《C/C++学习指南》 MySQL篇

1.1 概述

理解以下三个概念：

(1)数据 Data

(2)数据库 Database

(3)数据库服务器 Database Server

数据

本课程所要处理的数据，指是表格状数据。

行Row, 列Column

行：第一行称为一条记录 ( Record)

列：每一列称为一个字段 (Field)

学号 姓名 籍贯 手机号码

20150001 王 河北 18600012340

20150002 李 海南 15000010102

20150003 杨 湖南 15210020093

20150004 姜 安徽 18219029898

20150005 周 福建 13109092878

共5条记录，每条记录有4个字段

数据库 ，即“数据仓库”，存放数据的地方

例如，我把可以前面的表中数据存储在一个文件 里，则称该文件为一个数据库。

fopen(“mydata.db”, …)

fwrite(…)

fwrite(…)

则mydata.db是一个数据库

数据库服务器

一个可以操作数据库，并提供相关服务的软件。

例如，我们写一个软件SimpleDB，它有以下功能：

（1）增加一条记录，存到mydata.db

（2）删除一条记录

（3）修改现有记录的值

（4）查询某个记录的值

则，称此软件为一个数据库服务器。它提供了 “数据服务”

数据库服务器是一款软件，它提供了 数据相关的服务。例如，MySQL

有时，我们把数据库服务器简称为“数据库”

一般地，我们并不关心它内部是如何存储和

管理数据的。我们只关心如何使用它。

使用数据库

数据库有两类使用者：

（1）管理员 DB Administrator (简称DBA)

负责日常的维护工作：启动停止、备份恢复、 用户管理，等等。

（2）开发人员 Developer

在应用程序中，使用数据库服务。这是本课 程的重点。

小节

引入了数据库的概念

1.2 MySQL的安装与配置

几种常见的数据库

常见关系型数据库 (RDS)

MySQL ,  oracle,  DB2,  SQL Server  , Sybase …

其他：

sqlite: 本地文件数据库

hbase: Non‐SQL数据库（非关系型数据库）

关于MySQL

1. C/S模式 （客户端/服务器）

客户端与服务器通过网络连接交互数据

(当然，客户端和服务器可以在同一台电脑上)

2. Linux / Windows

MySQL在Linux和Windows上都有发行版，正式部 署时通常在Linux上。

本课程是在Windows安装MySQL

(课程的附录中将介绍Linux下的相关使用)

MySQL的安装

按照《MySQL安装使用说明(Windows) 》。。

在官网上下载这一篇教程。

1. 安装MySQL

2. 启动与停止

3. 安装后的配置

4. 使用SQLyog测试 (一款客户端软件)

测试MySQL

第1种测试方法： 使用客户端软件连接到MySQL

第2种测试方法： 管理端命令行

MySQL在安装的时候，随同安装一系列用于管理的命令行程序，如 mysqladmin

位置 ： C:\Program Files\MySQL\MySQL Server 5.5\bin

(已经加入到PATH环境变量中）

小节

讲解如何在Windows上安装MySQL服务器

1.3 创建数据库和表

添加和删除数据库

MySQL: 数据库服务器, 里面可以存放多个数 据库。

演示：

（1）在MySQL中添加一个数据库

（2）在MySQL中删除一个数据库

添加和删除表

一个库中可以存放多个表，表中可以有多个列

每个列的数据类型主要有4种：

（1）整数 tinyint, smallint, mediumint, int, bigint

（2）小数 float, double

（3）字符串 char, varchar

（4）长文本 text

（5）时间 date, time, datetime, timestamp

（6）二进制数据 blob

编辑表的格式

在新建表时，要设定其每一列的属性（类型、

索引、自增、默认值），统称为schema （大纲）

具体怎么设置，后面会继续讲解。

编辑表的内容

新增一行

删除一行

修改一行并保存

小节

学会了添加数据库和表

1.4 表的定义：列属性

列的属性

每个列有以下属性：

1. 列名，建议以英文命名

2. 数据类型： (整数、小数、字符串、日期等）

3. 长度:  (当为字符串类型，需指定长度)

