الفصل العاشر

إنشاء الجداول وقواعد البيانات والملفات

Data Definition Language (DDL)

أخيرا ، سوف نقوم بتكوين الجداول التي عملنا معها عدة فصول سابقة ، ولكننا الآن سوف نقوم بتكوينها بأنفسنا ، أليس ذلك شيقا ؟ ولأننا سنقوم بتكوين الجداول أو انشائها ان أردت ، فيجب أولا أن نقوم بانشاء قاعدة البيانات التي ستحتوي تلك الجداول.

بالانتهاء من هذا الفصل ستكتسب المعارف وتتدرب على المهارات التى تجعلك قادرا على:

- *إنشاء أو تكوين قاعدة بيانات* **Create Database**
- *إنشاء الجداول* **Create Tables**
- *تعديل مواصفات الجدول بعد إنشائه*
- *حذف جدول*
- *حذف قاعدة البيانات*
- *تكوين جدول جديد من جدول موجود*

طبعا سوف نتعرض لعدد من الأدوات والخصائص والاختيارات المصاحبة لتكوين قواعد البيانات وأيضا الجداول ، وقد حرصت أن يأتي هذا الفصل بعد عدد من الفصول التي تعاملنا فيها مع قواعد البيانات والجداول ، لكي يصبح الأمر شيقا من ناحية ، ومن ناحية أخرى فان تعاملك السابق مع قواعد البيانات سوف يعطيك فهما أكبر لما نحن مقدمون عليه في هذا الفصل

إنشاء قاعدة بيانات Create Database

رغم أن هناك العديد من الخصائص والمهام التي يمكن اضافتها أثناء انشاء أي قاعدة بيانات إلا أنك ستجد في معظم الأحيان أنك لن تحتاج إلا إلى تعليمات إنشاء قاعدة البيانات في أبسط صورة كما يلي :

الشكل العام لجملة إنشاء قاعدة البيانات كما يلي :

```
CREATE DATABASE database_name
```

مثال :

لإنشاء قاعدة بيانات باسم **payroll** استخدم المثال الآتي :

MySQL

```
Create database payroll;
```

وبنفس الطريقة يمكن انشاء قاعدة بيانات في كل من **Microsoft SQL** وغيرها . ولن نعلق طويلا على باقي التعليمات التي يمكن أن تضاف الى هذه **query** ففي معظم الأحيان كما قلنا لن نحتاج للمزيد ، وربما نتحدث عن باقي التعليمات في فصول أخرى .

لكي نتعامل مع قاعدة بيانات موجودة بعد إنشائها استخدم أمر **USE** كما يلي:

```
MYSQL > USE payroll
Database changed
```

إنشاء قاعدة بيانات باستخدام Oracle

يختلف إنشاء قاعدة البيانات مع نظام **Oracle** قليلا ، حيث يعتبر أوراكل أن إنشاء قاعدة بيانات عبارة عن إنشاء بيئة تشغيلية ، لذلك فعندما تريد إنشاء قاعدة بيانات فانك تقوم بإضافة مستفيد وبمجرد إضافة مستفيد " **user** " وإعطاء هذا المستفيد الصلاحيات اللازمة تقوم أوراكل بإنشاء قاعدة بيانات وتخصص لها الاسم الذي أعطيته للمستفيد .

في الإصدارات السابقة كانت تسمى قاعدة البيانات schema وكان يتم إنشاؤها بأمر CREATE SCHEMA . الآن يكفي إنشاء المستفيد لتقوم أوراكل بتكوين قاعدة البيانات تلقائيا . يتم إنشاء قاعدة البيانات بأمر CREATE USER كما في المثال التالي :

فيما يلي خطوات إنشاء قاعدة البيانات باسم payroll عن طريق إضافة user ومنح صلاحيات له .

يجب أولا إنشاء مستفيد باسم payroll وتخصيص كلمة السر (pay مثلا)كما يلي :

```
create user  payroll  IDENTIFIED BY pay ;
```

بعد ذلك يجب إخبار النظام أن يستخدم معطيات users عندما يقوم بتحديد حجم الجداول بأمر :

```
alter user payroll default tablespace users ;
```

ثم يتم إخبار النظام أن يستخدم معطيات temp عندما يقوم بتحديد حجم الجداول المؤقتة بأمر

```
alter user payroll temporary tablespace temp ;
```

إذا لم تحدد حجم الجدول بأمر Default Tablespace وأمر Temporary Tablespace فإن أوراكل تخصص له الحجم المناسب ولذلك يمكن أن نستغني عن هذين الأمرين .

