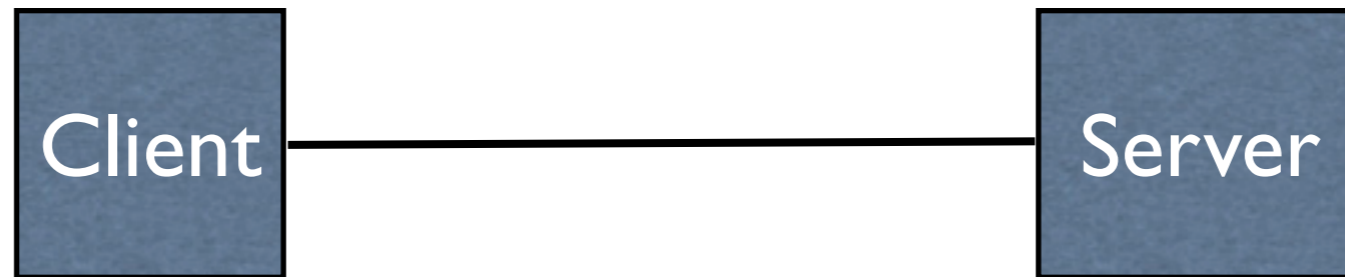


CS 580 Client-Server Programming  
Spring Semester, 2006  
Doc 9 SQL Intro  
Feb 21, 2006

Copyright ©, All rights reserved. 2006 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

## Jargon

### 2-Tier



### 3-Tier



## More 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

firstname	lastname	phone	code
John	Smith	555-9876	2000
Ben	Oker	555-1212	9500
Mary	Jones	555-3412	9900

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

PostgreSQL

# SQL History

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

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

I use DbVisualizer, & CocoaMySQL

DbVisualizer is Java based so runs on many platforms

<http://www.dbvis.com/products/dbvis/>

# SQL Syntax

## 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

Sequence	Meaning
\'	Single quote
\b	Backspace
\n	Newline
\r	Tab
\\	Backslash
\xxxx	Character where xxxx is the octal of ASCII code (PostgreSQL)

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

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

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

Type	Description
char(n)	Fixed-length blank padded
varchar(n)	Variable-length with limit
text	Variable unlimited length
bytea (PostgreSQL)	Variable (not specifically limited) length binary string
blob (MySQL)	Variable (not specifically limited) length binary string

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

Type	Description
timestamp [(p)] without time zone	both date and time
timestamp [(p)] [ with time zone ]	both date and time
interval [(p)]	for time intervals
date	dates only
time [(p)] [ without time zone ]	times of day only
time [(p)] with time zone	times of day only

(p) indicates optional number of fractional digits retained in the seconds field



## Date Formats - PostgreSQL

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

## Setting Date Formats - PostgreSQL

```
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

SELECT	Retrieves data from table(s)
INSERT	Adds row(s) to a table
UPDATE	Changes field(s) in record(s)
DELETE	Removes row(s) from a table Data Definition
CREATE TABLE	Define a table and its columns(fields)
DROP TABLE	Deletes a table
ALTER TABLE	Adds a new column, add/drop primary key
CREATE INDEX	Create an index
DROP INDEX	Deletes an index
CREATE VIEW	Define a logical table from other table(s)/view(s)
DROP VIEW	Deletes a view

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 postgresSQL 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 Example

```
AI 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

### 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

```
cs580whitney=> CREATE TABLE students
cs580whitney-> (
cs580whitney(> firstname CHAR(20) NOT NULL,
cs580whitney(> lastname CHAR(20),
cs580whitney(> phone CHAR(10),
cs580whitney(> code INTEGER
cs580whitney(> );
CREATE TABLE
```

```
cs580whitney=> select * from students;
firstname | lastname | phone | code
-----+-----+-----+-----
(0 rows)
```

## 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

```
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)
```

## More Select Examples

```
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