

SQL Introduction

Steven Jones, Genome Sciences Centre

History of Relational Databases

- Relational databases Initially described by E.F. Codd from IBM in June, 1970.
- IBM developed a language to access the database called “Structured English Query Language” – SEQUEL – later called SQL.
- First commercial product developed by a small startup company called Oracle

Salient Features of a Relational Database

- Data is held in tables.

Relational Databases

- Oracle
- DB2 (IBM)
- PostgreSQL
- MySQL

MySQL

- General Purpose Open Source RDBMS
- Optimized for speed, reliability and high-load
- Well supported and documented
- Easy to install, configure and manage
- Comes with many Linux distributions
- Widely used >10,000 server downloads per day
- Low cost of ownership

High-Profile users of MySQL

- NASA
- Yahoo finance
- Slashdot.org, Freshmeat.net, Linux.com
- Nortel, Ericsson, Alcatel, Telia, Nokia
- Motorola
- Compaq
- Ensembl

Accessible through many Programmable interfaces

- ADA
 - C
 - C++
 - LISP
 - Delphi
 - Dylan
 - Guile
- JDBC
 - MatLab
 - ODBC
 - Perl
 - PHP
 - Pike
 - Python
 - etc etc*

A typical Table

```
mysql> select * From Scientist ;
```

Scientist_ID	Room	Phone	Last_Name	First_Name	email
1	H201	2994561	Harries	John	jh1@hotmail.com
2	C405	2994562	Carter	Ross	rcarter@telus.net
3	C401	NULL	Juriloff	Nancy	nj34@msn.com
4	F301	2667567	Hawkins	Joyce	jhawkins@irn.ca

```
4 rows in set (0.00 sec)
```

Data Types

- Choosing the type of Data stored in each column is key to the utility and efficiency of your database
- Various data-types exist to store integer numbers, strings, time, date, binary objects (BLOBS).

Numeric Column Types

**TINYINT
SMALLINT
MEDIUM INT
INT
BIG INT
FLOAT
DOUBLE
DECIMAL**

String Column Types

**CHAR
VARCHAR
TINYBLOB
BLOB
MEDIUM BLOB
LONG BLOB
TINYTEXT
TEXT
MEDIUMTEXT
LONGTEXT**

Date and Time

**DATE
TIME
DATETIME
TIMESTAMP
YEAR**

Different Data-Types

Numeric Column Types

TINYINT

SMALLINT

MEDIUM INT

INT

BIG INT

FLOAT

DOUBLE

DECIMAL

String Column Types

CHAR

VARCHAR

TINYBLOB

BLOB

MEDIUM BLOB

LONG BLOB

TINYTEXT

TEXT

MEDIUMTEXT

LONGTEXT

Date and Time

DATE

TIME

DATETIME

TIMESTAMP

YEAR

Querying the data

```
mysql> select * FROM Scientist WHERE Last_Name="Harries";
```

Scientist_ID	Room	Phone	Last_Name	First_Name	email
1	H201	2994561	Harries	John	jh1@hotmail.com

```
1 row in set (0.00 sec)
```

Inserting Data

```
mysql> INSERT INTO Scientist (Room,Phone,Last_Name,First_Name) VALUES  
( 'H234','2334576','Michaels','James');
```

```
Query OK, 1 row affected (0.06 sec)
```

```
mysql> select * FROM Scientist
```

```
+-----+-----+-----+-----+-----+-----+  
| Scientist_ID | Room | Phone   | Last_Name | First_Name | email           |  
+-----+-----+-----+-----+-----+-----+  
|           1 | H201 | 2994561 | Harries   | John       | jh1@hotmail.com |  
|           2 | C405 | 2994562 | Carter    | Ross      | rcarter@telus.net |  
|           3 | C401 |      NULL | Juriloff  | Nancy     | nj34@msn.com     |  
|           4 | F301 | 2667567 | Hawkins   | Joyce     | jhawkins@irn.ca  |  
|           5 | H234 | 2334576 | Michaels  | James     |                  |  
+-----+-----+-----+-----+-----+-----+
```

```
5 rows in set (0.00 sec)
```

Editing Rows

```
mysql> UPDATE Scientist SET email="jmtelus.net" WHERE Scientist_ID='5';  
Query OK, 1 row affected (0.06 sec)  
ws matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * FROM Scientist ;
```

Scientist_ID	Room	Phone	Last_Name	First_Name	email
1	H201	2994561	Harries	John	jh1@hotmail.com
2	C405	2994562	Carter	Ross	rcarter@telus.net
3	C401	NULL	Juriloff	Nancy	nj34@msn.com
4	F301	2667567	Hawkins	Joyce	jhawkins@irn.ca
5	H234	2334576	Michaels	James	jmtelus.net

```
5 rows in set (0.00 sec)
```

```
mysql>
```

Exploring a Database

```
mysql> SHOW TABLES ;
```

```
+-----+
| Tables_in_clone_db |
+-----+
| Clone              |
| Scientist          |
+-----+
```

```
2 rows in set (0.00 sec)
```

```
mysql> Describe Clone;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type                | Null | Key | Default        | Extra          |
+-----+-----+-----+-----+-----+-----+
| Clone_ID      | int(10) unsigned    |      | PRI | NULL           | auto_increment |
| Requestor_ID  | int(11)             |      |     | 0              |                |
| Date_Received | date                |      |     | 0000-00-00     |                |
| Date_Completed | date                | YES  |     | NULL           |                |
+-----+-----+-----+-----+-----+-----+