4. 主键：

5. 非空：(是非允许空格）

6. 自增：

列的属性：列名

基本上可以像C/C++里的名字一样随意命名

第一种风格：

log\_text : 小写字母，以下划线连接各个单词

第二种风格：

logText : 第二个单词首字母大写

注：不要使用SQL的关键词作为列名字 ，如int,  text,  type,  limit , group等等

列的属性：数据类型

（1）整数 tinyint, smallint, mediumint, int, bigint

（2）小数 float, double

（3）字符串 char, varchar

（4）长文本 text

（5）时间 date, time, datetime, timestamp

（6）二进制数据 blob (大块数据)

整数:

tinyint (1字节) ‐128 ~127

smallint (2字节)

mediumint (3字节)

int (4字节)

long (8字节)

小数： float, double

另外还有（bit, bool, numberic, decimal类型，请 自行研究）

列的属性：数据类型

字符串类型：

定长字符串 char

变长字符串 varchar

文本 text

对于上述三种字符串/文本类型，要同时限定 其最大长度。

char : 输入不超过N个字符。当输入小于N个 字符时，MySQL内部仍然是占用了N个字节。 如， name char (32) ，输入"shaofa"，在保存时仍然用了32个字节

varchar: 输入不超过N个字节。MySQL会按输 入的实际长度保存（节省空间）。 (variable char)

列的属性：数据类型

从直观上看：( 以下仅供初学者参考）

注：不同的服务器的实现略有不同。比如SQL

server里面的varchar可能没这么长。

类型 适用长度 实际应用                   优点

char  0‐255   较短长度 0‐32            查询快

varchar  0‐65532 中等长度 0‐1024  节省空间

text    长文本    超长文本                  存储超长文本

列的属性：非空

空值：NULL

列的属性：自增

对于整数字段，可以选择为自增。

当为自增时，该列的值无需指定，内部自动 按递增顺序填充。

列的属性：主键

下节课再说。

小节

在设计一张表时，如何设置列的属性。

1.5 表的定义：主键与索引

主键 Primary Key

主键(Primary Key)： 用于唯一地标识一条记录

比如，对于学生信息表

每个学生的ID是不同的，所以可以用ID来唯一标识一条记录。所以， 可以把id设为主键

思考：可以用name字段当成主键吗？

主键 Primary Key

主键的性质：

(1)该字段不允许重复

(2)该字段不允许为NULL

联合主键

有时，可以把多个字段联合起来，构成该表的主键。

例如，有多个课程:  Linux,  Qt,  Java ，每个课程下面有视频文件，每个文件的属性： 名称(filename), 大小(size)，时长(duration)

