目录

[01、RDBMS（关系型数据库管理系统）： 2](#_Toc513384917)

[02、MySQL管理 2](#_Toc513384918)

[03、PHP链接mysql 3](#_Toc513384919)

[04、PHP创建数据库 3](#_Toc513384920)

[05、PHP删除数据库 4](#_Toc513384921)

[06、PHP选择数据库 4](#_Toc513384922)

[07、PHP创建数据表 4](#_Toc513384923)

[08、PHP删除数据表 4](#_Toc513384924)

[09、PHP向表中插入数据 4](#_Toc513384925)

[10、查询数据 5](#_Toc513384926)

[10、MySQL WHERE子句 5](#_Toc513384927)

[11、PHP更新修改表单中的数据： 5](#_Toc513384928)

[12、PHP删除表中的数据 5](#_Toc513384929)

[13、PHP读取表中的数据 5](#_Toc513384930)

[14、UNION操作符 5](#_Toc513384931)

[15、MySQL排序 5](#_Toc513384932)

[16、分组 5](#_Toc513384933)

[17、连接 6](#_Toc513384934)

[18、UNLL值处理 6](#_Toc513384935)

[19、正则表达式 6](#_Toc513384936)

[20、事务 7](#_Toc513384937)

[21、索引 7](#_Toc513384938)

[22、复制表 8](#_Toc513384939)

[23、MySQL元数据 8](#_Toc513384940)

[24、MySQL序列 8](#_Toc513384941)

[25、处理重复数据 8](#_Toc513384942)

[26、导出导入数据 8](#_Toc513384943)

# 01、RDBMS（关系型数据库管理系统）：

* RDBMS的特点：

1、数据以表格的形式出现

2、每行为各种记录名称

3、每列为记录名称所对应的数据域

4、许多的行和列组成一张表单

5、若干的表单组成database（数据库）

* RDBMS术语：

1、数据库：关联表的集合

2、数据表：数据的矩阵

3、列：一列（数据元素）包含了相同的数据

4、行：一行（元组或记录）是一组相关的数据

5、冗余：存储两倍的数据，冗余降低了性能，但提高了安全性

6、主键：主键是唯一的。一个数据表中只能包含一个主键，可以用主键查询数据

7、外键：用于关联两个表

8、复合键：复合键（组合键）将多个列作为一个索引键，一般用于复合索引

9、索引：使用索引可快速访问数据库表中的特定信息。索引是对数据表中一列或多列的值进行排序的一种结构，类似于书籍和目录。

10、参照完整性：参照的完整性要求关系中不允许引用不存在的实体，与实体完整性是关系模型必须满足的完整性约束条件，目的是保证数据的一致性。

# 02、MySQL管理

* USE database\_name：使用某个数据库
* SHOW DATABASES：列出所有数据库
* SHOW TABLES：列出某个数据库中的所有表
* SHOW COLUMNS FROM table\_name：显示数据表的属性

DESC table\_name：显示数据表的属性

* SHOW INDEX FROM table\_name：显示数据表的详细索引信息
* SHOW CREATE TABLE table\_name：显示数据表创建时的详细信息，用于数据表复制。
* CREATE DATABASE database\_name：创建数据库
* DROP DATABASE database\_name：删除数据库
* CREATE TABLE table\_name(column\_name coulmn\_type，……)：创建数据表
* DROP TABLE table\_name：删除数据表
* INSERT INTO table\_name(column\_name，……) VALUES (value，……)，(value, ……)：向表中插入数据

INSERT IGONRE INTO table\_name

(column\_name，……)

VALUES

(value，……), (value, ……)：忽略重复数据插入，插入重复数据会报错，并显示重复数据，插入多行数据时，重复数据不插入并报错，不重复的数据会插入。

* UPDATE table\_name SET column\_name，……=new-values，……

WHERE clause

更新修改表单中的数据，WHERE指定更改的位置，不指定的话则更改整个列。

* DELETE FROM table\_name

WHERE clause

删除表中的数据，WHERE指定删除的位置，不指定的话则删除所有数据。

* SELECT column\_name…… FROM table\_name：读取表中的数据

SELECT \* FROM table\_name：读取表中的全部数据

* ALTER TABLE table\_name DROP column\_name：删除表中的某列。

ALTER TABLE table\_name ADD column\_name column\_type [AFTER column\_name]：在表中新加入某一列，可以选择加在某列之后，默认最后，写FIRST则插入第一列。

ALTER TABLE table\_name MODIFY column\_name new\_column\_type：修改表中某列的类型。

ALTER TABLE table\_name CHANGE old\_column\_name new\_column\_name new\_type：修改表中的旧列为新列，并指定新列的类型。

