References


MySQL, DuBois, New Riders, 2000

PostgreSQL Developer’s Handbook, Geschwinde, Schönig, Sams, 2002

PostgreSQL Interactive Documentation [http://www.postgresql.org/idocs/](http://www.postgresql.org/idocs/)


Andrew Scherpier’s CS580 Lecture notes [http://www.eli.sdsu.edu/courses/spring97/cs596/notes/databases/databases.html](http://www.eli.sdsu.edu/courses/spring97/cs596/notes/databases/databases.html)
Database

Most servers will use some sort of database.

Jargon

2-Tier

Client

Server

3-Tier

Client

Server

Database
What is a database?

A database holds information and provides for a mechanism to access this information.

Examples of some common (electronic) databases:

• Unix password file
• IRS records system
• Rolodex(TM)
• Computer file system
• Library (object files)
• Student grades
• Telephone directory
Jargon

Sometimes **database** means a program for managing data

    Oracle Corporation is a database company.
    MS Access is database.

Sometimes **database** means a collection of data

    I keep a database of my CD collection on 3 by 5 cards

Sometimes **database** means a set of tables, indexes, and views

    My program needs to connect to the Airline Reservation database, which uses Oracle
Some Reasons for Using a Database

Persistence of data

Sharing of data between programs

Handle concurrent requests for data access

Transactions that can be rolled back

Report generation
Types of Databases

Relational

Data is stored in tables

Object-Oriented

Tables can be subclassed

Programmer can define methods on tables

Object

Objects are stored in the database
Relational, Object-Oriented Databases and SQL

Database consists of a number of tables

Table is a collection of records

Each Column of data has a type

<table>
<thead>
<tr>
<th>firstname</th>
<th>lastname</th>
<th>phone</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Smith</td>
<td>555-9876</td>
<td>2000</td>
</tr>
<tr>
<td>Ben</td>
<td>Oker</td>
<td>555-1212</td>
<td>9500</td>
</tr>
<tr>
<td>Mary</td>
<td>Jones</td>
<td>555-3412</td>
<td>9900</td>
</tr>
</tbody>
</table>

Use Structured query language (SQL) to access data
Some Available Databases

• Oracle
• DB2
• SQL Server
• Access
• Informix
• Ingres
• InterBase
• Sybase
• FileMaker Pro
• FoxPro
• Paradox
• dBase

Open Source Databases
• MySQL
• PostgresSQL
SQL History

Structured query language (SQL)

Dr. E. F. Codd develops relational database model
   Early 1970's

IBM System R relational database
   Mid 1970's
   Contained the original SQL language

First commercial database - Oracle 1979

SQL was aimed at:
   Accountants
   Business people

SQL89
   Not well followed
   ANSI X3.135-1989

SQL92
   First commonly followed standard
   ANSI X3.135-1992
   SQL2

ISO/IEC 9075-1 through 5
   New SQL standard
MySQL & PostgreSQL

Open source databases

http://www.mysql.com/

http://www.postgresql.org/

Above site have free downloads and documentation
MySQL Connecting to the Database

Can be done with:

• Mysql command line tool - mysql
• GUI clients
• Program

GUI Clients

If done well are very useful

There are many of these

MySql web site lists 10 pages of them, see:

http://www.mysql.com/portal/software/graphing/index.html

I use DbVisualizer,

DbVisualizer if Java based so runs on may platforms

http://www.dbvis.com/products/dbvis/
Names

Databases, tables columns & indexes have names

Legal Characters

Alphanumeric characters

\'_\'

'\$

Names can start with:
Letter
Underscore
Letter with diacritical marks and some non-latin letters

Name length

• 63 characters – default in PostgreSQL
• 64 characters - MySQL

Names are not case sensitive
Data Types

• Numeric Values
  Integer - decimal or hex
  Floating-point - scientific & 12.1234

• String Values
  ‘this is a string’ PostgreSQL
  ‘this is a string’ “this is also a string” MySQL

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘</td>
<td>Single quote</td>
</tr>
<tr>
<td>\b</td>
<td>Backspace</td>
</tr>
<tr>
<td>\n</td>
<td>Newline</td>
</tr>
<tr>
<td>\r</td>
<td>Tab</td>
</tr>
<tr>
<td>\</td>
<td>Backslash</td>
</tr>
<tr>
<td>\xxxx</td>
<td>Character were x is the octal of ASCII code (PostgreSQL)</td>
</tr>
</tbody>
</table>