考虑：

(1) 同一课程下，文件名不会重复

(2) 不同的课程下，文件名可能重复，比如Linux下有 1.mp4，Qt下也有1.mp4

解决办法：

课程(category) + 文件名(filename) 联合在一 起来标识一条记录。 也就是说，(category + filename)是全局唯一的。

注：（有人认为）联合主键不被推荐。

推荐作法：为每条记录设立一个单独的ID

索引

索引：用于加快排序与查找

比如，要查找student表中所有年龄在1992‐1994年的学 生的记录。

在未加索引时，需要从头遍历每一条记录，比较其 birthday字段。

注意：所有的记录在存储是未排序的。查询速度慢。

有没有能提高效率的办法？？

索引：用该字段的值单独存储，并且是排序 后存储。

birthday            指向的记录号

1992‐1‐2          1090

1992‐1‐13         2087

1992‐2‐22        1287

1992‐2‐25         980

1992‐2‐25         23

1992‐3‐12         532

好处： 查找速度加快

坏处：

（1）建立和更新索引需要时间

（2）索引本身多占了一些存储空间

索引

普通索引与唯一索引：

唯一索引(unique)：该索引的字段不能重复。

普通索引：允许重复。

主键(primary)：其实也是一种唯一索引。

全文索引(fulltext): 对于大文本进行加速检索

演示：唯一索引的使用

索引

索引的优点是显然的。

注意其缺点：

（1）索引的建立和更新和耗费时间

每做一个增、删、改操作，都要同步地更新索引。

（2）索引字段应该尽量得短。越短小越好。

因此，当一个数据库读多写少时，可以考虑加多个索 引以加快查询速度。

相反，如果写操作频繁，那么加索引的时候要慎重。

小节

介绍了主键和索引的概念和使用。

主键必须掌握。

索引对初学者来说可以暂时跳过。

2.1 认识SQL语句

SQL: Structured Query Language

结构化查询语言，用于跟SQL服务器对话 ，属于解释执行的编程语言，其代码文件通常 称为 脚本，直接被解释器执行。

编译执行：C, C++, Java, C#, Basic …

解释执行：Perl , Shell

SQL脚本示例

下面的SQL脚本中含有2条SQL语句

/\* 进入example 数据库 \*/

USE EXAMPLE;

/\* 查询student 表 \*/

SELECT \* FROM student;

末尾要分号

关键词:

USE,SELECT,FROM

SQL脚本示例

SQL语法初步：

(1) 使用 /\*   \*/ 添加注释

(2) 每个SQL语句末尾加分号，中间可以换行

(3) 关键词建议大写。有的服务器可能并不区分 大小写。

例如，以下两种写法可能都被支持。

SELECT \* FROM example;

select \* from example;

SQL脚本示例

USE `example`;

DROP TABLE   IF EXISTS   `sample`;

CREATE TABLE `sample` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` char(16) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=gbk;

insert into `sample`(`id`,`name`)

values (1,'LiMing'),(2,'YangHua');

不用担心，SQLyog可以帮我们自动生成SQL脚本！

我们只需要认识它的意义，但并不需要强记。

在SQLyog可演示

1. 编辑SQL 脚本

“文件 | 新查询编辑器”

打开SQL Query窗口

2. 执行SQL 脚本

实际上，SQLyog是把SQL语句传给了服务器， 让服务器来执行，并把结果呈现给用户。

SQL解释器

MySQL服务器负责解释执行SQL脚本

当SQL代码传给MySQL后，MySQL解释执行， 并将结果返回给客户端。

SQLyog

客户端

MySQL

服务器

发送SQL脚本

返回执行结果

小节

1. 初步认识了SQL语言

2. 学会使用SQLyog编辑和执行SQL语句

SQL语言是 客户端 和 服务器 之间对话的语言

2.2 SQL: 数据库的操作

内容提要

1. 使用SQL语句来创建和删除数据库

2. 使用SQL语句来创建和删除表

3. 使用SQLyog来导出和导入数据库（重点）。

创建与删除数据库

创建：

CREATE DATABASE `example`;

删除:

DROP DATABASE `example`;

注：如果名称里没有特殊字符，可以不用` `

SQLyog: 查看历史记录

历史记录窗口：列出了用户的每一步操作

菜单 :  “工具 | 历史记录”

这里面全是标准的SQL语句，初学者可以参考。

可见，SQLyog的可视化操作，内部其实还是 使用了SQL语句。

表的创建和删除

创建

CREATE TABLE `simple` (

`id` int(11) NOT NULL,

`name` char(32) DEFAULT NULL,

PRIMARY KEY (`id`)

);

删除

DROP TABLE `simple`;