الخطوة التالية هي إعطاء صلاحية للمستفيد payroll تمكنه من العمل كمدير قواعد بيانات " dba " داخل البيئة التي تم إنشاؤها payroll وهي نفسها تمثل قاعدة بيانات في الأنظمة الأخرى بالأمر:

```
grant dba to payroll ;
```

بالأمر :pay وكلمة السر كما ذكرنا هي payrollأخيرا يتم الارتباط بقاعدة البيانات

```
connect payroll/pay ;
```

لمزيد من المعلومات عن أوامر إنشاء قاعدة البيانات ومنح الصلاحيات والاتصال بقاعدة البيانات التي شرحناها هنا راجع الفصل السابع عشر .

تدريب عملي :

في التمرين التالي سوف نقوم باستخدام **SQL*Plus Worksheet** التي شرحناها في الفصل الثاني لإنشاء قاعدة بيانات **payroll** التي تحدثنا عنها، اتبع الخطوات التالية :

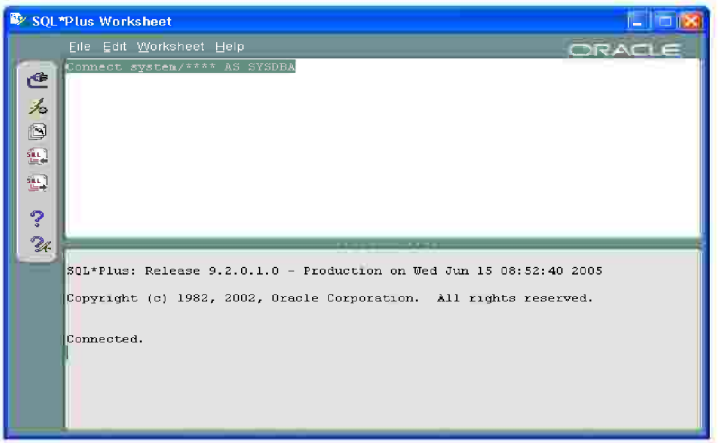
١. من قائمة **Start** اختر البرامج **Programs** ومن قائمة البرامج **programs** اختر **Oracle – Orahome92** تظهر قائمة أدوات أوراكل.

٢. اختر **Integrated Management Toll** . تظهر مجموعة أخرى من الأدوات . اختر **SQL Plus Worksheet** ، يظهر مربع الحوار **Oracle Enterprise Manager Login** ؟(راجع الفصل الثاني من الكتاب) .

٣. أمام مربع **username** قم بإدخال اسم المستفيد **system** وأمام مربع **password** اكتب كلمة السر التي قمت بتكوينها أثناء تثبيت البرنامج ثم اكتب **Sysdba** أمام مربع **Connect & As** ثم انقر الزر **Ok** انظر الشكل (١٠– ١) .

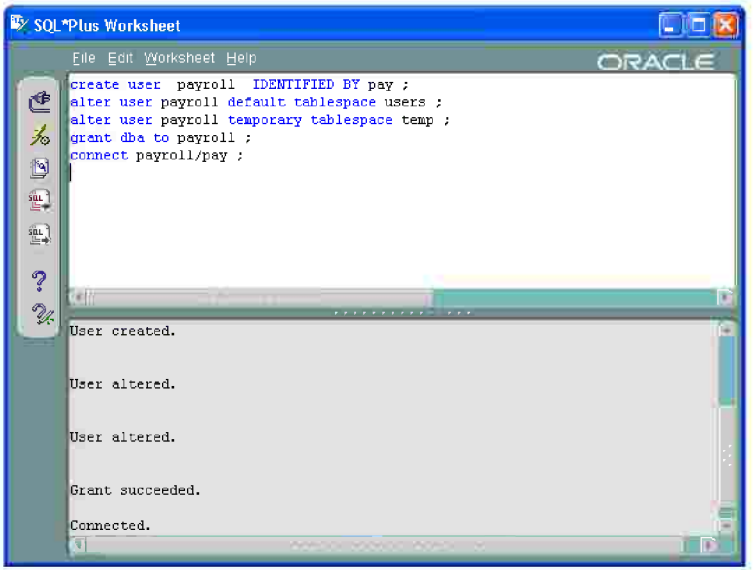
شكل ١٠–١ مربع **username**

٤. يظهر مربع الحوار Oracle SQL*Plus Worksheet حيث يمكنك كتابة كل الأوامر وتنفيذها كما في شكل (١٠-٢) .