Including a quote character in a string
Double quote the character
  'Don"t do it'

Escape the quote character with a backslash
  'Don\'t do it'
### Comments

-- this is a comment in MySQL and PostgreSQL

/* this is also a comment in MySQL and PostgreSQL */

# this is a comment in MySQL
### Numeric Data Types

<table>
<thead>
<tr>
<th>Type name</th>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>smallint</td>
<td>Fixed-precision</td>
<td>-32768 to +32767</td>
</tr>
<tr>
<td>integer</td>
<td>Usual choice for fixed-precision</td>
<td>-2147483648 to +2147483647</td>
</tr>
<tr>
<td>bigint</td>
<td>Very large range fixed-precision</td>
<td>-9223372036854775808 to 9223372036854775807</td>
</tr>
<tr>
<td>decimal</td>
<td>user-specified precision, exact</td>
<td>no limit</td>
</tr>
<tr>
<td>numeric</td>
<td>user-specified precision, exact</td>
<td>no limit</td>
</tr>
<tr>
<td>real</td>
<td>variable-precision, inexact</td>
<td>6 decimal digits precision</td>
</tr>
<tr>
<td>double precision</td>
<td>variable-precision, inexact</td>
<td>15 decimal digits precision</td>
</tr>
<tr>
<td>serial</td>
<td>autoincrementing integer</td>
<td>1 to 2147483647</td>
</tr>
</tbody>
</table>

Numeric(10, 2) defines a number with maximum of 10 digits with 2 of the 10 to the right of the decimal point

12345678.91

decimal and numeric are different names for the same type
### String Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char(n)</td>
<td>Fixed-length blank padded</td>
</tr>
<tr>
<td>varchar(n)</td>
<td>Variable-length with limit</td>
</tr>
<tr>
<td>text</td>
<td>Variable unlimited length</td>
</tr>
<tr>
<td>bytea (PostgreSQL)</td>
<td>Variable (not specifically limited) length binary string</td>
</tr>
<tr>
<td>blob (MySQL)</td>
<td>Variable (not specifically limited) length binary string</td>
</tr>
</tbody>
</table>

CHAR & VARCHAR are the most common string types

CHAR is fixed-width

Shorter strings are padded

TEXT can be any size

PostgreSQL limits a string to 1GB in storage space

MySQL limits CHAR and VARCHAR to 255 characters
## Date & Time Types - PostgreSQL

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp [(p)] without time zone</td>
<td>both date and time</td>
</tr>
<tr>
<td>timestamp [(p)] with time zone</td>
<td>both date and time</td>
</tr>
<tr>
<td>interval [(p)]</td>
<td>for time intervals</td>
</tr>
<tr>
<td>date</td>
<td>dates only</td>
</tr>
<tr>
<td>time [(p)] without time zone</td>
<td>times of day only</td>
</tr>
<tr>
<td>time [(p)] with time zone</td>
<td>times of day only</td>
</tr>
</tbody>
</table>

(p) indicates optional number of fractional digits retained in the seconds field
## Date Formats - PostgreSQL

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 8, 1999</td>
<td>Unambiguous</td>
</tr>
<tr>
<td>1999-01-08</td>
<td>ISO-8601 format, preferred</td>
</tr>
<tr>
<td>1/8/1999</td>
<td>U.S.; read as August 1 in European mode</td>
</tr>
<tr>
<td>8/1/1999</td>
<td>European; read as August 1 in U.S. mode</td>
</tr>
<tr>
<td>1/18/1999</td>
<td>U.S.; read as January 18 in any mode</td>
</tr>
<tr>
<td>19990108</td>
<td>ISO-8601 year, month, day</td>
</tr>
<tr>
<td>990108</td>
<td>ISO-8601 year, month, day</td>
</tr>
<tr>
<td>1999.008</td>
<td>Year and day of year</td>
</tr>
<tr>
<td>99008</td>
<td>Year and day of year</td>
</tr>
<tr>
<td>J2451187</td>
<td>Julian day</td>
</tr>
<tr>
<td>January 8, 99 BC</td>
<td>Year 99 before the Common Era</td>
</tr>
</tbody>
</table>