SQLyog: 导入与导出

使用SQLyog可以：

1. 将数据库里的内容导出（备份）

2. 导入数据（恢复）

导出的内容不仅包含表的结构，还可以包含 表的内容数据。

演示。。。观察导入的sql脚本内容。。。

小节

1. 大概了解CREATE和DROP的用法

2. 学会查看SQLyog的历史记录

3. 学会用SQLyog备份和恢复数据库

2.3 INSERT添加记录

内部提要

当数据库和表都已经创建好，下面就可以向 表中添加数据了。

(1)指定要操作的数据库

USE <数据库名>;

(2) 添加记录

INSERT INTO <表名>  <列名> VALUES <数据>

USE指定数据库

在开始操作之前，要先使用USE来指定数据库 ，例如，

USE `example`;

表示接下来的操作都是基于数据库`example`

USE 指定数据库

在SQLyog中，双击打开一个数据库时，内部

自动地调用了USE语句。

INSERT 添加记录

语法格式：

INSERT INTO  <表名>   <列名> VALUES <列值>

例如：

INSERT INTO `student`

(`id`, `name`, `birthday`, `cellphone`)

VALUES

('201511', 'wang', '1992‐12‐2', '18601088987');

1.  <表名> :   表的名称，如 `student`

2. <列名>  :   每一列的名称，用` ` 括起来

3. <列值> :  每一列的值。用单引号或双引号括 起来。

注：<列名> VALUES <列值>, 列名与列值要一一对 应。

INSERT 添加记录

注意：

(1)列名与列值要一一对应

(2)不一定要给出每一列的值

(3)可以一次INSERT多行数据

(4)空值以NULL指定

(5)表名、字段名用反引号 `xxx`，值用引号' '

(6)列的顺序其实改变

空值设成NULL，例如，把`birthday`的值设为 NULL

INSERT INTO `student`

(`id`, `name`, `birthday`, `cellphone`)

VALUES

('201512', 'li', NULL, '18601980879')  ;

注：前提是，该列在定义的时候允许为空才行。

一次插入多行记录，每行以括号括起来，中 间以逗号分隔。

INSERT INTO `student`

(`id`, `name`, `cellphone`)

VALUES

('201514', 'jiang', '18612338987'),

('201515', 'zheng', '18622238987')  ;

列的顺序与值的顺序只要对应就行，可以自 行调整列的顺序。比如，把`name`放在前面。

INSERT INTO `student`

(`name`, `id`,  `cellphone`)

VALUES

('zhou', '201516',  '18234238987');

INSERT 添加记录

不同的数据类型的值的写法:

1. 数字类型：可以加引号，也可以不加引号。

'123' 或 123都可以

2. 字符串类型：加引号。

'hello world'

3. 日期/时间：加引号。

'2015‐12‐15 21:30:20'

小节

1. 了解了USE的用法

USE用于指定要操作的目标数据库

2.  学习了INSERT语句的写法

2.4 SELECT查询记录: WHERE

准备

从官网下载example01.sql，导入到MySQL

查询记录

SELECT语句用于查询表中的数据。

SELECT <列>   FROM <表>

如，用\*表示所有列:

SELECT \* FROM `student`;

只查看`id`和`name`:

SELECT `id`, `name` FROM `student`;

查询记录

AS :  列的别名

例如，

SELECT `id` AS `学号` ,  `name` AS `姓名`  FROM `student`;

加上AS之后的结果如下：

WHERE 查询条件

一般来说，用户只想查询符合条件的记录。

比如，

“在1992年出生的”

“姓名以wang开头的”

“学号为201511的”

“在1995年以前出生的”并且“姓名以wang开头 的”。。。

WHERE 查询条件

语法：

SELECT <列>  FROM <表>  WHERE <条件>

其中，<条件> 子句用于判断该条记录是否满足 条件。

例如，

WHERE id=201511

WHERE name='wang bo'

WHERE 条件表达式

一、数值比较

判断 操作符 例子

大于 >      id > 123

等于 =      id = 123

小于 <      id < 123

不等于 <>      id <> 123

大于或等于 >=      id >= 123

小于或等于 <=      id <= 123

二、日期比较：

与数值比较相同，值要用于单引号

例如：

DELETE FROM `student`

WHERE birthday < '1994‐1‐1'

