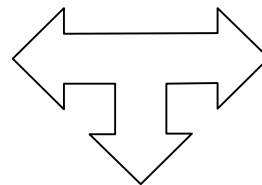
Natsumikan.com



Account Table
Member Table



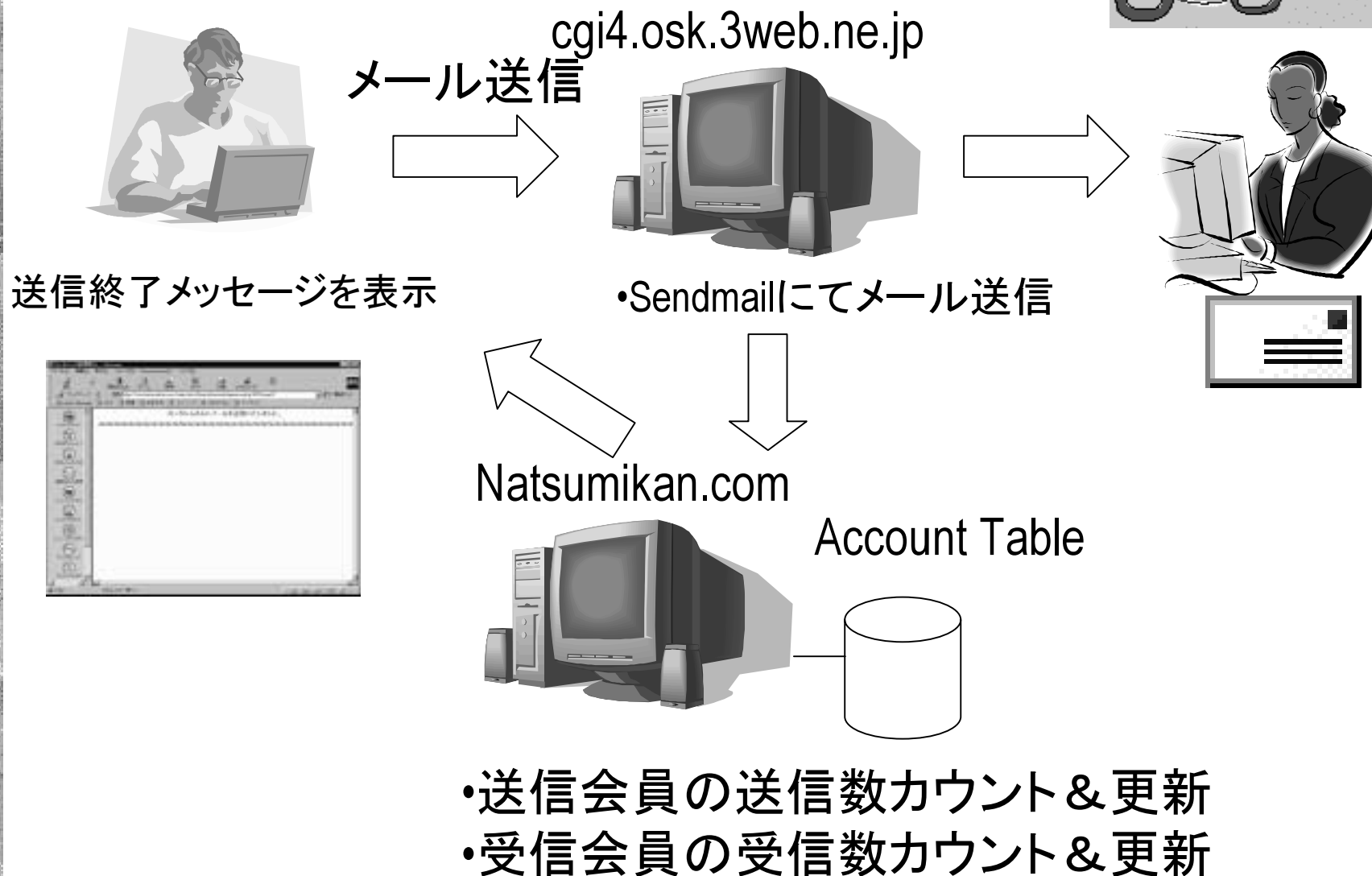
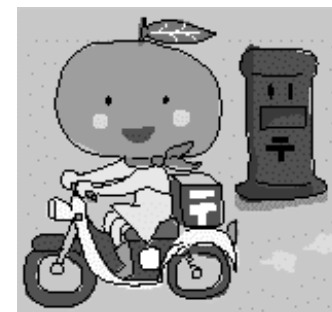
問題なければ
入力フォームを表示



