## 库的管理

### 1、库的创建

create database books;

### 2、库的修改

改名：rename database books to dictionary;//有数据丢失的危险，官方不推荐

alter database books character set utf8;

### 3、库的删除

drop database if exists books;

## 表的管理

### 1、创建表

* 语法格式

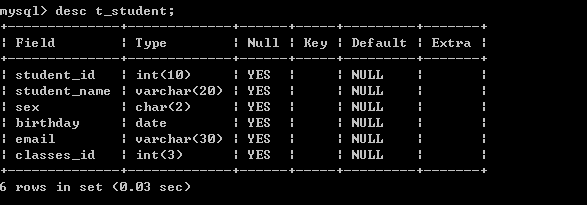
|  |
| --- |
| create table tableName(  columnName dataType(length),  ………………..  columnName dataType(length)  );  set character\_set\_results='gbk';  show variables like '%char%';  创建表的时候，表中有字段，每一个字段有：  \* 字段名  \* 字段数据类型  \* 字段长度限制  \* 字段约束 |

* MySql常用数据类型

|  |  |
| --- | --- |
| 类型 | 描述 |
| Char(长度) | 定长字符串，存储空间大小固定，适合作为主键或外键 |
| Varchar(长度) | 变长字符串，存储空间等于实际数据空间 |
| double(有效数字位数，小数位) | 数值型 |
| Float(有效数字位数，小数位) | 数值型 |
| Int( 长度) | 整型 |
| bigint(长度) | 长整型 |
| Date | 日期型 年月日 |
| DateTime | 日期型 年月日 时分秒 毫秒 |
| time | 日期型 时分秒 |
| BLOB | Binary Large OBject（二进制大对象） |
| CLOB | Character Large OBject（字符大对象） |
| 其它………………… |  |

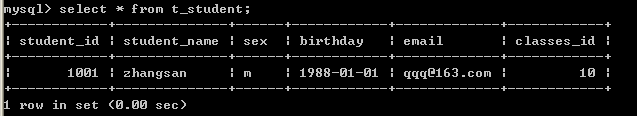
* 建立学生信息表，字段包括：学号、姓名、性别、出生日期、email、班级标识

|  |
| --- |
| use books;  drop tables t\_student;  create table t\_student(  student\_id int(10),  student\_name varchar(20),  sex char(2),  birthday date,  email varchar(30),  classes\_id int(3)  );  desc t\_student; |



* 向t\_student表中加入数据

|  |
| --- |
| insert into t\_student(student\_id, student\_name, sex, birthday, email, classes\_id) values(1001, 'zhangsan', 'm', '1988-01-01', 'qqq@163.com', 10);  select \* from t\_student; |



* 向t\_student表中加入数据（使用默认值）

|  |
| --- |
| drop table if exists t\_student;  create table t\_student(  student\_id int(10),  student\_name varchar(20),  sex char(2) default 'm',  birthday date,  email varchar(30),  classes\_id int(3)  );  insert into t\_student(student\_id, student\_name, birthday, email, classes\_id)  values(1002, 'zhangsan', '1988-01-01', 'qqq@163.com', 10)  select \* from t\_student; |

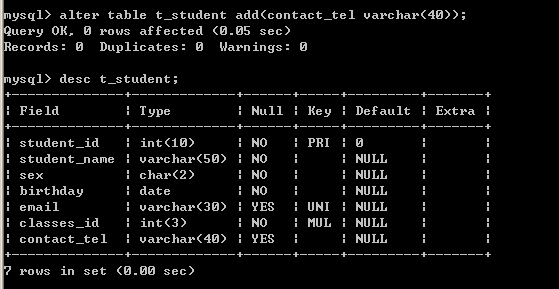
### 2、增加/删除/修改表结构

采用alter table来增加/删除/修改表结构，不影响表中的数据

#### 2.1、添加字段

如：需求发生改变，需要向t\_student中加入联系电话字段，字段名称为：contatct\_tel 类型为varchar(40)