三、字符串比较：

(1) 大小比较

数值比较的几种操作符都支持，意思和C语言 里的strcmp相同。

例如，

name = 'shao fa'

name < 'shao fa'

(2) LIKE 通配符匹配

例如， 查找所有以'shao'开头的

WHERE `name` LIKE 'shao%'

所有以'fa'结尾的

WHERE `name` LIKE '%fa'

其中，%是通配符，可以通配多个字符。\_是单 字符通配符。

(3) REGEXP 正则表达式匹配

MySQL还支持用 正则表达式 来匹配字符串， 不在本课程中讲解。

四、空值比较：判断是否为NULL

是空值：

WHERE `birthday`  IS NULL

不是空值：

WHERE `birthday`  IS NOT NULL

WHERE 逻辑运算符

和C语言中的与或非类似，SQL里也有与或非

AND  :  并且，相当于 &&

OR     :  或者,  相当于 ||

NOT  ：非， 相当于 !

expr1  AND expr2

expr1  OR expr2

NOT expr1

( NOT expr1 )  AND ( expr2  OR expr 3)

如：

WHERE ( `birthday` IS NOT NULL ) AND (`id` < 201516)

小节

1. 学习SELECT语句来作条件查询

2. 掌握WHERE子句中条件表达式的写法，以 及AND, OR, NOT的意义

2.5 SELECT查询记录: 排序与分页

准备

从官网下载example01.sql，导入到MySQL

ORDER BY 对结果排序

接上节课，对SELECT查询的结果，把所有满足条 件的行称为： 结果集 ResultSet

可以使用ORDER BY子句，令MySQL排序后输出。

ASC升序，DESC降序

[asc](https://www.baidu.com/s?wd=asc&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkuHT3uhfLmWfvm1c3mH7-0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHT3PW0LnjT1" \t "https://zhidao.baidu.com/question/_blank)ending order  升序  
  
descending order 降序

ORDER BY <列名>  ASC;

ORDER BY <列名>  DESC;

ORDER BY 对结果排序

例如，

SELECT \* FROM `student`

ORDER BY `birthday` ASC ;

将结果集按birthday升序输出

ORDER BY 对结果排序

如果有WHERE子句，则放在WHERE子句之后。

SELECT \* FROM `student`

WHERE `birthday` IS NOT NULL

ORDER BY `birthday` ASC ;

总而言之，先查询，再对结果集排序。

LIMIT 分页显示

当结果集中的行数太多，无法一次性取回。则应 该使用分页显示。

LIMIT子句用于指定每次取回的行号范围。

LIMIT start, n

其中，

start表示从第几行开始，n表示取多少行。

(注：LIMIT关键字是MySQL特有的)

LIMIT 分页显示

例：

SELECT \* FROM `student`  ORDER BY `id` ASC  LIMIT 0, 4;

表示从结果集中取回4行：0,1,2,3

(start=0, n=4)

问题：怎么知道结果集中一共有多少行?

MySQL函数

MySQL服务器内置了一些“函数”，可以在SQL脚本中 调用。

比如，COUNT函数用于得到结果集的行数

SELECT COUNT(`id`) AS `result` FROM `student`

WHERE NAME LIKE 'shao%'

注：COUNT(`id`) : 对ID字段进行计数 ，AS `result` : 重命名为result

知道结果集的总行数之后，便可以分页显示了。。。

MySQL函数

再比较，DAYOFWEEK()函数用于求一个日期对 应是星期几。

SELECT DAYOFWEEK(`birthday`) AS `result`  FROM `student`

更多的函数请自己百度 " MySQL 函数"

小结

1. ORDER BY 对结果集排序 ，此SQL语句是通用的。

2. LIMIT分页显示 （不通用，只有MySQL支持 这个关键词，别的服务器有别的办法）

3. 在SQL可以使用函数（不通用，必须使用 MySQL支持的函数)