ALTER TABLE old\_table\_name RENAME TO new\_table\_name：修改表名。

SHOW TABLE STATUS：查看表类型。

# 03、PHP链接mysql

* 语法：mysqli\_connect(host, uesrname, password, dbname, port, socket)：连接数据库

host：主机名或ip地址

username：用户名

password：用户密码

dbname：选择默认使用的数据库

port：链接到MySQL服务器的端口号

socket：规定socket或使用已命名的pipe

注意：链接成功后会返回一个标识符，这个标识符作为本次链接的标志。

mysqli\_close(mysqli\_link)：关闭连接的数据库

mysqli\_link：数据库的连接标识

* 连接本地服务器的例子：

<?php

$dbhost = 'localhost:3306'; //本地服务器地址

$dbuser = 'root'; //用户名

$dbpass = ''; //没有设定密码

$conn = mysqli\_connect($dbhost, $dbuser, $dbpass);//连接数据库，把连接标识赋 值给$conn

if(!$conn){

echo 'could not connect';

}

else{

echo 'could connect';

}

mysqli\_close($conn); //关闭数据库

?>

输出：could connect

# 04、PHP创建数据库

概述：mysqli\_query()函数可以将MySQL命令由PHP脚本传递给MySQL。

* mysqli\_query(connection, query, resultode)

connection：MySQL连接标识

query：要执行的MySQL命令

resultmode：常量，MYSQLI\_USE\_RESULT（检索大量数据时使用）

MYSQLI\_STORE\_RESULT（默认）

* PHP创建数据库例子：

<?php

//数据库连接没有写

$create\_database = "CREATA DATABASE runoob"; //数据库命令，创建一个数据库。

$retval = mysqli\_query($conn, $create\_database);

if(!$retval){

die(“create database fail”.mysqli\_error($conn)) //返回数据框数据框错误信息

}

else{

echo "create database success";

}

?>

输出：create database success

# 05、PHP删除数据库

* 使用mysqli\_query()函数

例子：

<?php

//数据库连接没有写

$create\_database = "DROP DATABASE RUNOOB"; //数据库命令

$retval = mysqli\_query($conn, $create\_database);

if(!$retval){

die(“drop database fail”.mysqli\_error($conn))

}

else{

echo "drop database success";

}

?>

输出：drop database success

# 06、PHP选择数据库

* mysqli\_select\_db(connection, database\_name)
* mysqli\_query(connection, query, resultode)

两种方法都可以

# 07、PHP创建数据表

* mysqli\_query(connection, query, resultode)

# 08、PHP删除数据表

* mysqli\_query(connection, query, resultode)

# 09、PHP向表中插入数据

* mysqli\_query(connection, query, resultode)

# 10、查询数据

* mysqli\_query(connection, query, resultode)：用这个函数获取数据。
* mysqli\_fetch\_array( )：从获取数组函数的结果中取得一行作为关联数组。

# 10、MySQL WHERE子句

描述：WHERE子句是MySQL中的条件语句，可以为MySQL表达式增设条件（使用AND或者OR指定一个或多个条件）

# 11、PHP更新修改表单中的数据：

* mysqli\_query(connection, query, resultode)

# 12、PHP删除表中的数据

* mysqli\_query(connection, query, resultode)

# 13、PHP读取表中的数据

* mysqli\_query(connection, query, resultode)
* MySQL中的LIKE子句，一般和%和-一起使用。

# 14、UNION操作符

概述：用于连接两个以上的SELECT语句的结果组合到一个结果集合中，多个SELECT语句会删除重复的数据。

* 语法：SELECT column\_name……

FROM table\_name

[WHERE clause]

UNION [ALL | DISTINCT]

SELECT column\_name……

FROM table\_name

[WHERE clause];

说明：column\_name：要检索的列，可以有多个

table\_name：要检索的数据表

WHERE clause：检索的条件

DISTINCT：删除结果中的重复数据

ALL：返回的数据集中，包含重复的数据。

# 15、MySQL排序

概述：可以用ORDER BY子句对读取的数据进行排序。

* 语法：一个SELECT语句使用ORDER BY子句对查询数据排序后再返回数据的例子：

SELECT column\_name……FROM table\_name……

ORDER BY column\_name……[ASC or DESC]

注释：column\_name[ASC or DESC]：作为排序的字段是升序还是降序，默认是升序。

# 16、分组

概述：根据一个或多个列对数据集进行分组，在分组的列上可以使用一些统计函数。

* 语法：SELECTE column, function(column) //分组被列出的字段

FROM table\_name //从哪个表获取字段

WHERE [clause] //设置查询条件

GROUP BY column [HAVING clause] //按某字段分组，且满足条件的组才能输出

ORDER BY column\_name [ASC|DESC] //按该字段中的数据进行排序，默认升序排 列

* group\_concat(column\_name)：显示分组中某字段的全部组员信息，在select中使用。
* 条件：

|  |  |
| --- | --- |
| 查询条件 | 符号 |
| 比较 | =, <, <=, >, >=, !=, <>, !>, !< |
| 指定范围 | BETWEEN AND, NOT BETWEEN AND |
| 指定集合 | IN, NOT IN |
| 匹配字符 | LINKE, IS ONT LIKE |
| 是否为空 | IS NULL, IS NOT NULL |
| 多个条件查询 | AND, OR |
| 通配符 | %匹配任意一个或多个字符，- 一个字符 |

