## MySQL 连接

mysql -u root –p

## MySQL 创建数据库

create database 数据库名;

mysqladmin -u root -p create数据库名;

## MySQL 删除数据库

drop database <数据库名>;

mysqladmin -u root -p drop <数据库名>;

## MySQL 选择数据库

use <数据库名>

## MySQL 数据类型

MySQL支持多种类型，大致可以分为三类：数值、日期/时间和字符串(字符)类型。

日期和时间类型

表示时间值的日期和时间类型为DATETIME、DATE、TIMESTAMP、TIME和YEAR。

字符串类型

字符串类型指CHAR、VARCHAR、BINARY、VARBINARY、BLOB、TEXT、ENUM和SET。该节描述了这些类型如何工作以及如何在查询中使用这些类型。

## MySQL 创建数据表

创建MySQL数据表需要以下信息：

* 表名
* 表字段名
* 定义每个表字段

语法

以下为创建MySQL数据表的SQL通用语法：

CREATE TABLE table\_name (column\_name column\_type);

CREATE TABLE IF NOT EXISTS `runoob\_tbl`(

`runoob\_id` INT UNSIGNED AUTO\_INCREMENT,

`runoob\_title` VARCHAR(100) NOT NULL,

`runoob\_author` VARCHAR(40) NOT NULL,

`submission\_date` DATE,

PRIMARY KEY ( `runoob\_id` )

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

如果你不想字段为 NULL 可以设置字段的属性为 NOT NULL， 在操作数据库时如果输入该字段的数据为NULL ，就会报错。

AUTO\_INCREMENT定义列为自增的属性，一般用于主键，数值会自动加1。

PRIMARY KEY关键字用于定义列为主键。 您可以使用多列来定义主键，列间以逗号分隔。

ENGINE 设置存储引擎，CHARSET 设置编码。

## MySQL 删除数据表

以下为删除MySQL数据表的通用语法：

DROP TABLE table\_name ;

## MySQL 插入数据

MySQL 表中使用 INSERT INTO SQL语句来插入数据。

以下为向MySQL数据表插入数据通用的 INSERT INTO SQL语法：

INSERT INTO table\_name ( field1, field2,...fieldN )

VALUES

( value1, value2,...valueN );

## MySQL 查询数据

MySQL 数据库使用SQL SELECT语句来查询数据。

以下为在MySQL数据库中查询数据通用的 SELECT 语法：

SELECT column\_name,column\_name

FROM table\_name

[WHERE Clause]

[LIMIT N][ OFFSET M]

查询语句中你可以使用一个或者多个表，表之间使用逗号(,)分割，并使用WHERE语句来设定查询条件。

SELECT 命令可以读取一条或者多条记录。

你可以使用星号（\*）来代替其他字段，SELECT语句会返回表的所有字段数据

你可以使用 WHERE 语句来包含任何条件。

你可以使用 LIMIT 属性来设定返回的记录数。

你可以通过OFFSET指定SELECT语句开始查询的数据偏移量。默认情况下偏移量为0。

## MySQL WHERE 子句

我们知道从 MySQL 表中使用 SQL SELECT 语句来读取数据。

如需有条件地从表中选取数据，可将 WHERE 子句添加到 SELECT 语句中。

语法

以下是 SQL SELECT 语句使用 WHERE 子句从数据表中读取数据的通用语法：

SELECT field1, field2,...fieldN FROM table\_name1, table\_name2...

[WHERE condition1 [AND [OR]] condition2.....

查询语句中你可以使用一个或者多个表，表之间使用逗号, 分割，并使用WHERE语句来设定查询条件。

你可以在 WHERE 子句中指定任何条件。

你可以使用 AND 或者 OR 指定一个或多个条件。

WHERE 子句也可以运用于 SQL 的 DELETE 或者 UPDATE 命令。

WHERE 子句类似于程序语言中的 if 条件，根据 MySQL 表中的字段值来读取指定的数据。

## MySQL UPDATE 更新

如果我们需要修改或更新 MySQL 中的数据，我们可以使用 SQL UPDATE 命令来操作。

语法

以下是 UPDATE 命令修改 MySQL 数据表数据的通用 SQL 语法：

UPDATE table\_name SET field1=new-value1, field2=new-value2

[WHERE Clause]