### Setting the Date Format

```sql
SET DateStyle TO 'US'
SET DateStyle TO 'NonEuropean'

Sets date format to  month day year

SET DateStyle TO 'European'

Sets date format to  day month year
```

Default is ISO style
Dates – MySQL

DATETIME – ‘YYYY-MM-DD HH:MM:SS’ format

DATE – ‘YYYY-MM-DD’ format

TIMESTAMP
• Changed in MySQL 4.1
• Basically now is same as DATETIME
## Common SQL Statements

<table>
<thead>
<tr>
<th>SQL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>Retrieves data from table(s)</td>
</tr>
<tr>
<td>INSERT</td>
<td>Adds row(s) to a table</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Changes field(s) in record(s)</td>
</tr>
<tr>
<td>DELETE</td>
<td>Removes row(s) from a table</td>
</tr>
<tr>
<td>CREATE TABLE</td>
<td>Define a table and its columns(fields)</td>
</tr>
<tr>
<td>DROP TABLE</td>
<td>Deletes a table</td>
</tr>
<tr>
<td>ALTER TABLE</td>
<td>Adds a new column, add/drop primary key</td>
</tr>
<tr>
<td>CREATE INDEX</td>
<td>Create an index</td>
</tr>
<tr>
<td>DROP INDEX</td>
<td>Deletes an index</td>
</tr>
<tr>
<td>CREATE VIEW</td>
<td>Define a logical table from other table(s)/view(s)</td>
</tr>
<tr>
<td>DROP VIEW</td>
<td>Deletes a view</td>
</tr>
</tbody>
</table>

SQL is not case sensitive
Examples That Follow

Will use mysql command line tool

Used the command

```
mysql -h host -u user -p
```

to connect to the database, where host and user are given the correct value

On rohan the full name of command is:

```
/opt/local/mysql/bin/mysql
```

Some examples will also show postgresQL text client
CREATE DATABASE

General Form

CREATE DATABASE [IF NOT EXISTS] db_name
   [create_specification [, create_specification] ...]

create_specification:
   [DEFAULT] CHARACTER SET charset_name
   | [DEFAULT] COLLATE collation_name

Example

mysql> create database lectureExamples;
Query OK, 1 row affected (0.00 sec)

PosgreSQL

Al 15->psql -h bismarck.sdsu.edu cs580whitney cs580whitney
Password:
Welcome to psql 7.4, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
   \h for help with SQL commands
   \? for help on internal slash commands
   \g or terminate with semicolon to execute query
   \q to quit

cs580whitney=> create database lectureExamples;
ERROR: permission denied to create database

cs580whitney=>

Student accounts do not have authority to create new databases
**USE**

Sets a default database for subsequent queries

**General Form**

```
USE db_name
```

**Example**

```
mysql> use lectureExamples;
Database changed
```
CREATE table

Creates a table.

General Form

CREATE TABLE table_name (
    col_name    col_type [ NOT NULL | PRIMARY KEY]
    [, col_name col_type [ NOT NULL | PRIMARY KEY]]*
)

Example

mysql> CREATE TABLE students
    ( firstname  CHAR(20) NOT NULL, lastname  CHAR(20),
      phone      CHAR(10), code      INTEGER
    );

mysql> CREATE TABLE codes
    ( code       INTEGER, name       CHAR(20)
    );
**PostgreSQL Example**

```sql
CREATE TABLE students (
    firstname CHAR(20) NOT NULL,
    lastname CHAR(20),
    phone CHAR(10),
    code INTEGER
);
```

```sql
CREATE TABLE

select * from students;
```

<table>
<thead>
<tr>
<th>firstname</th>
<th>lastname</th>
<th>phone</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----------</td>
<td>----------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>(0 rows)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