# 17、连接

概述：从多个表中读取数据，并把数据放在一起读出。

* 语法：SELECT table1.column, table2.column //连接的表要显示的字段

FROM table1\_name [INNER, LEFT, RIGHT] JOIN table2\_name //哪些表连接

ON clause //连接表的条件，一般是对字段设置条件，显示符合条件的字段。

注释：

1、INNER JOIN：获取两个表中字段匹配关系的记录，内连接。

2、LEFT JOIN：获取左边表的所有记录，即使右边没有对应的匹配，外连接。

3、RIGHT JOIN：获取右边的所有记录，即使右边没有对应的匹配，外连接。

# 18、UNLL值处理

概述：MySQL中使用条件语句查询个表中的值为NULL的项时，用WHERE object=NULL

得不到结果，这与MySQL中的NULL值的性质设置有关，NULL不能与任何值相 比较。

* IS NULL：当值为NULL时返回true

IS NOT NULL：当值不为NULL时返回true

<=>：当比较的的两个值为NULL时返回true

例：SELECT \* FROM runoob\_test\_tbl WHERE runoob\_count IS NULL;

查找runoob\_count列中所有的空值。

# 19、正则表达式

概述：可以用正则表达式来进行匹配

* 语法：REGEXP 正则表达式
* ^：匹配输入字符串的开始位置

$：匹配字符串的结束位置

. ：匹配除‘\n’之外的任意单个字符

[……]：匹配任何单个字符

[^……]：匹配未包含的任意字符

\*：匹配前面的子表达式零次或多次

+：匹配前面的字符串一次或多次

{n}：匹配n次

{n,m}：匹配n到m次

|：或

# 20、事务

概述：一系列数据库操作语句构成事务。

* 事务控制语句：

1、BEGIN：显示地开启一个事务

2、COMMIT：提交事务，并使对数据库进行的所有修改变为永久。

3、ROLLBACK：回滚，结束事务，并撤销正在进行的所有为提交的修改。

4、SAVEPOINT identifier：创建一个保存点，一个事务可以有多个保存点。

5、ROLLBACK TO identifier：把事务回滚到标记点。

6、SET TRANSACTION：设置事务的隔离级别

# 21、索引

* 1、表建立后创建索引：

CREATE INDEX index\_name ON table\_name(column\_name(length))

INDEX：索引类型

index\_name：索引名称

table\_name：在哪个表创建索引

column\_name：哪些列作为索引

2、添加索引：

ALTER TABLE table\_name ADD INDEX index\_name(column\_name)

3、创建表时指定：

INDEX index\_name (column\_name(length))

4、删除索引：

DROP INDEX index\_name ON table\_name

注意：索引的名字是可选的，不指定的话以索引列表的第一个字段名为索引名。

* 索引的分类

1、普通索引：一张表中可以有多个普通索引

创建标识：INDEX 或 KEY

2、唯一索引：一张表中可以有多个唯一索引

创建标识：UNIQUE KEY

3、主键索引：一张表中只能有一个主键，被标志为自动增长的字段一定是主键，但主 键不一定是自动增长的，主键的类型最好是数值。

创建标识：PRIMARY KEY

4、全文索引：

创建标识：FULL TEXT

5、外键索引：外键会自动和对应的其他表的主键关联

创建标识：FOREIGN KEY

* SHOW INDEX FROM table\_name：显示索引信息

# 22、复制表

概述：赋值一个表

* 语法步骤：

第一步：使用SHOW CREATE TABLE table\_name语句，获取表结构信息，并复制。

第二步：使用CREATE TABLE table\_name语句创建新表，并将旧表中的表结构信息复制到新表中作为新表的结构信息，这时已经创建了一个和旧表有相同结构的新表。注意：表结构信息包括，字段名，字段类型，索引等。

第三步：使用INSERT INTO new\_table\_name(column\_name)

SELECT(column\_name)

FROM(old\_table\_name)

将旧表中的数据复制到新表中去。

# 23、MySQL元数据

* SELECT VERSION()：服务器版本信息
* SELECT DATABASE()：当前数据库名（或者返回空）
* SELECT USER()：当前用户名
* SHOW STATUS：服务器状态
* SHOW VARIABLES：服务器配置变量

# 24、MySQL序列

概述：MySQL序列是一组整数：1，2，3……，由于一张数据表只能有一个字段自增主键，若想实现其他字段也实现自动增加，就可以使用MySQL序列来实现。

* 语法：AUTO\_INCREMENT：定义一个序列

# 25、处理重复数据

* 防止出现重复数据：将字段设置为PRIMARY KEY(主键)或者UNIQUE(唯一)索引来保证数据唯一性。
* 统计重复数据：

SELECT COUNT(\*) as repetitions, column\_name //将column\_name中重复的数赋值给repeitions，然后显示出来

FROM table\_name

GROUP BY column\_name

HAVE repetitions>n; //条件，设置显示重复次数为n次的column\_name

* 过滤重复数据DISTINCT关键字：

SELECT DISTINCT column\_name…… FROM table\_name //显示所有不重复的数据

* 删除重复数据

# 26、导出导入数据