2.6 SELECT查询记录: 联合查询

准备

从官网下载example02.sql，导入到MySQL

背景

表student:  学生的基本信息（学号、姓名…） ，共5个学生

表exam:  某次考试的成绩 ，所有学生全部参考考试 ，记录每人的chinese , english,  math的成绩

表game:  某次竞赛的成绩 ，部分同学参加 ，记录初试成绩preTest,决赛成绩finalTest。

联合查询

任务1：列出本次考试的所有姓名、成绩

考虑：姓名在`student`表中，成绩在`exam`表 中，需要联合查询。

SELECT \* FROM student, exam

WHERE student.id = exam.stuId ;

任务2：列出本次竞赛的参赛者的姓名、成绩

考虑：有5名学生，但参赛者只有2人。

SELECT \* FROM student, game  WHERE student.id = game.stuId ;

注意：只有两个表中的共同项被列出（交集）

联合查询：列的指定

联合查询时，可以分别从两个表中抽取指定 的列。

下面从student表中抽取id, name，从exam表 中抽取chinese:

SELECT student.id, student.name, exam.chinese  FROM student, exam  WHERE exam.`stuId` = student.id;

可以为表设置临时的别名，以简化SQL代码

SELECT t1.id, t1.name, t2.chinese  FROM student  t1,  exam t2  WHERE t1.id = t2.stuId;

在FROM子句中，将student重命名为t1，将 exam重命名为t2

一个小问题：

为什么不把 "学生信息" 和 "考试成绩"整合

在一张表里呢？分成两张表有什么好处？

内联与外联

当两个表联合查询时，分为内联和外联。

内联：INNER JOIN

外联：

(1) 左外联 LEFT OUTER JOIN

(2) 右外联 RIGHT OUTER JOIN

(3) 全外联 FULL  OUTER JOIN

内联 INNER JOIN

前面所说的方法，以下两种方法结果相同：

SELECT \* FROM student , game  WHERE student.id = game.stuId;

SELECT \* FROM student INNER JOIN game  ON student.id = game.stuId;

注意，关键字 … FROM …  INNER JOIN … ON …

ON指定了联接的条件

内联 INNER JOIN

内联查询的特点：

返回两个表的共同项（交集）

左外联 LEFT OUTER JOIN

仍然以`student`和`game`为例：

student：共有5个学生

game：只有2个学生参加了竞赛

问：如何将两个张拼接(Join)起来，在student 为主，game为辅，缺的项以NULL显示？

左外联 LEFT OUTER JOIN

仔细观察：

(1)左student和右game，两表被Join起来

(2)student为主，game为辅，缺项显示NULL

左外联 LEFT OUTER JOIN

左外联合查询：

SELECT \*

FROM student LEFT OUTER JOIN game  ON game.stuId = student.id ;

右外联 RIGHT OUTER JOIN

右外联：右表为主，左表为辅

以表ta, tb为示例:

SELECT \*

FROM ta RIGHT OUTER JOIN tb  ON ta.id = tb.id;

全外联 FULL OUTER JOIN

全外联：就是求并集，缺项填NULL

(MySQL不支持，可能是因为没有实际意义)

小结

学习了联合查询的几种方式：内联，左联， 右联。

注：LEFT OUTER JOIN 可以简写为LEFT JOIN， 含义不变。

2.7 修改与删除记录

UPDATE 修改记录

使用UPDATE语句可以修改记录的值

UPDATE <表>  SET <列1>= <值1> , <列2>= <值2>  WHERE <条件> ;

(1) 指定表名

(2) WHERE指定限制条件（要修改哪些行？）

(3) SET指定新的值

例如，

UPDATE `student`  SET `birthday` = '1992‐06‐12'  , `cellphone` = '18612111127'  WHERE `id` = 201514;