4/14/05
Select

Gets data from one or more tables

General Form

```
SELECT [STRAIGHT_JOIN]
  [SQL_SMALL_RESULT] [SQL_BIG_RESULT]
  [SQL_BUFFER_RESULT] [SQL_CACHE | SQL_NO_CACHE]
  [SQL_CALC_FOUND_ROWS] [HIGH_PRIORITY]
  [DISTINCT | DISTINCTROW | ALL]
  select_expression,...
  [INTO {OUTFILE | DUMPFILE} 'file_name' export_options]
  [FROM table_references
    [WHERE where_definition]
    [GROUP BY {unsigned_integer | col_name | formula} [ASC | DESC],...
    [WITH ROLLUP]]
    [HAVING where_definition]
    [ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC],...]
  [LIMIT [offset,] row_count | row_count OFFSET offset]
  [PROCEDURE procedure_name(argument_list)]
  [FOR UPDATE | LOCK IN SHARE MODE]]
```

Example

```
mysql> SELECT * FROM students;
Empty set (0.00 sec)
```
Insert

Add data to a table

General Form

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
    [INTO] tbl_name [(col_name,...)]
VALUES ((expression | DEFAULT),...),(...),...
[ ON DUPLICATE KEY UPDATE col_name=expression, ... ]
```

Examples

```
mysql> INSERT
    INTO students (firstname, lastname, phone, code)
    VALUES ('Roger', 'Whitney', '594-3535', 2000 );

mysql> INSERT
    INTO codes (code, name)
    VALUES (2000, 'marginal' );

mysql> SELECT * FROM students;

+-----------+----------+----------+------+
| firstname | lastname | phone    | code |
+-----------+----------+----------+------+
| Roger     | Whitney  | 594-3535 | 2000 |
+-----------+----------+----------+------+
1 row in set (0.01 sec)
```
More Select Examples

```sql
mysql> SELECT firstname, phone FROM students;
+-----------+----------+
| firstname | phone    |
+-----------+----------+
| Roger     | 594-3535 |
+-----------+----------+
1 row in set (0.00 sec)

mysql> SELECT lastname, name
    FROM students, codes
    WHERE students.code = codes.code;
+----------+----------+
| lastname | name     |
+----------+----------+
| Whitney  | marginal |
+----------+----------+
1 row in set (0.00 sec)

mysql> SELECT students.lastname, codes.name
    FROM students, codes
    WHERE students.code = codes.code;
+----------+----------+
| lastname | name     |
+----------+----------+
| Whitney  | marginal |
+----------+----------+
1 row in set (0.00 sec)
```
Update

Modify existing data in a database

**General Form**

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name [, tbl_name ...]
    SET col_name1=expr1 [, col_name2=expr2 ...]
    [WHERE where_definition]
```

**Example**

```
mysql> UPDATE students
    SET firstname='Sam'
    WHERE lastname='Whitney';

Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```
Alter Table

Modify the table structure – add/delete columns, change column type

General Form

```
ALTER [IGNORE] TABLE tbl_name alter_specification [, alter_specification] ...
```

```
alter_specification:
   ADD [COLUMN] create_definition [FIRST | AFTER column_name ]
| ADD [COLUMN] (create_definition, create_definition,...)
| ADD INDEX [index_name] [index_type] (index_col_name,...)
| ADD [CONSTRAINT [symbol]] PRIMARY KEY [index_type]
   (index_col_name,...)
| ADD [CONSTRAINT [symbol]] UNIQUE [index_name] [index_type]
   (index_col_name,...)
| ADD [CONSTRAINT [symbol]] FOREIGN KEY [index_name]
   (index_col_name,...) [reference_definition]
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] old_col_name create_definition
  [FIRST | AFTER column_name]
| MODIFY [COLUMN] create_definition [FIRST | AFTER column_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP INDEX index_name
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO] new_tbl_name
| ORDER BY col
| CHARACTER SET character_set_name [COLLATE collation_name]
| table_options
```
Example

mysql> ALTER TABLE students ADD column foo CHAR(40);

Query OK, 1 row affected (0.03 sec)
Records: 1  Duplicates: 0  Warnings: 0
Drop Table

Remove a table from the database

General Form

```
DROP [TEMPORARY] TABLE [IF EXISTS]
tbl_name [, tbl_name,...] [RESTRICT | CASCADE]
```

Example

```
mysql> DROP TABLE students;
Query OK, 0 rows affected (0.01 sec)
```
Drop Database

Removes a database and all its tables

General Form

DROP DATABASE [IF EXISTS] db_name

Example

mysql> DROP DATABASE lectureexamples;
Query OK, 0 rows affected (0.00 sec)