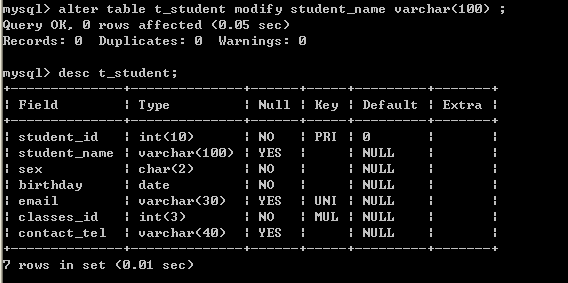
|  |
| --- |
| alter table t\_student add contact\_tel varchar(40); |



#### 2.2、修改字段

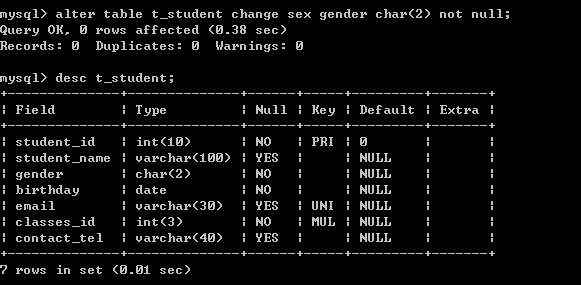
如：student\_name无法满足需求，长度需要更改为100

|  |
| --- |
| alter table t\_student modify student\_name varchar(100) ; |



如sex字段名称感觉不好，想用gender那么就需要更改列的名称

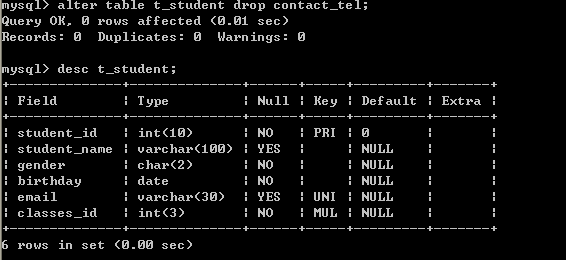
alter table t\_student change sex gender char(2) not null;



#### 2.3、删除字段

如：删除联系电话字段

|  |
| --- |
| alter table t\_student drop contact\_tel; |



### 3、添加、修改和删除

#### 3.1、insert

CREATE TABLE EMP

(EMPNO int(4) not null ,

ENAME VARCHAR(10),

JOB VARCHAR(9),

MGR INT(4),

HIREDATE DATE DEFAULT NULL,

SAL DOUBLE(7,2),

COMM DOUBLE(7,2),

primary key (EMPNO),

DEPTNO INT(2)

);

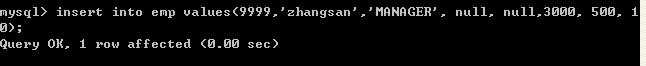
添加、修改和删出都属于DML，主要包含的语句：insert、update、delete

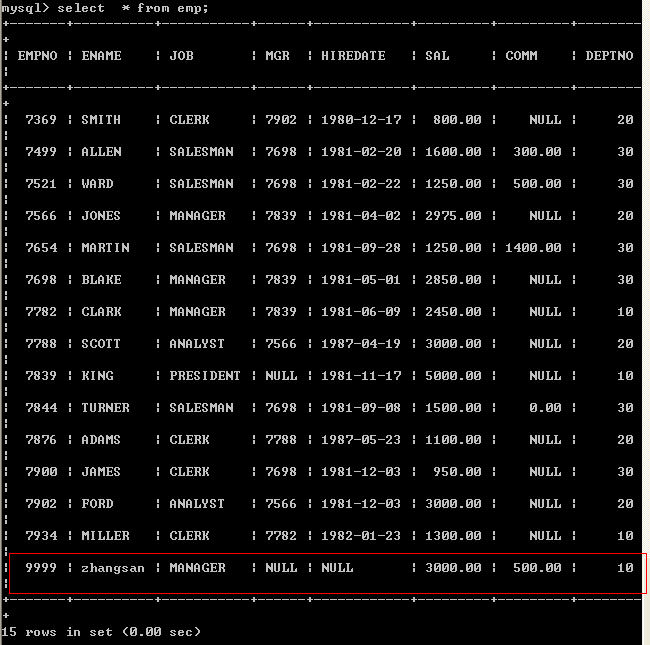
* Insert语法格式

|  |
| --- |
| Insert into 表名(字段，。。。。) values(值,………..) |

* 省略字段的插入

|  |
| --- |
| insert into emp values(9999,'zhangsan','MANAGER', null, null,3000, 500, 10); |