注：更新多个字段时，中间以逗号分开

DELETE 删除记录

使用DELETE语句，从数据库中删除记录

删除所有记录

DELETE FROM <表> ;

删除满足条件的记录

DELETE FROM <表>   WHERE <条件> ;

DELETE 删除记录

演示：

DELETE FROM `simple`;

DELETE FROM `student`  WHERE `id` < 201518 AND `id` >201513

小结

1. 学会以UPDATE来更新记录

2. 学会以DELETE来删除记录

至此，已经讲完CRUD四种操作：

C： Create,      创建 (INSERT)

R:     Retrieve,  获取 (SELECT)

U:     Update,   更新 (UPDATE)

D:     Delete,     删除 (DELETE)

3.1 数据库编程与API

数据库编程

数据库编程：就是在我们的应用程序中，用代码来 与数据库服务器交互。

例如，

我们用C++编写一个程序： 用户输入学号，点查询， 便能查到该学生的信息。

数据库编程

SQLyog就是一个基于数据库的应用程序，它 里面就用到了数据库编程的技术。

区别：

SQLyog：面向程序员、DBA、设计人员

我们的应用程序：面向普通用户 ，只需要点一下按钮，就可能查出，操作难 度低。用户不需要懂SQL的语法。

API

那么，如何在我们的代码中与MySQL交互呢？

理论上，使用以下方式可以与MySQL交互：

1. 创建 TCP socket （参考《应用篇》）

2. 根据MySQL的协议，与之交互。

但这样操作难度较大，使得程序员望而却步。

API

从MySQL作者的角度来看：自己开发了一款服务 器产品，自然是想着用户越多越好。。。

那怎么才能让用户都用它的产品呢？

易用: 一个产品越易于使用，就越有可能成功。 而前面的方式显然不太容易: 每个程序员都要先 熟读它的协议，还要写程序调试，有这个必要吗？

所以，MySQL提供了一套库，程序员只需要 调用这个库里的少数的几个函数，就能够与 MySQL交互。(补充篇）

这套库就称为API

API:  Application Programming Interface ，应用程序编程接口

API

最终的架构表示如下：

应用程序调用API里的函数，由API内部与MySQL 交互。

显然，API和MySQL是同一方提供的，它们之 间的交互会比较轻松。

API的优势：

1. 大大降低了用户（程序员）的使用难度

程序员只需要调用几个函数，就能够与MySQL交 互。

2. 屏蔽了内部协议

程序员在使用MySQL时，并不需要它的内部协议。

所以，发布方将来可以很容易升级协议 （更新 一份新的API库就行）

小结

1. 介绍了什么是数据库编程

应用程序中使用代码进行数据库操作

2. 介绍了什么是API

API就是一个库，调用里面提供的函数接口便 可以与服务器交互。

3.2 准备MySQL开发环境

获取API

在MySQL 5.5的安装包中已经自带了API库

获取文档 Document

只有头文件和lib，如果没有文档说明，也是 没法使用的。

在官网 www.mysql.com 找到其开发文档，按 文档进行。

创建第一个程序

创建第一个程序，在里面使用MySQL的API 在VC中演示。。。

小结

1. 设置好了MySQL的开发环境

2. 建立了第一个数据库应用程序

3.3 执行SQL语句

API初始化

在使用API中的其他函数之前，需要先初始化。

通常在程序启动时初始化（main函数开始处）

mysql\_library\_init

程序退出之前（但由于程序即将退出，所以 不调用也没关系）

mysql\_library\_end

连接服务器

MYSQL：采用面向对象的设计，每个对象对应一个MySQL 连接。

MYSQL conn;

mysql\_init (&conn);

mysql\_real\_connect (&conn, …);

