

新聞資料庫試作

林 文 賢* 嚴 竹 華**

* 東吳大學日文系副教授

** 康寧護資專資管科講師

中文摘要

本論文概述作者在網路上建置的新聞資料庫製作過程；包括電視新聞的文字化、資料庫化以及網路化。

新聞データベースの制作について

林 文 賢* 嚴 竹 華**

* 東吳大學日文系副教授

** 康寧護資專資管科講師

要 旨

本論文は作者がウェブに公開しているニュースデータベースの製作過程の報告であり、テレビ新聞の文字化、データベース化、ウェブ化などに触れる。

A Report on Building a News Database

Lin Wen-Shian* Yan Chu-Hwa**

* Associate Professor Department of Japanese Language and
Culture Soochow University

** Instructor, Information Management Department Kang-Ning
Junior College of Medical Care and Management

Abstract

In this paper, the writers report the manufacture process of a news database open to world wide web. It touches on the literation of a TV news, database creation and how to open to web.

新聞資料庫試作

林 文 賢* 嚴 竹 華**

* 東吳大學日文系副教授

** 康寧護資專資管科講師

一、前 言

自從畢昇發明活字版印刷以來，文字傳播，夾著文字儲藏及積累知識的驚人力量，直到現在依然是傳播的主流。爾後電傳視訊，網路通訊，力道雖然剛猛，但是撼動不了文字傳播的寶座。

以日語教學來說，起自芝山岩國語傳習所的日語教學，靠的不只是面對面的聲音傳播，形之於教材的印刷文字傳播，依然有莫大的威力。電傳視訊時代來臨後，雖然引爆了媒體革命，歷經空中日語教室（收音機の日語教學），空中大學電視日語課程開播，乃至晚近遠距教學同步及非同步日語課程，依然得依靠文字傳播的力量，才能更加發揮媒體的教學效果。

本論文概述影音檔案的文字轉換過程；如果，聲音、影像、文字都作符號看，那麼本論文處理的是符號的操作。包括符號的轉換、儲存與傳布；重點在於廣播新聞的文字化、資料庫化以及網路化。

二、文字化

我們從 1998 年暑假，開始對 NHK 兒童新聞週刊進行錄音與錄影。用的是 SONY MZ-R50 型的 MD WALKMAN。這是一種小型的數位錄音機。因為是數位錄音，所

以音質良好，播放出來，簡直跟電視播出的沒什麼差別。NHK 兒童新聞週刊節目長達 30 分鐘。但是，我們只擷取裡頭 4~5 分鐘的新聞部份。這是一個每週一次，以兒童為對象的節目。用字措詞，簡單明確，最重要的是以對話的方式播講新聞；非常適合中級日文的學習者利用。我們經常挑選合適的新聞錄音，在中級日文的課堂上逐句播放，讓學生聽寫。

這些新聞稿，最初由作者及日文系研究生，當家課聽寫，然後繕打成文字檔當作作業繳來。最後轉成網頁檔，置放於網頁。（請造訪 <http://mail.scu.edu.tw/~mark/sinbun.htm>）

1999 年 3 月，日籍教師柳本真理子加入團隊，專責錄音錄影，並負責將新聞部份文字化；柳本老師的加入，大大提升文字化的品質。1999 年至今，我們累積了上百週的新聞剪輯在網路上頭。以每週 4~5 則新聞來算，至今已有 700 多則新聞廣播稿。

2001 年 2 月，我們開始架設一種利用串流語音技術的伺服器，叫做 RealServer。這是購買影音轉換軟體 RealProducer 所附贈的測試版，人數限制在一次最多只能 25 人連線。從這時開始，我們學習利用 SMIL 來讓聲音影像與文字同步播放。SMIL 發音和 smile 一樣，意謂 Synchronized Multimedia Integration Language。SMIL 算不上程式語言，是一種描述語言，用來描述何時及如何播放影音資料。我們用這種語言讓文字註解聲音，提升語音的理解度。（請造訪 <http://mail.scu.edu.tw/~mark/sinbun.htm>，並點擊 <http://webjapanese.qjp.scu.edu.tw:3814/ramgen/News20010303.smi>）

三、資料庫化

文字化的資料，早先我們是在麥金塔上利用 HyperCard 程式語言來處理。HyperCard 一如其名，是利用卡片的觀念來處理資料；我們將一則一則文字化的廣播稿輸入 HyperCard 內。（如圖一）

