修改提示符

方法一：mysql -uroot -proot --prompt \h

结果：localhost

方法2：localhostprompt mysql

结果:mysql

提示符：\D:完整的日期\d：当前数据库\h：服务器名称\u：当前用户

use mysql

SELECT VERSION();当前服务器版本

SELECT NOW();当前日期

SELECT USER();当前用户

规范：

关键字和函数名称全部大写

数据库名称，表名称，字段名称全部小写

SQL语句必须以分号结尾

创建新数据库：CREATE DATABASE first

查看数据库：SHOW DATABASES

创建重名数据库忽略error：CREATE DATABASE IF NOT EXISTS first;

提示1warning

查看警告：SHOW WARNINGS;

查看创建数据库的字符集：SHOW CREATE DATABASE first;

创建自定义字符集数据库：CREATE DATABASE IF NOT EXISTS second CHARACTER SET gbk;

修改字符集：ALTER DATABASE second CHARACTER SET=utf8;

删除数据库：DROP DATABASE first;

数据类型:

整型：

TINYINT

SMALLINT

MEDIUMINT

INT

BIGINT

浮点型：

FLOAT[{M,D}]

DOUBLE[{M,D}]

日期时间型：

YEAR

TIME

DATE

DATETIME

TIMESTAMP

字符型

CHAR

VARCHAR

TINYTEXT

TEXT

MEDIUMTEXT

LOGNTEXT

ENUM('value1','value2')

SET('value1,'value2',)

表：

SELECT DATABASE();显示当前用户打开的数据库

创建表

CREATE TABLE tb1(

username VARCHAR(20),

age TINYINT UNSIGNED,不能为负数

salary FLOAT(8,2) UNSIGNED 总共8位 小数2位

);

SHOW TABLES 显示当前数据库表

SHOW TABLES FROM mysql显示mysql下的所有表，当前数据库不变

SHOW COLUMNS FROM tb1查看tb1表结构

INSERT tb1 VALUES('Tom',25,1564.1)插入所有数据

INSERT tb1(username,salary) VALUES('Tom',1562.3);部分插入

SELECT \* FROM tb1查看数据

AUTO\_INCREMENT自动编号，必须定义为主键

CREATE TABLE tb3(

id SMALLINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(20));

唯一约束 UNIQUE KEY

可保证记录的唯一性，字段可为null，每张数据表可以存在多个唯一约束

默认约束 DEFAULT

当插入记录时，如果没有明确为字段赋值，将自动赋予默认值

sex ENUM('1','2','3') DEFAULT '3'

约束分为表级约束和列级约束

列级约束 只针对某一个字段

表级约束 多个字段

约束类型：5种

非空约束

主键约束

唯一约束

默认约束

外键约束FOREIGN KEYp:保证数据一致性完整性，实现一对一或一对多关系

编辑数据表的默认存储引擎

default-storage-engine=INNODB

外键约束 必须完全一致

mysql> CREATE TABLE province (父表，子表所参照的表

id SMALLINT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,参照列,主键，创建时已有索引

pname varchar(20) NOT NULL);

-----------------------------------------------------------------------------+

mysql> CREATE TABLE users(创建users表外键依赖province

-> id SMALLINT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

-> username VARCHAR(20) NOT NULL,

-> pid SMALLINT UNSIGNED,外键列

-> FOREIGN KEY(pid) REFERENCES province (id));子表 有外键的表

SHOW INDEXES FROM province 显示索引

SHOW INDEXES FROM province\G 以网格形式显示索引

系统将自动为外键列添加索引

mysql> CREATE TABLE users1(创建users1表外键依赖province并增加 ON DELETE CASCADE);

-> id SMALLINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

-> username VARCHAR(20) NOT NULL,

-> pid SMALLINT UNSIGNED,

-> FOREIGN KEY(pid) REFERENCES province(id) ON DELETE CASCADE);cascade

mysql> INSERT province(pname) VALUES('A');父表插入数据pname

mysql> INSERT province(pname) VALUES('B');

mysql> INSERT province(pname) VALUES('C');

mysql> SELECT \* FROM province;

+----+-------+

| id | pname |

+----+-------+

| 1 | A |

| 2 | B |

| 3 | C |

+----+-------+

mysql> INSERT users1(username,pid) VALUES('Tom',3);

Query OK, 1 row affected (0.04 sec)

mysql> INSERT users1(username,pid) VALUES('John',7);pid依赖父表id，id7在父表中不存在，可改成1可执行

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`first`.`users1`, CONSTRAINT `users1\_ibfk\_1` FOREIGN KEY (`pid`) REFERENCES `province` (`id`) ON DELETE CASCADE)

mysql> INSERT users1(username,pid) VALUES('John',1);

Query OK, 1 row affected (0.04 sec)

mysql> INSERT users1(username,pid) VALUES('Rose',3);

Query OK, 1 row affected (0.06 sec)

mysql> SELECT \* FROM users1;查看users1表中数据，虽然7插入未成功，仍消耗一个id

+----+----------+------+

| id | username | pid |

+----+----------+------+

| 1 | Tom | 3 |

| 3 | John | 1 |

| 4 | Rose | 3 |

+----+----------+------+

3 rows in set (0.00 sec)

mysql> DELETE FROM province WHERE id=3;删除父表id3记录

Query OK, 1 row affected (0.18 sec)

mysql> SELECT \* FROM users1;子表中pid为3的记录同时也被删除

+----+----------+------+

| id | username | pid |

+----+----------+------+

| 3 | John | 1 |

+----+----------+------+

1 row in set (0.00 sec)