شكل ١٠-٢ ورقة العمل

٥. اكتب جمل SQL التي شرحناها في المثال السابق داخل الجزء العلوي ثم اضغط مفتاح F5 ومعناه نفذ . شكل (١٠-٣) .



شكل ١٠-٣ ورقة العمل بعد إدخال كلمة SQL

أصبح لديك الآن قاعدة بيانات باسم **Payroll** وهي أيضا اسم مستفيد ، وبعد تنفيذ الأمر الأخير **Connect payroll/pay** فانك الآن تعتبر متصلا بقاعدة البيانات الجديدة **payroll** يمكنك إنشاء جداول وغير ذلك باستخدام جمل **Query** كما ستعرف بعد قليل .

إنشاء الجداول Create Tables

ومن الأمور الهامة عند تصميم قواعد البيانات تحديد هيكل الجدول بحيث يحتوي علي مفتاح أساسي (**Primary Key**) ومفتاح خارجي (**Foreign Key**) ويجب أن يحقق المفتاح الأساسي الهدف من عدم تكرار بيانات السجل الواحد أكثر من مرة . أما حقل المفتاح الخارجي فيقوم بربط الجدول مع جدول آخر بناء علي حقل مفتاح أساسي أو خارجي . والمثال التالي يوضح ذلك :

سوف تعرف بالتفصيل استخدام المفتاح الاساسي والمفتاح الخارجي في الفصل القادم.

مثلا عندما ننشي جدول يحتوي بيانات الفواتير ونسميه **Invoice** فانه سوف يحمل البيانات التالية :

اسم الحقل	طول الحقل	معناه
Invoice_No	8	رقم الفاتورة
Customer_Name	30	اسم العميل
Customer_Id	5	رقم العميل
Amount	6	قيمة الفاتورة
Customer_Address	40	عنوان العميل
City	15	المدينة
State	15	المحافظة

هذا الجدول سوف يعمل بدون أية مشاكل ، ولكن تخيل معي كل مرة سوف نقوم باصدار فاتورة جديدة لنفس العميل فاننا سوف نقوم بادخال بيانات العميل مرة أخرى من اسم ورقم وعنوان والمدينة والمحافظة ، تخيل اصدار ٣٠ فاتورة مثلا لعميل واحد و ١٠٠ فاتورة لعميل آخر وهكذا فكيف ستمتليء قاعدة البيانات بحجم وبيانات مكررة لا داعي لتكرارها أبدا ، يكفي أن يكون لدينا جدول باسماء وعناوين العملاء يكتب فيه بيانات العميل كالتالي :

اسم الحقل	معناه
Customer_Name	اسم العميل
Customer_Id	رقم العميل
Customer_Address	عنوان العميل
City	المدينة
State	المحافظة

وبالتالي يكون جدول الفواتير **Invoices** في الصورة التالية :

اسم الحقل	معناه
Invoice_No	رقم الفاتورة
Customer_Name	اسم العميل
Customer_Id	رقم العميل

وبهذا فان بيانات العميل سوف نقوم بادخالها مرة واحدة وعند اصدار الفاتورة يتم سحب بيانات العميل من الجدول الخاص بالعملاء **Customers** أليس ذلك أفضل تنظيم ؟

ولكي نقوم ببناء الجداول في قاعدة البيانات فاننا نحتاج للالمام ببعض القواعد التي تتبع مع مكونات أي جدول وهي :

- اسم الجدول **Table Name** وشروط اختيار الاسم
- اسماء الحقول داخل الجدول **Fields Name** وشروطها
- أنواع البيانات لكل حقل **Field's Data Type**

اسم الجدول Table Name

عند إنشاء جدول يجب الالتزام بعدة قواعد لتسمية الجدول هي :

