# SQL语句DDL

## DataGrip

```sql
#显示所有数据库
show databases ;

#创建数据库
create database database1;

#如果不存在database1就创建
create database if not exists database1;

#使用数据库
use database1;

#创建表
create table student(
    sid int(11),
    name varchar(255), #字符
    age int(11)
);

# 删除数据库
drop database database1;

#如果存在database1就删除
drop database if exists database1;

#修改数据库编码
alter database database1 char set utf8;

desc database1.t_stu;

#修改表结构
alter table student add tel char(11);#添加一列
alter table student drop age;#删除类
alter table student modify column tel int(11);#修改列数据类型
alter table student change tel telephone char(11);#修改tel列为telephone

# 修改表名
rename table student to t_stu;

#查看建表语句
show create table t_stu;
CREATE TABLE `t_stu` (
  `sid` int DEFAULT NULL,
  `name` varchar(20) DEFAULT NULL,
  `gender` varchar(20) DEFAULT NULL,
  `birth` date DEFAULT NULL,
  `address` varchar(20) DEFAULT NULL,
  `score` double DEFAULT NULL,
  `telephone` char(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

```

#查看建库语句
show create database database1;
CREATE DATABASE `database1` /*!40100 DEFAULT CHARACTER SET utf8mb3 */
/*!80016 DEFAULT ENCRYPTION='N' */


#插入一条数据
insert into database1.student (sid, `name`, age) values(0,"王五",24);
```

## 数据表的约束

```
1.主键约束
主键约束即primary key用于唯一的标识表中的每一行。被标识为主键的数据在表中是唯一的且其值不能为空。
这点类似于我们每个人都有一个身份证号，并且这个身份证号是唯一的。
主键约束基本语法：
字段名  数据类型   primary key;
设置主键约束的两种方式
1. create table student (id int primary key);
2. create table student (id int , primary key (id));
-----------------------------------------------------------------
2.非空约束
非空约束即not null指的是字段到的值不能为空，

非空约束基本语法：
字段名  数据类型  not null;
-----------------------------------------------------------------
3.默认值约束
默认值约束即DEFAULT用于给数据表中的字段指定默认值，即当在表中插入一条新记录时若未给该字段赋值，那么，数据库系统会自动为这个字段插人默认值；

默认值约束语法：
字段名  数据类型  default 默认值;
-----------------------------------------------------------------
4.唯一性约束
唯一性约束即unique用于保证数据表中字段的唯一性，即表中字段的值不能重复出现，其基本的语法格式如下所示：

唯一性约束语法：
字段名  数据类型  unique;
-----------------------------------------------------------------
5.外键约束
外键约束即foreign key常用于多张表之间的约束

基本语法：
1. 创建数据表时指定：constraint  外键名  foreign key (从表外键字段) references  主表 （主键字段）;
2. 创建数据表后指定：alter table 表名  add constraint 外建名  foreign key(从表外键字段) references（主键字段);
-----------------------------------------------------------------
create table if not exists student(
    sid int primary key  auto_increment,
    name varchar(20) unique ,
    age int(11) not null
```

```
39 );
```

# 数值类型

整型

| 类型 | 大小 | 范围（有符号） | 范围（无符号） | 用途 |
| --- | --- | --- | --- | --- |
| TINYINT | 1 byte | (-128，127) | (0，255) | 小整数值 |
| SMALLINT | 2 bytes | (-32 768，32 767) | (0，65 535) | 大整数值 |
| MEDIUMINT | 3 bytes | (-8 388 608，8 388 607) | (0，16 777 215) | 大整数值 |
| INT或INTEGER | 4 bytes | (-2 147 483 648，2 147 483 647) | (0，4 294 967 295) | 大整数值 |
| BIGINT | 8 bytes | (-9,223,372,036,854,775,808，9 223 372 036 854 775 807) | (0，18 446 744 073 709 551 615) | 极大整数值 |
| FLOAT | 4 bytes | (-3.402 823 466 E+38，3.402 823 466 351 E+38) | 0，(1.175 494 351 E-38，3.402 823 466 E+38) | 单精度 浮点数值 |
| DOUBLE | 8 bytes | (-1.797 693 134 862 315 7 E+308，1.797 693 134 862 315 7 E+308) | 0，(2.225 073 858 507 201 4 E-308，1.797 693 134 862 315 7 E+308) | 双精度 浮点数值 |
| DECIMAL | | 依赖于M和D的值 | 依赖于M和D的值 | 小数值 |

字符串

| 类型 | 大小 | 用途 |
| --- | --- | --- |
| CHAR | 0-255 bytes | 定长字符串 |
| VARCHAR | 0-65535 bytes | 变长字符串 |
| TINYBLOB | 0-255 bytes | 不超过 255 个字符的二进制字符串 |
| TINYTEXT | 0-255 bytes | 短文本字符串 |
| BLOB | 0-65 535 bytes | 二进制形式的长文本数据 |
| TEXT | 0-65 535 bytes | 长文本数据 |
| MEDIUMBLOB | 0-16 777 215 bytes | 二进制形式的中等长度文本数据 |
| MEDIUMTEXT | 0-16 777 215 bytes | 中等长度文本数据 |
| LONGBLOB | 0-4 294 967 295 bytes | 二进制形式的极大文本数据 |
| LONGTEXT | 0-4 294 967 295 bytes | 极大文本数据 |

日期

| 类型 | 大小 ( bytes) | 范围 | 格式 | 用途 |
| --- | --- | --- | --- | --- |
| DATE | 3 | 1000-01-01/9999-12-31 | YYYY-MM-DD | 日期值 |
| TIME | 3 | '-838:59:59'/'838:59:59' | HH:MM:SS | 时间值或持续时间 |
| YEAR | 1 | 1901/2155 | YYYY | 年份值 |
| DATETIME | 8 | 1000-01-01 00:00:00/9999-12-31 23:59:59 | YYYY-MM-DD HH:MM:SS | 混合日期和时间值 |
| TIMESTAMP | 4 | 1970-01-01 00:00:00/2038<br>结束时间是第 2147483647 秒，北京时间 2038-1-19 11:14:07，格林尼治时间 2038年1月19日 凌晨 03:14:07 | YYYYMMDD HHMMSS | 混合日期和时间值，时间戳 |

# 对表结构的常用操作-其他操作

| 功能 | SQL |
| --- | --- |
| 查看当前数据库的所有表名称 | show tables; |
| 查看指定某个表的创建语句 | show create table 表名; |
| 查看表结构 | desc 表名 |
| 删除表 | drop table 表名 |