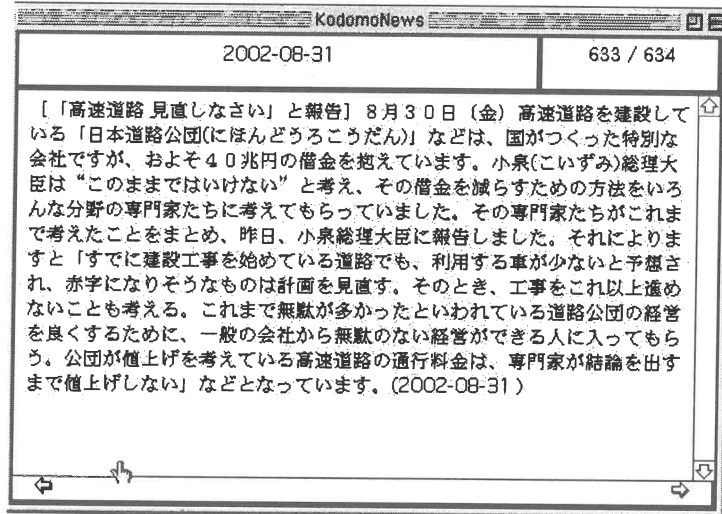


圖 一

這些進電腦的資料，利用 HyperCard 語言來查詢，也能發揮速度快的功能。但是 HyperCard 也有不方便的地方。第一，台灣使用麥金塔的人口不多，資料交換不易。第二，將資料庫擺在麥金塔的 HyperCard 上，無法透過網路提供給他人擷取使用。因此，在 2001 年我們脫離微軟的掌控，使用免費的 linux 作業系統，並利用免費的 PostgreSQL 資料庫查詢語言，搭配 PHP 網頁寫作語言，將資料庫上網公開（資料庫規格如附錄一），提供眾人擷取。（請造訪 <http://webjapanese.qjp.scu.edu.tw/~marklim/newstext.php>）

資料庫相關畫面如圖二：

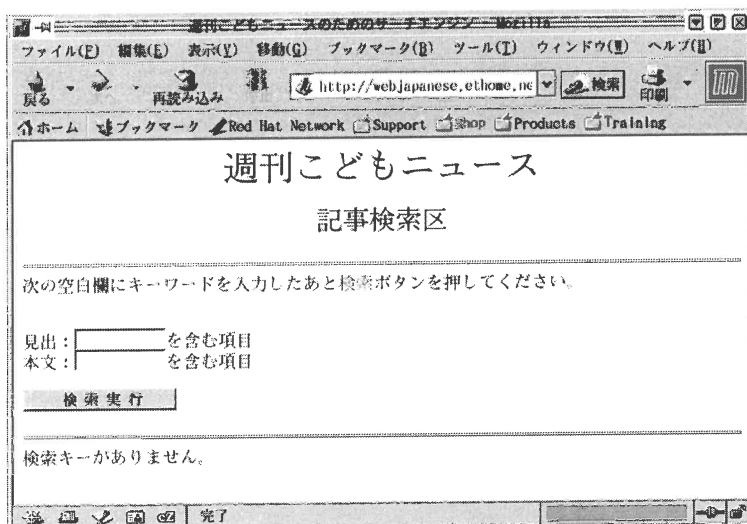


圖 二

如以關鍵字「李登輝」查詢，則檢索出資料一筆如圖三（程式碼如附錄二）。

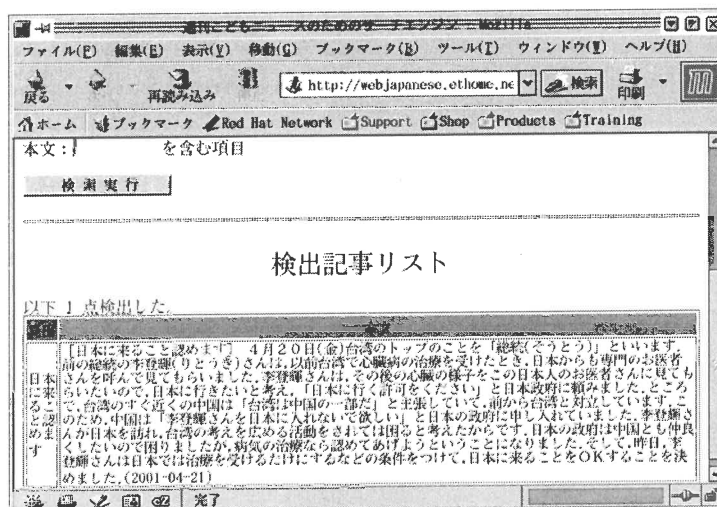


圖 三

四、網路化

單機電腦和網路電腦最大的差別，在於單機電腦的資料只能定點使用；而網路電

腦的資料，能發揮網路無遠弗屆的功能，提供全世界的學習者使用。本研究的新聞資料庫為上網公開，提供服務，本研究團隊試驗性地在研究室電腦上架設伺服器。計有：Apache 網頁伺服器、RealServer 影音伺服器、PostgreSQL 資料庫查詢伺服器、PHP 網頁寫作引擎。茲分述如下：