添加列age

mysql> ALTER TABLE users1 ADD age TINYINT UNSIGNED NOT NULL DEFAULT 10;

添加列password指定在username列后面

ALTER TABLE users1 ADD password VARCHAR(32) NOT NULL AFTER username;

添加列truename位于第一列

ALTER TABLE users1 ADD truename VARCHAR(20) NOT NULL FIRST ;

添加多列:只能从最后一列开始添加，不能指定插定位置

删除列

ALTER TABLE users1 DROP truename;

删除多列

ALTER TABLE users1 DROP password,DROP age;

添加主键约束，只能有一个

mysql> ALTER TABLE users2 ADD id SMALLINT UNSIGNED;

mysql> ALTER TABLE users2 ADD CONSTRAINT PK\_users2\_id PRIMARY KEY(id); constraint

添加唯一约束，可以有多个

mysql> ALTER TABLE users2 ADD UNIQUE (username);

添加外键约束

mysql> ALTER TABLE users2 ADD FOREIGN KEY(pid) REFERENCES province(id);

添加/删除默认约束

mysql> ALTER TABLE users2 ADD age TINYINT UNSIGNED NOT NULL;

mysql> ALTER TABLE users2 ALTER age SET DEFAULT 15;

mysql> ALTER TABLE users2 ALTER age DROP DEFAULT;

删除主键约束

mysql> ALTER TABLE users2 DROP PRIMARY KEY;

删除唯一约束

mysql> SHOW INDEXES FROM users2\G;查看INDEX,存在username

mysql> ALTER TABLE users2 DROP INDEX username;

mysql> SHOW INDEXES FROM users2\G;再次查看，只存在一个INDEX

删除外键约束

mysql> SHOW CREATE TABLE users2;查看外键名

mysql> ALTER TABLE users2 DROP FOREIGN KEY users2\_ibfk\_1;删除外键

修改列定义

mysql> ALTER TABLE users2 MODIFY id SMALLINT UNSIGNED NOT NULL FIRST;位置

mysql> ALTER TABLE users2 MODIFY id TINYINT UNSIGNED NOT NULL FIRST;数据类型

修改列名称 CHANGE比MODIFY更强大，还可以修改列名称

mysql> ALTER TABLE users2 CHANGE pid p\_id TINYINT UNSIGNED NOT NULL;

数据表改名

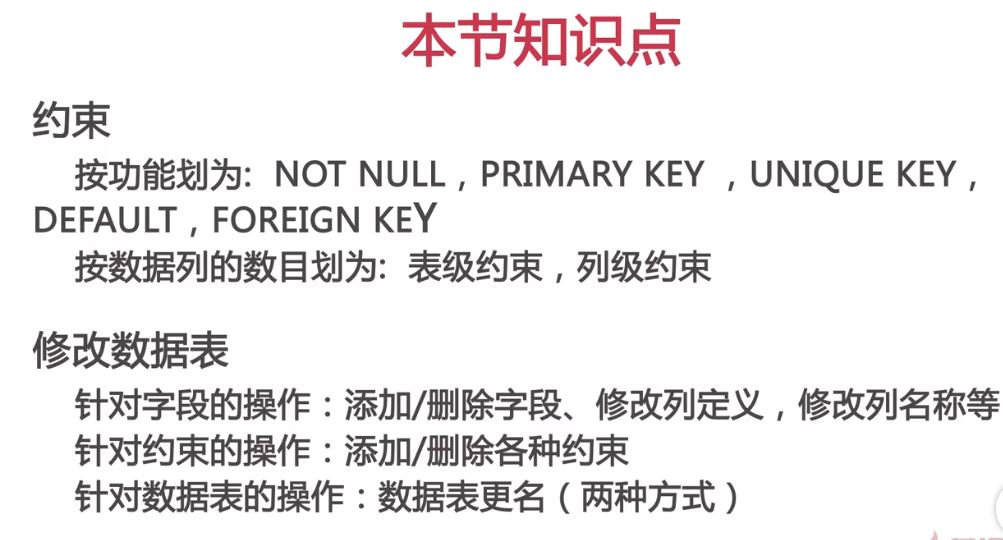
方法1：mysql> ALTER TABLE users2 RENAME users3;

方法2：mysql> RENAME TABLE users3 TO users2;

默认约束和非空约束只能为列级约束

其他均可为表级约束

总结



mysql> ALTER TABLE users ADD enjoyment VARCHAR(20) NOT NULL;

mysql> ALTER TABLE users ALTER enjoyment SET DEFAULT 'swimming';

mysql> INSERT users VALUES(DEFAULT,'Jack',456,23,1,DEFAULT);

mysql> SELECT \* FROM users;

+----+----------+----------+-----+------+-----------+

| id | username | password | age | sex | enjoyment |

+----+----------+----------+-----+------+-----------+

| 1 | Tom | 123 | 18 | 1 | |

| 2 | John | 12356 | 26 | 1 | |

| 3 | Jack | 456 | 23 | 1 | swimming |

+----+----------+----------+-----+------+-----------+

3 rows in set (0.00 sec)