```
-- 查看当前数据所有的表
show tables;

-- 查看指定表的创建语句
show create table student;

CREATE TABLE `student` (
  `sid` int NOT NULL AUTO_INCREMENT,
  `name` varchar(20) DEFAULT NULL,
  `age` int NOT NULL,
  PRIMARY KEY (`sid`),
  UNIQUE KEY `name` (`name`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb3

-- 查看表结构
desc student;

+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| sid   | int         | NO   | PRI | NULL    | auto_increment |
| name  | varchar(20) | YES  | UNI | NULL    |                |
| age   | int         | NO   |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+

-- 删除表
drop table student;
```

## 对表结构的常用操作-修改表结构格式

```
-- 添加列
alter table student add dept varchar(20);
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| sid   | int         | NO   | PRI | NULL    | auto_increment |
| name  | varchar(20) | YES  | UNI | NULL    |                |
| age   | int         | NO   |     | NULL    |                |
| dept  | varchar(20) | YES  |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+


-- 修改列名和类型
alter table student change dept department varchar(30);
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| sid        | int         | NO   | PRI | NULL    | auto_increment |
| name       | varchar(20) | YES  | UNI | NULL    |                |
| age        | int         | NO   |     | NULL    |                |
| department | varchar(30) | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+


-- 修改表删除列
alter table student drop department;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| sid   | int         | NO   | PRI | NULL    | auto_increment |
| name  | varchar(20) | YES  | UNI | NULL    |                |
| age   | int         | NO   |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+


-- 修改表名
rename table student to stu;
+---------------------+
| Tables_in_database1 |
+---------------------+
| stu                 |
+---------------------+
```

## 数据库操作DML

```
-- 数据插入
insert into student (sid,name,age)values(5,'老七',19),(6,'老八',20);

insert into student(sid,age) values(100,20);


insert into student values(7,'老九',21),(8,'老十',22);
```

```
sid name age
1   张三  20
2   李四  20
3   王五  24
4   老六  18
5   老七  19
6   老八  20
7   老九  21
8   老十  22
100     20
```

```
-----------------------------------------------------------------
```

```
-- 数据修改
-- 将所有学生的年纪修改为20
update student set address ='武汉';

-- 将4的地址改为葛店
update student set address ='葛店' where sid =4;

update student set address ='长职' where sid >4;

-- 将id为3的学生地址修改为光谷 年龄修改为18
update student set address = '光谷',age=18 where sid =3;
```

```
-----------------------------------------------------------------
```

```
-- 数据删除
-- 删除sid为4的学生数据
delete from student where sid =4;

-- 删除表所有数据
delete from student;

-- 清空表数据
truncate table student;
truncate student;
```

```
-----------------------------------------------------------------
```

```
-- 案例
-- 创建员工表employee
-- id name gender salary
create table if not exists database1.employee(
id int,
name varchar(20),
gender varchar(10),
salary double
);

-- 插入数据
-- 1 张三 男 2000
-- 2 李四 男 1000
-- 3 王五 女 4000
insert into employee values(1,'张三','男',2000),(2,'李四','男',1000),(3,'王五','女',4000);

-- 修改表数据
```

```sql
-- 将所有员工薪水修改为5000元
update employee  set salary=5000;

-- 将姓名为张三的员工薪水修改为3000元
update employee set salary=3000 where name = '张三';

-- 将姓名为李四的员工薪水修改为4000元 gender改为女
update employee set salary=4000,gender='女' where name = '李四';

-- 将王五的薪水在原有的基础上增加1000元
update employee set salary=salary+1000 where name = '王五';
```

# Mysql约束

主键约束

方式1-语法：

```sql
-- 在 create table 语句中，通过 PRIMARY KEY 关键字来指定主键。
--在定义字段的同时指定主键，语法格式如下：
create table 表名(
    ...
    <字段名> <数据类型> primary key
    ...
)
```

方式1-实现：

```sql
create table emp1(
    eid int primay key,
    name VARCHAR(20),
    deptId int,
    salary double
);
```

方式2-语法：

```sql
--在定义字段之后再指定主键，语法格式如下：
create table 表名(
    ...
    [constraint <约束名>] primary key [字段名]
);
```

方式2-实现：

```sql
create table emp2(
    eid INT,
    name VARCHAR(20),
    deptId INT,
    salary double,
    constraint  pk1 primary key(id)
);
```

# DOS

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| database1          |
| information_schema |
| mysql              |
```

```
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use database1
Database changed
mysql> show tables;
+---------------------+
| Tables_in_database1 |
+---------------------+
| student             |
+---------------------+
1 row in set (0.00 sec)

mysql> desc student;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| sid   | int         | NO   | PRI | NULL    | auto_increment |
| name  | varchar(20) | YES  | UNI | NULL    |                |
| age   | int         | NO   |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
```