1. Apache 網頁伺服器

這是最流行的網頁伺服器。相較於微軟 15 人版的個人網頁服務 (pws)，Apache 伺服器可以提供 1500 人以上的來訪容量，算是大哥級的伺服器。免費、穩定、功能強大、容易安裝，是 Apache 網頁伺服器最大的特點。

2. RealServer 影音伺服器

這是 RealNetworks 所發行的影音伺服器。這個伺服器處理的是經過 RealProducer 所壓縮過的影音檔，格式是 *.rm。一般的聲音檔格式是 *.wav，檔案非常大。一個一分鐘的新聞廣播，大概要一個 Mega Byte 的容量。即便成功地把 wav 送上網路上；點擊這個檔案的使用者依然得將一個 Mega Byte 全部下載完後，才能播送。寬頻用戶，或許不覺得有什麼問題；但是，一般的電話撥接用戶，可不一定能順利地下載檔案。

RealProducer 這個軟體，就是用來解決這個問題。RealProducer 能將影音檔壓縮為原來的十分之一，大大減輕網路的交通負擔。但是，影音檔一般都很大，壓縮完後，維持在一個 Mega Byte 的檔案也是常有的事。網路交通的問題依然存在。

RealServer 影音伺服器可以解決這個問題。RealServer 影音伺服器採用最新的影音串流技術，當你點擊檔案後，不必全部下載完畢就可以播放。RealServer 影音伺服器直接在伺服器上播放，點擊檔案後，收到的不是檔案本身，而是播放出來的聲音。

3. postgresSQL 資料庫查詢伺服器

SQL 是 IBM 發展出來的資料庫查詢語言，現在已經成為資料庫程式語言寫作的標準。在 linux 系統上，postgresSQL 通常可搭配 PHP 程式語言使用。

4. PHP 引擎

PHP 程式語言是動態網頁寫作的新寵。一般的超文件標示語言 (HTML) 所寫的

網頁都是靜態網頁。動態網頁從來都得利用共通閘道界面（CGI）配合外部語言如 C、Perl、Visual Basic 等來寫作。PHP 程式語言的出現打破了這個觀念。PHP 程式語言鑲嵌在 HTML 語言裡面；因此不須外部語言來處理。但是，PHP 程式語言須有 PHP 引擎才能執行。

首先，我們安裝 Apache，以便提供網頁服務——讓作者能透過網頁管理資料庫，也讓使用者能透過網頁使用資料庫。我們將資料存到 PostgreSQL 的資料庫（資料庫欄位規格如附錄一），然後使用 PHP 語言寫些小程序，以便管理及維護資料庫。麥金塔上的舊資料是透過轉檔進到資料庫；至於新的資料則從網頁上登錄（如圖四，其表單如附錄三）。