```

```
4 rows in set (0.01 sec)
```

Creating Tables

Creating the “Clone” Table

```
CREATE TABLE Clone (  
    Clone_ID int(10) unsigned NOT NULL auto_increment PRIMARY KEY,  
    Requestor_ID int(11) NOT NULL  
    Date_Received date NOT NULL,  
    Date_Completed date,  
  
)
```

NOT NULL – This field cannot be empty

Auto_increment – Automatically Generate a new clone_ID for each row added

PRIMARY KEY – Creates an index for the table based on this value

The Clone Table

```
mysql> select * FROM Clone ;
```

Clone_ID	Requestor_ID	Date_Received	Date_Completed
1	2	2002-08-01	2002-08-15
2	2	2002-08-02	2002-08-12
3	3	2002-08-19	NULL
4	4	2002-08-20	NULL
5	4	2002-08-20	2002-08-26
6	4	2002-08-22	NULL

```
6 rows in set (0.00 sec)
```

Goals

- Calculate the turnaround time on completed clones?
- What is the outstanding time on incomplete clones?
- Who has outstanding clones?
- Who has used the service the most?

Calculating turnaround time

```
mysql> SELECT Clone_ID, Date_Received, (TO_DAYS(Date_Completed))-  
TO_DAYS(Date_Received) AS Turnaround FROM Clone WHERE Date_Completed IS NOT NULL ;
```

```
+-----+-----+-----+  
| Clone_ID | Date_Received | Turnaround |  
+-----+-----+-----+  
|         1 | 2002-08-01   |          14 |  
|         2 | 2002-08-02   |          10 |  
|         5 | 2002-08-20   |           6 |  
+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

Note the `TO_DAYS` function turns a date value into the number of DAYS since 1st of Jan 0001.

Determining Outstanding Clones

```
mysql> SELECT Clone_ID, Date_Received, (TO_DAYS(CURRENT_DATE))-  
TO_DAYS(Date_Received) AS Turnaround FROM Clone WHERE Date_Completed IS NULL ;
```

```
+-----+-----+-----+  
| Clone_ID | Date_Received | Turnaround |  
+-----+-----+-----+  
|          3 | 2002-08-19   |          57 |  
|          4 | 2002-08-20   |          56 |  
|          6 | 2002-08-22   |          54 |  
+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

Note `CURRENT_DATE` simply returns today's date. You can also use this if you are entering data.

Who is waiting for clones?

```
mysql> SELECT Scientist.Last_Name, COUNT(Clone.Clone_ID) AS Clones FROM
Scientist, Clone WHERE Scientist.Scientist_ID = Clone.Requestor_ID AND
Clone.Date_completed IS NULL GROUP BY Scientist.Last_name ;
```

```
+-----+-----+
| Last_Name | Clones |
+-----+-----+
| Hawkins   |      2 |
| Juriloff  |      1 |
+-----+-----+
2 rows in set (0.00 sec)
```

Note we are now extracting data from multiple tables, therefore we need to be explicit in which fields are taken from each table. We do this through the table.field nomenclature. We “join” the tables through the fact that ID of the scientists is common between the tables. GROUP BY tells MySQL How to cluster the values after counting them.

Who has requested the most Clones

```
mysql> SELECT Scientist.Last_Name, COUNT(Clone.Clone_ID) AS Clones FROM
Scientist, Clone WHERE Scientist.Scientist_ID = Clone.Requestor_ID GROUP BY
Scientist.Last_name ORDER BY Clones DESC ;
```

```
+-----+-----+
| Last_Name | Clones |
+-----+-----+
| Hawkins   |      3 |
| Carter    |      2 |
| Juriloff  |      1 |
+-----+-----+
```

```
3 rows in set (0.00 sec)
```

Easy Administration through myPHPAdmin

Home

clone_db (2)

clone_db

- Clone
- Scientist

Database clone_db running on nmemosyne.bcgsc.bc.ca

Table	Action	Records	Type	Size
<input type="checkbox"/> Clone	Browse Select Insert Properties Drop Empty	6	MyISAM	2.1 KB
<input type="checkbox"/> Scientist	Browse Select Insert Properties Drop Empty	5	MyISAM	2.3 KB
2 table(s) Sum		11	--	4.4 KB

↑ [Check All](#) / [Uncheck All](#) With selected: ▾

- [Print view](#)
- Run SQL query/queries on database clone_db [[Documentation](#)]:
 Show this query here again
 Or Location of the textfile:
- [Query by Example](#)
- View dump (schema) of database

Clone	<input checked="" type="radio"/> Structure only
Scientist	<input type="radio"/> Structure and data
	<input type="radio"/> Data only

[Select All](#) / [Unselect All](#)
 - Add 'drop table'
 - Complete inserts
 - Extended inserts
 - Enclose table and field names with backquotes
 - Save as file ("zipped" "gzipped")

Database clone_db - table Scientist running on nmemosyne.bcgsc.bc.ca

Showing rows 0 - 4 (5 total)

SQL-query : [\[Edit\]](#)
 SELECT * FROM `Scientist` LIMIT 0, 30

Show: rows starting from
 in mode and repeat headers after cells

		Scientist_ID	Room	Phone	Last_Name	First_Name	email
Edit	Delete	1	H201	2994561	Harries	John	jh1@hotmail.com
Edit	Delete	2	C405	2994562	Carter	Ross	rcarter@telus.net
Edit	Delete	3	C401	NULL	Juriloff	Nancy	nj34@msn.com
Edit	Delete	4	F301	2667567	Hawkins	Joyce	jhawkins@irn.ca
Edit	Delete	5	H234	2334576	Michaels	James	jmtelus.net

Show: rows starting from
 in mode and repeat headers after cells

[Insert new row](#)

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

Recommended Reading