- يتكون اسم الجدول من ١ الى ٣٠ من الحروف الانجليزية كلها فيما عدا "/" أو "."
- أول حرف في اسم الجدول يجب أن يكون من **A** الى **Z** والباقي ممكن أن يشتمل على حروف أو أرقام أو بعض الحروف الخاصة مثل " _ "," @ "," # "
- يجب عدم تكرار اسم الجدول في قاعدة بيانات واحدة
- في بعض قواعد البيانات مثل **Oracle 9i** يمكن استخدام الحروف الكبيرة أو الصغيرة على السواء **Upper or Lower Case** في أسماء الجداول
- يمنع استخدام الكلمات المحجوزة في أسماء الجداول مثل **SELECT**
- في قواعد البيانات **MySQL** يمكن أن يصل الاسم الى ٦٤ حرف

أسماء الحقول Field Names

لا يمكن تكرار اسم الحقل داخل جدول واحد ، بالطبع يمكن تكرار اسم الحقل داخل قاعدة البيانات ولكن في جداول أخرى ، مثلا يمكن استخدام **Customer_Id** كاسم لحقل ولكن في جدول **Invoices** مرة وفي جدول **Customers** مرة أخرى .

أنواع البيانات Data Types

المقصود هو نوع البيانات التي سيتم تخزينها في كل حقل من الحقول ، وهذا يعني أن كل حقل نقوم بإنشائه داخل الجدول يجب أن يكون له نوع ومساحة فمثلا حقل سيستقبل

اسم فيه يختلف عن حقل سيستقبل رقم وغير ذلك .

الجدول التالي ١-٩ يشتمل على معظم أنواع البيانات التي يمكن تخصيص الحقل بها وشرح لهذه الأنواع :

نوع البيانات	معناها أو استخدامها
CHAR(SIZE)	هذا النوع يستخدم مع الحقول التي سوف نستخدمها في تخزين الحروف ، ومساحة البيانات التي نكتبها تحدد طول الحقل ، فمثلا لو كتبنا CHARS(10) معناها أننا حجزنا حقل طوله ١٠ حروف ، فلو كتبنا فيه ٤ حروف فسوف يتم اضافة ٦ مسافات
DATE	هذا الحقل يتم حجزه للتاريخ حسب طريقة قواعد البيانات في كتابة التاريخ فبعضها يتكون من القرن والسنة والشهر واليوم والساعة والدقيقة والثانية
LONG	هذا الحقل يتم حجزه لاستخدامه في الحروف والكلمات وقد يصل طوله الى ٢ جيجابايت ، ويستخدم في كتابة ملاحظات أو بيانات كبيرة وسوف نوضحها فيما بعد
LONG RAW	نوع هذا الحقل BINARY ويصل طوله الى ٢ جيجابايت
NUMBER	يستخدم هذا النوع في حجز حقل لاستخدامه في الأرقام سواء الموجبة أو السالبة ويمكن حجزه بطريقتين : NUMBER(SIZE) ويتم تحديد طول الحقل بالخانات NUMBER(SIZE,D) وهنا يتم حجز طول الحقل بعدد الأرقام
SMALLINT	يستخدم كاستخدام النوع السابق NUMBER
RAW(SIZE)	يستخدم في التعامل مع BINARY لطول يصل الى 255Bytes

نوع البيانات	معناها أو استخدامها
raw MLSLABEL	يستخدم مع بعض أنظمة التشغيل Binary format for a Secure Operating System labels
Rowid	يتم استخدام هذا النوع لكي يتم حفظ عنوان فريد لكل حقل
Varchar2(size)	يستخدم هذا النوع لتخزين بيانات ذات أطول مختلفة تتراوح ما بين ١ الى ٢٠٠٠ حرف أو رقم
Varchar(size)	نفس استخدام varchar2 وقد يكتفى في المستقبل باحدهما
Integer	نفس استخدام number ولا يسمح بالعلامة العشرية
Integer(n)	يتم تحديد عدد الأرقام التي يمكن تخزينها في الحقل
Long varchar	نفس استخدام long

دائما ما نطلق على Long أنها Memo Type لأنها في الواقع تستخدم في كتابة المذكرات أو الحواشي وغير ذلك مما يأخذ كتابة كبيرة الحجم .

أما Long Raw فيطلق عليها في بعض قواعد البيانات Binary Large Object (BLOB) والحقول من هذه النوعية تستخدم في تخزين الصور وملفات الصوت .