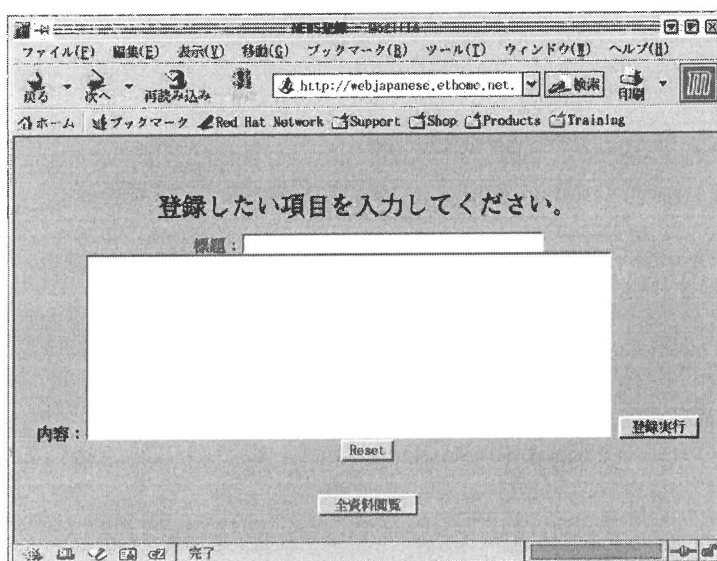


圖 四

我們透過電子郵件取得柳本老師聽寫的廣播稿，一則一則地貼到恰當的欄位（如圖五）。

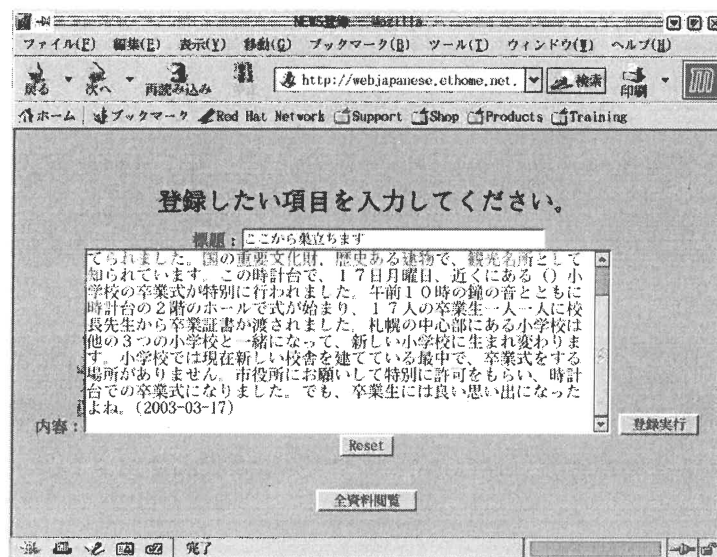


圖 五

剪貼完畢，按下【登録実行】，會有畫面告知是否登録成功（如圖六，其 PHP 程式如附錄四、五）。



圖 六

四、教學運用

本資料庫適合中級學習者，不適合初級使用。教師與學習者可以透過網際網路直接取得文本使用。也可以運用資料庫查詢系統，取用所需的資料。

五、著作權

本資料庫所處理的都是時事報導。按照日本的著作權法規定，「事實の伝達にすぎない雑報及び時事の報道は（中略）著作物に該当しない¹。」因此沒有侵犯著作權法的問題。

六、未來方向

本研究目前只提供網頁、資料庫及新聞討論群組的資料。未來將朝向建構郵遞論壇（mailing list）的方向進行。郵遞論壇一旦建構完成，未來只要有新的廣播稿或使用者的意見進來，即可藉由電子郵件轉寄服務，寄發給登錄在案的每一位使用者。

參考文獻

中文

Mis2000 Lab (2002)。 Linux:Mandrake 8.2 入門、管理與運用。台北：文魁資訊。

位元文化 (2000)。 PHP4&MySQL 動態網頁入門實務。台北：文魁資訊。

1 日本著作權法第2章第1節第10條。

林世捷・林金微・蔡修偉 (2002)。 Red Hat Linux 資料庫實務：使用 PostgreSQL。台北：文魁資訊。

李立功・趙揚 (2000)。 MySQL 程式設計與資料庫管理。台北：文魁資訊。

吳佳彥 (2001)。 LINUX 玩家寶典：Red Hat Linux 7.2 + Mandrake 8.1。台北：文魁資訊。

施威銘研究室 (2001)。 Linux 7 架站實務。台北：旗標。

蔡奇玉・連振漢 (1996)。 WWW 之 HTML 與 CGI 寫作大全。台北：第三波文化事業。

日文

石井達夫 (2001)。 PC UNIX ユーザのための PostgreSQL 完全攻略ガイド。東京：技術評論社。

廉升烈 (2001)。 PostgreSQL による Linux データベース構築。東京：翔泳社。

サーバー構築研究会編著 (2003)。 Red Hat Linux で作るネットワークサーバ構築ガイド。東京：秀和システム。

堀田倫英・石井達夫・廣川類 (2002)。 PHP 徹底攻略改訂版：Web DB プログラミング徹底入門。東京：ソフトバンク。

廣川類・桑村潤・小山哲志 (2002)。 PHP 徹底攻略実践編：実践的 Web アプリケーション開発技法。東京：ソフトバンク。

附錄一

底下列出 PostgreSQL 指令 pg_dump 所備份的資料庫欄位格式及第一筆資料：

```
--
-- PostgreSQL database dump
--
\connect - marklim
SET search_path = public, pg_catalog;
--
-- TOC entry 2 (OID 57786)
-- Name: list; Type: TABLE; Schema: public; Owner: marklim
--
CREATE TABLE list (
    head text,
    body text
);
--
-- TOC entry 3 (OID 57786)
-- Name: list; Type: ACL; Schema: public; Owner: marklim
--
REVOKE ALL ON TABLE list FROM PUBLIC;
GRANT ALL ON TABLE list TO PUBLIC;
--
-- Data for TOC entry 4 (OID 57786)
-- Name: list; Type: TABLE DATA; Schema: public; Owner: marklim
--
COPY list (head, body) FROM stdin;
```