不建议使用此种方式，因为当数据库表中的字段位置发生改变的时候会影响到insert语句

* 指定字段的插入(建议使用此种方式)

|  |
| --- |
| insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values(9999,'zhangsan','MANAGER', null, null,3000, 500, 10); |



出现了主键重复的错误，主键表示了记录的唯一性，不能重复



如何插入日期：

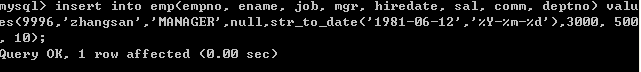
**第一种方法，插入的日期格式和显示的日期格式一致**

|  |
| --- |
| insert into emp(empno, ename, job, mgr, hiredate, sal, comm, deptno) values(9997,'zhangsan','MANAGER', null, '1981-06-12',3000, 500, 10); |



**第二种方法，采用str\_to\_date**

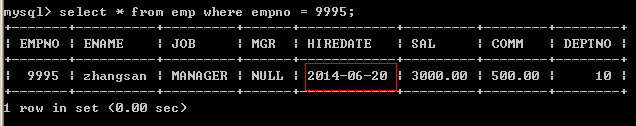
|  |
| --- |
| insert into emp(empno, ename, job, mgr, hiredate, sal, comm, deptno) valu  es(9996,'zhangsan','MANAGER',null,str\_to\_date('1981-06-12','%Y-%m-%d'),3000, 500, 10); |



**第三种方法，添加系统日期（now()）**

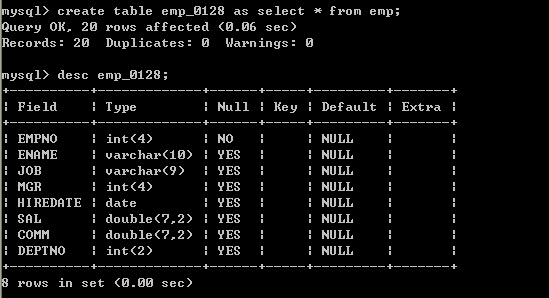
|  |
| --- |
| insert into emp(empno, ename, job, mgr, hiredate, sal, comm, deptno) values(9995,'zhangsan','MANAGER',null,now() ,3000, 500, 10); |