تأكد من نظام قواعد البيانات التي تستخدمه لترى مدى مساندة هذا النظام لأنواع البيانات التي تريد استخدامها في الحقول فمثلا سوف تجد أن Number غير مستخدمة في MySQL وبدلا منها نستخدم Numeric(n) أو Int(n).

تعال معي ننشئ جدول الفواتير **invoices** وجدول العملاء **customers** باستخدام **MySQL** على النحو التالي :

أولا سننشيء قاعدة البيانات :

```
> create database invoicing;
Query OK, 1 Rows affected (0.06 sec)
> use invoicing;
Database changed
```

ثم ننشيء الجدول الأول :

```
> create table invoices (
> invoice_no varchar(8),
> customer_id  varchar(6),
> amount numeric);
Query OK, 0 Rows affected (0.06 sec)
```

ثم الجدول الثاني :

```
> create table customers (
> customer_id varchar(8),
> address  varchar(40),
> city  varchar(15),
> state varchar(15));
Query OK, 0 Rows affected (0.08 sec)
```

القيمة **NULL** *و القيمة* **NOT NULL**

هل هناك فرق ؟ طبعا يعرف كل منا وخاصة مطوري التطبيقات أن هناك فرق بين **NULL** و **NOT NULL** فهناك فرق بين خانة بها مسافة أو خانة بها صفر فالمسافة **Blank** والصفر هما قيم هامة ومؤثرة وتختلف تماما عن خانة ليست بها أي قيمة فيجب الاحتياط عند تعريف الحقول والتفريق بين لا قيمة **NULL** وعدم ترك الحقل بدون قيمة **NOT NULL** وهذا يعتمد على طريقة تصميمك للجداول واستخدامك للحقول .

عند إنشاء الجداول تسمح معظم نظم قواعد البيانات بتعريف الحقل باستخدام NOT NULL . ومعناها أن العمود لا يجب أن يحتوي علي قيمة NULL أي لابد أن يحتوي علي قيمة في جميع سجلات الجدول والمثال التالي يوضح استخدام NOT NULL :

```
SQL > CREATE TABLE INVOICES(
  NAME CHAR (15) NOT NULL,
  AMOUNT NUMBER,
  ACCOUNT_ID NOT NULL);
```

وباستخدام MYSQL فإن المثال السابق يصبح كما يلي :

```
SQL > CREATE TABLE INVOICES(
      > NAME CHAR (15) NOT NULL,
      > AMOUNT NUMERIC,
      > ACCOUNT_ID NUMERIC NOT NULL);
```

تعديل مواصفات الجدول بعد إنشائه :

كثيرا ما تحتاج لتعديل محتويات الجداول بعد إنشائها . يستخدم أمر ALTER TABLE بواسطة مسئول قواعد البيانات أو المصمم لتغيير مواصفات الجدول بعد إنشائه .ويوفر هذا الأمر حذف الجدول بعد إنشائه عند الرغبة في تعديله وعموما يستخدم أمر ALTER TABLE في :

- إضافة عمود (حقل)لجدول موجود .
- تعديل مواصفات جدول موجود .
- حذف حقل موجود .

والشكل العام لأمر ALTER TABLE كما يلي :

```
ALTER TABLE table_name
< ADD column_name data_type |
< MODIFY column_name data_type
```

حيث ADD لإضافة عمود جديد متبوعة باسم العمود ونوع البيانات التي ستخزن به و MODIFY لتعديل مواصفات حقل موجود متبوعة باسم الحقل ونوع البيانات الجديدة .

مثال :

لتعديل طول invoice_no الموجود بجدول INVOICES الذي أنشأناه قبل قليل ليصبح طوله ١٥ بدلا من ٨ استخدم الأمر التالي :

```
> ALTER TABLE INVOICES
> MODIFY INVOICE_NO VARCHAR(15);
```

لاحظ أنه لا يمكن تغيير طول الحقل إلى أقل من طول أطول قيمة أو معلومة موجودة في الجدول حتى لا تفقد البيانات الموجودة .