ウソはモーつかないで [ウソはモーつかないで] お店で、商品の値段を消して安い値段にしました、と売っていることがあるよね。でも、公正取引委員会という役所が調べたところ、大きなスーパー7つが牛肉を売るとき、いったん高い値段をつけて、それを線で消して安い値段を書いていた。ところが、線で消した高い値段で売っていたことはほとんどありませんでした。7つのスーパーは、ダイエー、イトーヨーカ堂、西友、マルエツ、ライフコーポレーション、東急ストア、それに大丸ピーコックです。公正取引委員会は、これは買い物客にいつもの値段より安いと誤解させて売るやり方で、法律に違反していると考え、16日火曜日、7つのスーパーに警告、つまり厳しく注意しました。(1999-03-23)\r

附録二 新聞廣播稿搜尋引擎 (newstext.php) 程式碼

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ecu-jp">
<title>週刊 こどもニュースのためのサーチエンジン</title>
</head>
<body>
  <H1 align="center">週刊 こどもニュース</H1>
  <h2 align="center">記事検索区</h2><hr>
  次の空白欄にキーワードを入力したあと<font color=red>検索</font>ボタンを
  押してください。 <br><br>
  <form action="newstext.php" method="post">
  <br>
  見出 : <input type="text" name="head" value="" size="10">を含む項目<br>
  本文 : <input type="text" name="body" value="" size="10">を含む項目
  <br><br>
  <input type="submit" name="submit" value="  検 索 実 行  ">
  </form><hr>
```

```
<?PHP
```

```
// DB へ接続します。
```

```
$conn = pg_connect("host=localhost port=5432 dbname=newstext user=mark
password=scut012");
```

```
// あらかじめ SQL 文を入力しておきます。
```

```
$query = "select * from newstext ";
```

```
// 検索項目が入っているかどうかを判定させるフラグを初期化します。
```

```
$flag = 0;
```

```
// 検索用関数を定義します。
```

```
function head_fnc($head, $value) {
  if($value != "") {
    global $flag, $query;
    $value = "%" . $value . "%";
    if($flag) {
      $query = $query . " and";
    } else {
```

```

        $query = $query . " where";
    }
    $query = $query . " $head like '$value'";
    $flag++;
}
};

function body_fnc($body, $value) {
    if($value != "") {
        global $flag, $query;
        $value = "%" . $value . "%";
        if($flag) {
            $query = $query . " and";
        } else {
            $query = $query . " where";
        }
        $query = $query . " $body like '$value'";
        $flag++;
    }
};

// 関数を実行します。
body_fnc("head", $_POST['head']);
body_fnc("body", $_POST['body']);
// 検索結果を返します。
if($flag) {
    $query = $query ;
} else {
    print("検索キーがありません。");
    exit;
}
?>
<?PHP
$result = pg_Exec($conn, $query);

// 検索キー該当項目が何行あるか、行数を求めます。
$rowCount = pg_NumRows($result);
?>

```



```

<h2 align="center">検出記事リスト</h2>
<?PHP echo "以下 $rowCount 点検出した。"; ?>
<!-- 検索結果リスト表示 -->
<TABLE BORDER=1 CELLPADDING="1">
<TR>
<TD align="center" bgcolor="55cc33">
    <FONT SIZE="-1">見出</FONT>
<TD align="center" bgcolor="55cc33">
    <FONT SIZE="-1">本文</FONT>
</TD></TR>

<?php
// for 文で検索キー該当項目を表示します。
for($i=0; $i<$rowCount; $i++) {
    // テーブルからデータを取り出して変数に入れます
    // $user_code = pg_Result($result,$i,"user_code");
    // $date = pg_Result($result,$i,"date");
    $head = pg_Result($result,$i,"head");
    $body = pg_Result($result,$i,"body");
    // 実際の表示処
    printf ("
        <tr><td>
            <font size=\"-1\"><b>%s</b></font>
        </td> <td>
            <font size=\"-1\"><b>%s</b></font>
        </td></tr>
        ",$head,$body);
}
// 使用したメモリの解放
pg_freeResult($result);
// DB への接続を切る
pg_close($conn);
?>
</table>
</body>
</html>

```

附錄三 登錄表單