// 指定IP/用户名/密码等参数，连接服务器

mysql\_xxx (…)    相关业务操作

mysql\_close (&conn);  关闭连接

执行SQL语句

这里，只介绍INSERT, UPDATE, DELETE操作，

SELECT操作后面再介绍。。。

mysql\_query 执行SQL语句

mysql\_affected\_rows 受影响的行数

要点：在应用程序中，要拼凑一个SQL语句

注意事项

1. 线程安全：可以在不同的线程创建自己的 MYSQL对象，互相没有影响。

2. 短连接 or 长连接：

长连接：执行完操作之后，不关闭连接，等 下次继续用。

短连接：每次执行完操作之后，立即关闭连 接。若下次需要，则创建新的连接。

推荐使用短连接，不要长期占着一个连接。

小结

1. 介绍MySQL API的一般用法

2. 演示了INSERT等语句折执行, UPDATE与

DELETE也是一样，就是要自己拼凑一个 SQL语句。

3.4 执行SELECT查询

SELECT查询

返回值：

1. 成功 or 失败

2. 如果成功，则返回一个ResultSet（即符合条件的 行）

整体结构

const char\* sql = …      拼凑SQL语句

mysql\_query(&conn, sql)    执行SQL语句

// 保存结果

MYSQL\_RES \* result = mysql\_store\_result(conn);

if ( result )

{

… 处理每一行数据 …

mysql\_free\_result(result) ;   // 释放内存

}

如何取得每一行数据?

MYSQL\_ROW row;

while ((row = mysql\_fetch\_row(result)))

{

unsigned long \*lengths = mysql\_fetch\_lengths(result);

for(int i = 0; i < num\_fields; i++)

{

char\* field = row[i]; // 字段

unsigned int field\_length = lengths[i]; // 字段长度

}

}

还原每个字段

MySQL返回来的值都是字符串类型

1. tinyint, samllint, mediumint, int

调用AfMysql::AsInt()

2. bitint

调用AfMysql::AsBigInt()

3. char, varchar, text

调用AfMysql::AsString()，以std::string保存

4. date, time, datetime

调用AfMysql::AsDate(), AsTime()或AsDateTime()

注意事项

MYSQL\_RES \* result = mysql\_store\_result(conn);

while( mysql\_fetch\_row (…) )

{

}

1. mysql\_store\_result从服务器取回数据

2. 此时，mysql\_fetch\_row是从本地内存读取行集

问题：如果行数太多，则时间要等待太长。。。

注意事项

改成：

MYSQL\_RES \* result = mysql\_use\_result(conn);

while( mysql\_fetch\_row (…) )

{

}

1. mysql\_use\_result: 初始化操作

2. mysql\_fetch\_row: 从服务器获取一行

优点：可以增加与用户的交互友好性

小结

1. SELECT查询的使用方法，处理保存和处理 结果集。

3.5 在Qt项目中使用数据库

Qt中连接数据库

Qt项目：使用了Qt库的C++项目

MySQL项目：使用了MySQL库的C++项目

一个C++项目不能同时使用两个动态库吗？能！

实例

开发一个GUI程序，显示`example`数据库中的 `student`表中的数据。

演示：。。。

开发环境检查

检查MySQL开发环境:

(1) MySQL API位置在 E:\SDK\mysql5.5

(2) VC的系统目录设置(INCLUDE, LIB)

(3) PATH环境变量设置 E:\SDK\mysql5.5\lib

MySQL数据库

数据库`example`

字符集为gbk

代码

创建一个Qt项目，把3\_4B的代码照搬过 来。。。

最好显式地指定Connection所用的字符集…

MYSQL conn;

mysql\_init(&conn);

mysql\_real\_connect(…);

// "gbk" or "utf8"

if (!mysql\_set\_character\_set(&conn, "gbk"))

{

printf("Charset for the connection : %s\n",

mysql\_character\_set\_name(&conn));

}

小结

介绍了在Qt项目中连接MySQL数据库的方法

注意：应显式地指定connect所用的字符集

（客户与服务器通讯时所用的字符编码）