- 送信会員の送信数チェック
- 受信会員の受信数チェック

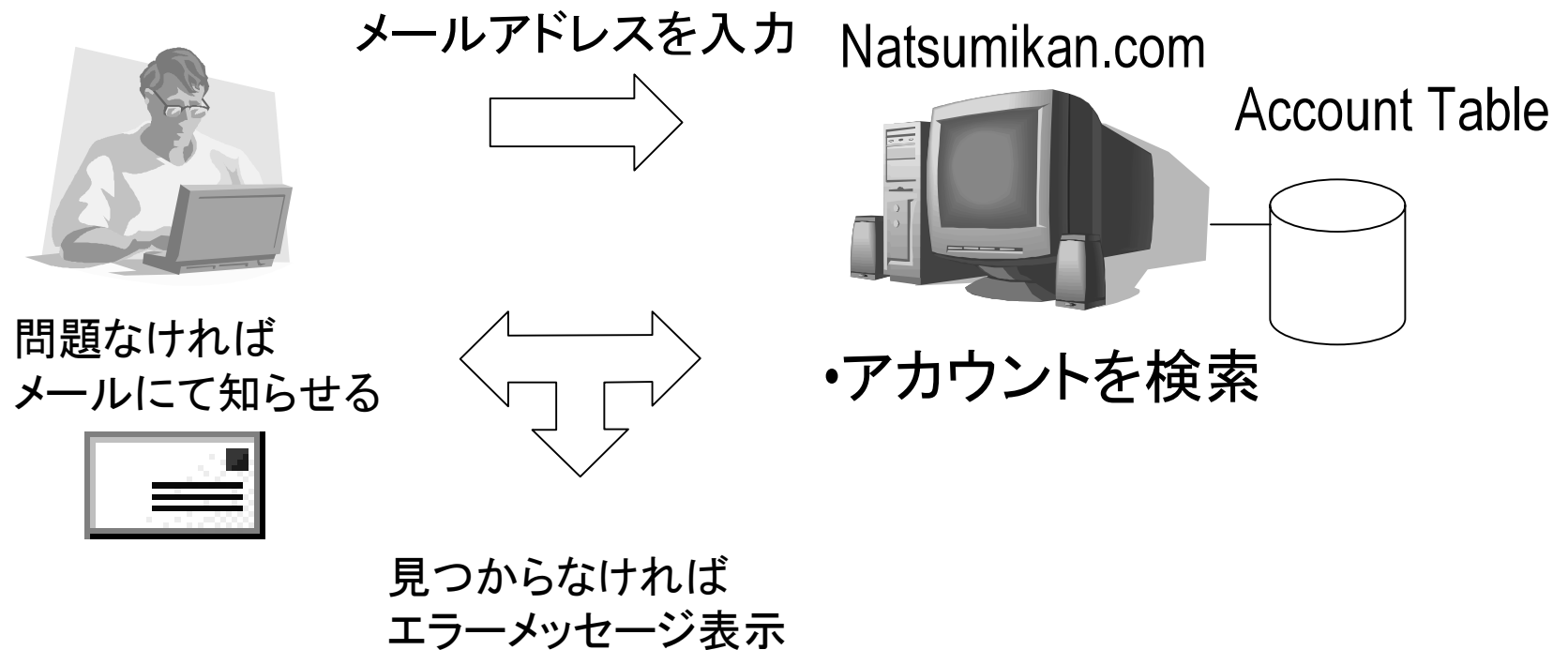
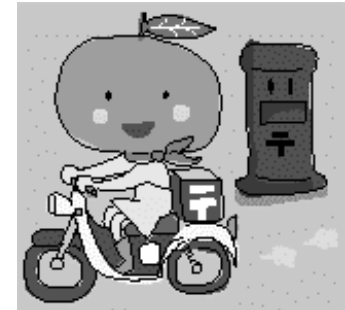
送信数および受信数が越えた場合は
エラーメッセージ表示

システムの流れ7 (メール送信)



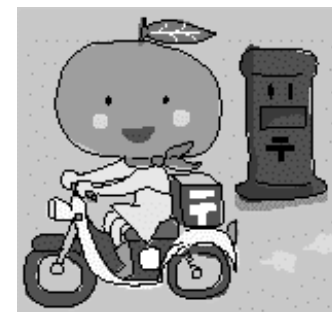
システムの流れ8

(アカウント検索)



システムの流れ9

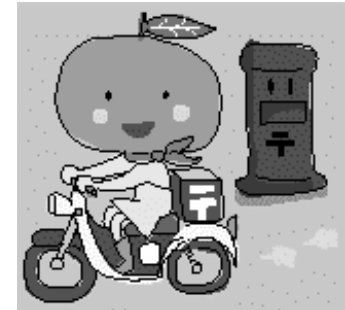
(その他)



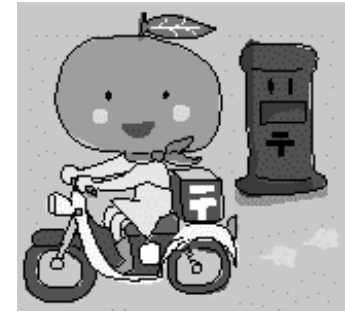
- ✳ 掲示板
- ✳ 会員メンテナンス(会員検索,メール送信等)
- ✳ 1日毎の処理回数の集計(ログイン回数,検索回数等)
- ✳ 入会時アンケートの集計

なっちゃんの野望

- ★ 会員を増やしたい(特に女性)
- ★ i-mode版の開発
- ★ ビジネスしたい!!(主に広告)

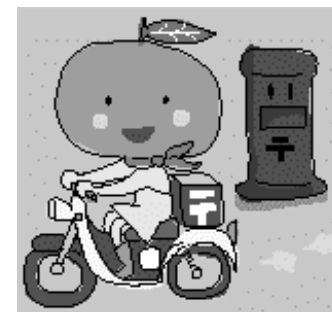


なっちゃんの悩み



- ✳ 高アクセス時でのサーバレスポンス
- ✳ ホスティング会社との契約での制約
(CGIの第三者にレンタル使用禁止等)
- ✳ 安定したメール送信

最後に...



「なっちゃんの郵便屋さん」のURLは...

<http://www.natsumikan.com/>

または

<http://www.natsumikan.net/>



ご静聴ありがとうございました。

Kansai Perl Mongers

Perl Mongers is a not-for-profit organization whose mission is to establish Perl user groups. In the legal sense, Perl Mongers is simply a legal entity completely separate from any Perl user group. Perl Mongers is organized under the laws of the state of New York.