你可以同时更新一个或多个字段。

你可以在 WHERE 子句中指定任何条件。

你可以在一个单独表中同时更新数据。

当你需要更新数据表中指定行的数据时 WHERE 子句是非常有用的。

## MySQL DELETE 语句

你可以使用 SQL 的 DELETE FROM 命令来删除 MySQL 数据表中的记录。。

语法

以下是 SQL DELETE 语句从 MySQL 数据表中删除数据的通用语法：

DELETE FROM table\_name [WHERE Clause]

如果没有指定 WHERE 子句，MySQL 表中的所有记录将被删除。

你可以在 WHERE 子句中指定任何条件

您可以在单个表中一次性删除记录。

当你想删除数据表中指定的记录时 WHERE 子句是非常有用的。

## MySQL 事务

MySQL 事务主要用于处理操作量大，复杂度高的数据。比如说，在人员管理系统中，你删除一个人员，你即需要删除人员的基本资料，也要删除和该人员相关的信息，如信箱，文章等等，这样，这些数据库操作语句就构成一个事务！

在 MySQL 中只有使用了 Innodb 数据库引擎的数据库或表才支持事务。

事务处理可以用来维护数据库的完整性，保证成批的 SQL 语句要么全部执行，要么全部不执行。

事务用来管理 insert,update,delete 语句

一般来说，事务是必须满足4个条件（ACID）：：原子性（Atomicity，或称不可分割性）、一致性（Consistency）、隔离性（Isolation，又称独立性）、持久性（Durability）。

原子性：一个事务（transaction）中的所有操作，要么全部完成，要么全部不完成，不会结束在中间某个环节。事务在执行过程中发生错误，会被回滚（Rollback）到事务开始前的状态，就像这个事务从来没有执行过一样。

一致性：在事务开始之前和事务结束以后，数据库的完整性没有被破坏。这表示写入的资料必须完全符合所有的预设规则，这包含资料的精确度、串联性以及后续数据库可以自发性地完成预定的工作。

隔离性：数据库允许多个并发事务同时对其数据进行读写和修改的能力，隔离性可以防止多个事务并发执行时由于交叉执行而导致数据的不一致。事务隔离分为不同级别，包括读未提交（Read uncommitted）、读提交（read committed）、可重复读（repeatable read）和串行化（Serializable）。

持久性：事务处理结束后，对数据的修改就是永久的，即便系统故障也不会丢失。

在 MySQL 命令行的默认设置下，事务都是自动提交的，即执行 SQL 语句后就会马上执行 COMMIT 操作。因此要显式地开启一个事务务须使用命令 BEGIN 或 START TRANSACTION，或者执行命令 SET AUTOCOMMIT=0，用来禁止使用当前会话的自动提交。

事务控制语句：

BEGIN 或 START TRANSACTION 显式地开启一个事务；

COMMIT 也可以使用 COMMIT WORK，不过二者是等价的。COMMIT 会提交事务，并使已对数据库进行的所有修改成为永久性的；

ROLLBACK 也可以使用 ROLLBACK WORK，不过二者是等价的。回滚会结束用户的事务，并撤销正在进行的所有未提交的修改；

SAVEPOINT identifier，SAVEPOINT 允许在事务中创建一个保存点，一个事务中可以有多个 SAVEPOINT；

RELEASE SAVEPOINT identifier 删除一个事务的保存点，当没有指定的保存点时，执行该语句会抛出一个异常；

ROLLBACK TO identifier 把事务回滚到标记点；

SET TRANSACTION 用来设置事务的隔离级别。InnoDB 存储引擎提供事务的隔离级别有READ UNCOMMITTED、READ COMMITTED、REPEATABLE READ 和 SERIALIZABLE。

MYSQL 事务处理主要有两种方法：

1、用 BEGIN, ROLLBACK, COMMIT来实现

BEGIN 开始一个事务

ROLLBACK 事务回滚

COMMIT 事务确认

2、直接用 SET 来改变 MySQL 的自动提交模式:

SET AUTOCOMMIT=0 禁止自动提交

SET AUTOCOMMIT=1 开启自动提交