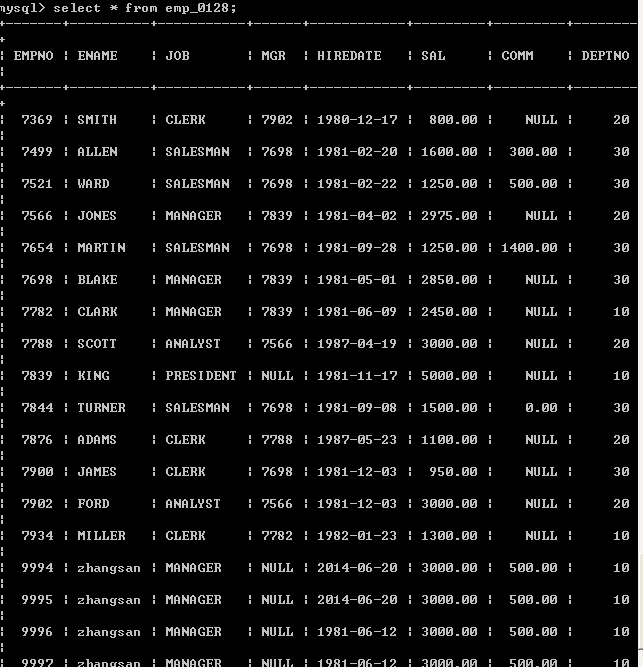




* 表复制

|  |
| --- |
| create table emp\_bak as select empno,ename,sal from emp; |





以上方式，会自动创建表，将符合查询条件的数据自动复制到创建的表中

* 如何将查询的数据直接放到已经存在的表中，可以使用条件

|  |
| --- |
| insert into emp\_bak select \* from emp where sal=3000; |



#### 3.2、update

可以修改数据，可以根据条件修改数据

* 语法格式：

|  |
| --- |
| update 表名 set 字段名称1=需要修改的值1, 字段名称2=需要修改的值2 where ……. |

* 将job为manager的员工的工资上涨10%

|  |
| --- |
| update emp set sal=sal+sal\*0.1 where job='MANAGER'; |

#### 3.3、delete

可以删除数据，可以根据条件删除数据

* 语法格式：

|  |
| --- |
| Delete from表名 where 。。。。。 |

* 删除津贴为500的员工

|  |
| --- |
| delete from emp where comm=500; |

* 删除津贴为null的员工

|  |
| --- |
| delete from emp where comm is null; |

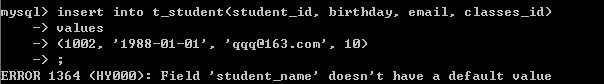
### 4、创建表加入约束

* 常见的约束
  1. 非空约束，not null
  2. 唯一约束，unique
  3. 主键约束，primary key
  4. 外键约束，foreign key
  5. 默认值： default
  6. 自定义检查约束，check（不建议使用）(在mysql中现在还不支持)

#### 4.1、非空约束，not null

非空约束，针对某个字段设置其值不为空，如：学生的姓名不能为空

|  |
| --- |
| drop table if exists t\_student;  create table t\_student(  student\_id int(10),  student\_name varchar(20) not null,  sex char(2) default 'm',  birthday date,  email varchar(30),  classes\_id int(3)  )  insert into t\_student(student\_id, birthday, email, classes\_id)  values  (1002, '1988-01-01', 'qqq@163.com', 10) |

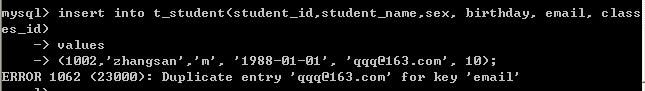
****

以上错误为加入的学生姓名为空。

#### 4.2、唯一约束，unique

唯一性约束，它可以使某个字段的值不能重复，如：email不能重复：

|  |
| --- |
| drop table if exists t\_student;  create table t\_student(  student\_id int(10),  student\_name varchar(20) not null,  sex char(2) default 'm',  birthday date,  email varchar(30) unique,  classes\_id int(3)  )  insert into t\_student(student\_id, student\_name , sex, birthday, email, classes\_id)  values  (1001,'zhangsan','m', '1988-01-01', 'qqq@163.com', 10) |



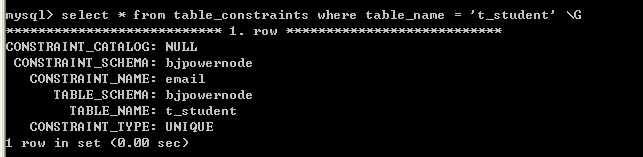
以上插入了重复的email，所以出现了“违反唯一约束错误”，所以unique起作用了

同样可以为唯一约束起个约束名

* 我们可以查看一下约束

mysql> use information\_schema;

mysql> select \* from table\_constraints where table\_name = 't\_student';



关于约束名称可以到table\_constraints中查询

以上约束的名称我们也可以自定义。

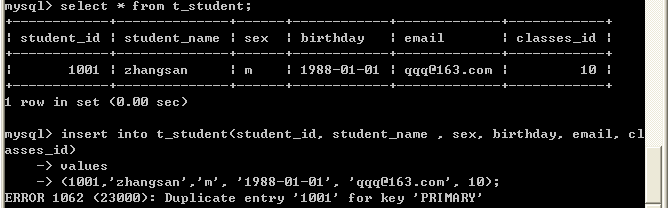
|  |
| --- |
| drop table if exists t\_student;  create table t\_student(  student\_id int(10),  student\_name varchar(20) not null,  sex char(2) default 'm',  birthday date,  email varchar(30) ,  classes\_id int(3) ,  constraint email\_unique unique(email)/\*表级约束\*/  ) |

#### 4.3、主键约束，primary key

每个表应该具有主键，主键可以标识记录的唯一性，主键分为单一主键和复合（联合）主键，单一主键是由一个字段构成的，复合（联合）主键是由多个字段构成的

|  |
| --- |
| drop table if exists t\_student;  create table t\_student()  student\_id int(10) primary key,/\*列级约束\*/  student\_name varchar(20) not null,  sex char(2) default 'm',  birthday date,  email varchar(30) ,  classes\_id int(3)  )  insert into t\_student(student\_id, student\_name , sex, birthday, email, classes\_id)  values  (1001,'zhangsan','m', '1988-01-01', 'qqq@163.com', 10) |