```
<html>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-JP">
<meta name="GENERATOR" content="Quanta Plus">
<title>NEWS 登録</title>
<body bgcolor="#bbaabb"><br><br>
<center><br>
<font size="+2"><b>登録したい項目を入力してください。</b></font>
<form action="toroku.php" method="get">
<br>
標題：<input type="text" name="toroku_head" value="" size="40"><br>
内容：<textarea name="toroku_body" cols="60" rows="10">
</textarea>
<input type="submit" name="submit" value="登録実行"><br>
<input type="reset" name="reset" value="Reset"><br>
</form><br>
<form action="toroku_1.php" method="get">
<input type="submit" name="submit" value="全資料閲覧"><br>
</form>
</body>
</html>
```

附錄四 登錄畫面中「登錄實行」按鈕(toroku.php)的程式碼

```
<html>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-JP">
<meta name="GENERATOR" content="Quanta Plus">
<title>登録結果</title>
<body bgcolor="#bbaabb"><br><br>
<?php
    $conn = pg_pconnect("", "", "", "", "newstext");
    //Connect Data
    if (!$conn) {
        echo "<center>";
        echo "<h1>DB に接続出来ませんでした。 \n</h1>";
        echo "</center>";
        exit;
    }
    echo '<center>';
    echo "<h1>DataBase NewsText has been connected\n</h1>";
    echo '</center>';
    //SQL を作成します。
    $sql = sprintf("insert into list values('%s','%s')",
    $_GET['toroku_head'], $_GET['toroku_body']);
    //no data, reinput
    if (!$_GET['toroku_head']||!$_GET['toroku_body']) {
        echo "<center>";
        echo "<h3>reinput, please\n</h3>";
        echo "</center>";
    }
    echo $_GET['toroku_head'].'<br><br>';
    echo $_GET['toroku_body'];
    echo "<br>";
    //echo $sql;
    echo "<br>";
    //Query
```

```

@$result = pg_exec($conn, $sql);
//実行した query の error を確認します。
if (!$result){
    echo "<center>";
    echo "<h2>key duplicate</h2>";
    echo "</center>";
    exit;
}
else {
    echo "<center>";
    echo "<h2>正常に登録出来ました。¥n</h2>";
    echo "</center>";
}
//memory をクリヤします。
pg_FreeResult($result);
//DB close
pg_close($conn);
?>
</body>
</html>

```

附錄五 登錄畫面中〔全資料閱覽〕按鈕(toroku_1.php)的程式碼

```
<html>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-JP">
<meta name="GENERATOR" content="Quanta Plus">
<title>検索結果</title>
<body bgcolor="#bbaabb"><br><br>
<?php
    $con = pg_connect("host=localhost port=5432 dbname=newstext user=mark
password=scut012");
    //Connect Data
    if (!$con) {
        echo "<center>";
        echo "<h1>DB に接続出来ませんでした。 \n</h1>";
        echo "</center>";
        exit;
    }
    //SQL を作成します。
    $sql = sprintf("select * from newstext");
    echo "<center>";
    echo '<table border =1>';
    echo '<tr align="center">
        <td bgcolor="55cc33"><font size=+1">番号</font></td>
        <td bgcolor="55cc33"><font size=+1">見出</font></td>
        <td bgcolor="55cc33"><font size=+1">本文</font></td></tr>';
    $result_out= pg_query($con,$sql); //select を実行する。
    $num_p = pg_num_rows($result_out);
    $rows_p = 0;
    echo "以下 $num_p 点検出した。 ";
    while ($rows_p < $num_p){
        $head = pg_fetch_result($result_out,$rows_p,"head");
        $body = pg_fetch_result($result_out,$rows_p,"body");
        printf("
        <tr><td align=\"center\">
```

```

        <font size=\"+1\">%s</font>
    </td><td>
        <font size=\"+1\">%s</font>
    </td><td>
        <font size=\"+1\">%s</font>
    </td></td>
    ",$rows_p+1,$head,$body);
    $rows_p++;
}
echo '        </table>';
echo "        </center>";
    //memory をクリヤします。
    pg_FreeResult($result_out);
    //DB close
    pg_close($con);
    ?>
</body>
</html>

```