ولإضافة حقل التاريخ إلى الجدول السابق استخدم المثال التالي :

```
> ALTER TABLE INVOICES
> ADD INV_DATE date;
```

ولحذف حقل استخدم Drop Field كما يلي :

```
>Alter table orders
>Drop order_date;
records: 13   duplication: 0   Warnings: 0
```

حذف جدول

قد تحتاج لحذف جدول موجود في قاعدة البيانات . ومن الأمثلة التي تحتاج فيها لحذف الجدول عندما تنشئ جدولا بغرض الاختبار فعندما تنتهي من الجدول واختباره يمكنك حذف هذا الجدول .لاحظ أن حذف الجدول يحذف كل ما يتعلق به مثل الفهارس والبيانات فكن حذرا عند استخدام هذا الأمر .

يستخدم أمر DROP TABLE لحذف جدول موجود والشكل العام لهذا الأمر كما يلي :

```
DROP TABLE table_name
```

ولحذف جدول orders استخدم الأمر التالي :

```
Drop table orders;
Query OK, 0 rows affected (0.02 sec)
```

حذف قاعدة البيانات

يمكن حذف قاعدة البيانات باستخدام أمر DROP DATABASE بنفس طريقة حذف جدول البيانات والشكل العام لهذا الأمر هو :

DROP DATABASE database_name

لحذف قاعدة البيانات **stockcontrol** استخدم الأمر التالي :

Drop database stockcontrol;
Query OK, 2 rows affected (0.06 sec)

تكوين جدول جديد من جدول موجود

أحيانا يكون من المفيد تكوين جدول مشابه للجدول الموجود . في هذه الحالة استخدم أمر **CREATE TABLE** كما يلي :

```
CREATE TABLE new_table (field1,field2,field3)
AS ( SELECT field1,field2,field3
FROM old_table < WHERE >
```

والمثال التالي يوضح ذلك :

نحتاج أحيانا لانشاء جدول مماثل لجدول موجود من حيث الحقول والخصائص ، هب أن لديك جدول للاعبي كرة القدم وتريد انشاء جدول آخر للاعبين أكثر من ٢١ سنة ، الواقع أنك تريد انشاء جدول متطابق مع الجدول الأساسي فيما عدا أن البيانات في الجدول الجديد لن تشتمل الا على بعض اللاعبين ممن هم تحت سن ٢١ سنة .

الجدول الأساسي بياناته كالتالي :

Select * from first_team;
localhost / football / first_team (14 Records)

F_Name	L_Name	Club	Age
Atef	Obaid	Tarsana	33
Wael	Riyad	Ahly	22
Ahmed	Hassan	Ahly	19
Nader	Sayed	Masry	35
Essam	Hadary	Ahly	30

Ibraheem	Hassan	Masry	34
Hazem	Emam	Zamalek	29
Khaled	Ghandoor	Zamalek	30
Mohamed	Sabry	Zamalek	18
Alaa	Hassan	Ismaely	17
Hany	Gamal	Mansoura	20
Ahmed	Yakoub	Ahly	27
Mohmmad	Hamed	Ahly	21
Farook	Hosny	Assuit	17

generated 1/24/2005 7:09:40 PM by MySQL-Front 1.22

سنقوم الآن بإنشاء جدول للناشئين أقل من ٢١ سنة من الجدول الرئيسي وذلك باستخدام MySQL Query التالية :

```
create table youth_team
as ( select * from first_team
where age < 21 );
```

ولإظهار محتويات الجدول الجديد

```
select * from youth_team;
```

localhost / football / youth_team (5 Records)

F_Name	L_Name	Club	Age
Ahmed	Hassan	Ahly	19
Mohamed	Sabry	Zamalek	18
Alaa	Hassan	Ismaely	17
Hany	Gamal	Mansoura	20
Farook	Hosny	Assuit	17

generated 1/24/2005 7:19:45 PM by MySQL-Front 1.22

نقوم بإنشاء جدول آخر للكبار أكبر من ٢٠ سنة باستخدام الجدول الرئيسي كالتالي :

```
create table stars_team
 as ( select * from first_team where age > 20);
select * from stars_team;
```

localhost / football / stars_team (9 Records)

F_Name	L_Name	Club	Age
Atef	Obaid	Tarsana	33
Wael	Riyad	Ahly	22
Nader	Sayed	Masry	35
Essam	Hadary	Ahly	30
Ibraheem	Hassan	Masry	34
Hazem	Emam	Zamalek	29
Khaled	Ghandoor	Zamalek	30
Ahmed	Yakoub	Ahly	27
Mohmmad	Hamed	Ahly	21

generated 1/24/2005 7:25:01 PM by MySQL-Front 1.22