向以上表中加入学号为1001的两条记录，出现如下错误，因为加入了主键约束



我们也可以通过表级约束为约束起个名称：

|  |
| --- |
| drop table if exists t\_student;  create table t\_student(  student\_id int(10),  student\_name varchar(20) not null,  sex char(2) default 'm',  birthday date,  email varchar(30) ,  classes\_id int(3),  CONSTRAINT p\_id PRIMARY key (student\_id)  )  insert into t\_student(student\_id, student\_name , sex, birthday, email, classes\_id)  values  (1001,'zhangsan','m', '1988-01-01', 'qqq@163.com', 10) |

#### 4.4、外键约束，foreign key

外键主要是维护表之间的关系的，主要是为了保证参照完整性，如果表中的某个字段为外键字段，那么该字段的值必须来源于参照的表的主键，如：emp中的deptno值必须来源于dept表中的deptno字段值。

建立学生和班级表之间的连接

首先建立班级表t\_classes

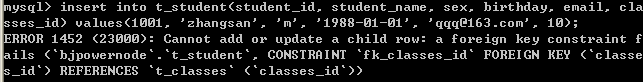
|  |
| --- |
| drop table if exists t\_classes;  create table t\_classes(  classes\_id int(3),  classes\_name varchar(40),  constraint pk\_classes\_id primary key(classes\_id)  ) |

在t\_student中加入外键约束

|  |
| --- |
| drop table if exists t\_student;  create table t\_student(  student\_id int(10),  student\_name varchar(20),  sex char(2),  birthday date,  email varchar(30),  classes\_id int(3),  constraint student\_id\_pk primary key(student\_id),  constraint fk\_classes\_id foreign key(classes\_id) references t\_classes(classes\_id)  ) |

向t\_student中加入数据

|  |
| --- |
| insert into t\_student(student\_id, student\_name, sex, birthday, email, classes\_id) values(1001, 'zhangsan', 'm', '1988-01-01', 'qqq@163.com', 10) |



出现错误，因为在班级表中不存在班级编号为10班级，外键约束起到了作用

存在外键的表就是子表，参照的表就是父表，所以存在一个父子关系，也就是主从关系，主表就是班级表，从表就是学生表



以上成功的插入了学生信息，当时classes\_id没有值，这样会影响参照完整性，所以我们建议将外键字段设置为非空

|  |
| --- |
| drop table if exists t\_student;  create table t\_student(  student\_id int(10),  student\_name varchar(20),  sex char(2),  birthday date,  email varchar(30),  classes\_id int (3) not null,  constraint student\_id\_pk primary key(student\_id),  constraint fk\_classes\_id foreign key(classes\_id) references t\_classes(classes\_id)  )  insert into t\_student(student\_id, student\_name, sex, birthday, email, cla  sses\_id) values(1001, 'zhangsan', 'm', '1988-01-01', 'qqq@163.com', null); |

再次插入班级编号为null的数据



添加数据到班级表，添加数据到学生表，删除班级数据，将会出现如下错误：

|  |
| --- |
| insert into t\_classes (classes\_id,classes\_name) values (10,'366');  insert into t\_student(  student\_id, student\_name, sex, birthday, email, classes\_id  ) values(  1001, 'zhangsan', 'm', '1988-01-01', 'qqq@163.com', 10  )  mysql> update t\_classes set classes\_id = 20 where classes\_name = '366';    因为子表（t\_student）存在一个外键classes\_id，它参照了父表（t\_classes）中的主键，所以先删除子表中的引用记录，再修改父表中的数据。  我们也可以采取以下措施 级联更新。  mysql> delete from t\_classes where classes\_id = 10;    因为子表（t\_student）存在一个外键classes\_id，它参照了父表（t\_classes）中的主键，所以先删除父表，那么将会影响子表的参照完整性，所以正确的做法是，先删除子表中的数据，再删除父表中的数据，采用drop table也不行，必须先drop子表，再drop父表  我们也可以采取以下措施 级联删除。 |