

SQL语句DDL

DataGrip

```
1  #显示所有数据库
2  show databases ;
3
4  #创建数据库
5  create database database1;
6
7  #如果不存在database1就创建
8  create database if not exists database1;
9
10 #使用数据库
11 use database1;
12
13 #创建表
14 create table student(
15     sid int(11),
16     name varchar(255), #字符
17     age int(11)
18 );
19
20 # 删除数据库
21 drop database database1;
22
23 #如果存在database1就删除
24 drop database if exists database1;
25
26 #修改数据库编码
27 alter database database1 char set utf8;
28
29 desc database1.t_stu;
30
31 #修改表结构
32 alter table student add tel char(11);#添加一列
33 alter table student drop age;#删除类
34 alter table student modify column tel int(11);#修改列数据类型
35 alter table student change tel telephone char(11);#修改tel列为telephone
36
37 # 修改表名
38 rename table student to t_stu;
39
40 #查看建表语句
41 show create table t_stu;
42 CREATE TABLE `t_stu` (
43     `sid` int DEFAULT NULL,
44     `name` varchar(20) DEFAULT NULL,
45     `gender` varchar(20) DEFAULT NULL,
46     `birth` date DEFAULT NULL,
47     `address` varchar(20) DEFAULT NULL,
48     `score` double DEFAULT NULL,
49     `telephone` char(11) DEFAULT NULL
50 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

```

51
52 #查看建库语句
53 show create database database1;
54 CREATE DATABASE `database1` /*!40100 DEFAULT CHARACTER SET utf8mb3 */
    /*!80016 DEFAULT ENCRYPTION='N' */
55
56
57 #插入一条数据
58 insert into database1.student (sid, `name`, age) values(0,"王五",24);

```

数据表的约束

```

1  1.主键约束
2  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。
3  这点类似于我们每个人都有身份证号，并且这个身份证号是唯一的。
4  主键约束基本语法：
5  字段名 数据类型 primary key;
6  设置主键约束的两种方式
7  1. create table student (id int primary key);
8  2. create table student (id int , primary key (id));
9  -----
10 2.非空约束
11 非空约束即not null指的是字段到的值不能为空，
12
13 非空约束基本语法：
14 字段名 数据类型 not null;
15  -----
16 3.默认值约束
17 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入默认值；
18
19 默认值约束语法：
20 字段名 数据类型 default 默认值；
21  -----
22 4.唯一性约束
23 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：
24
25 唯一性约束语法：
26 字段名 数据类型 unique;
27  -----
28 5.外键约束
29 外键约束即foreign key常用于多张表之间的约束
30
31 基本语法：
32 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)；
33 2. 创建数据表后指定：alter table 表名 add constraint 外键名 foreign key(从表外键
    字段) references (主键字段)；
34  -----
35 create table if not exists student(
36     sid int primary key auto_increment,
37     name varchar(20) unique ,
38     age int(11) not null

```

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 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

```
1  -- 查看当前数据所有的表
2  show tables;
3
4  -- 查看指定表的创建语句
5  show create table student;
6
7  CREATE TABLE `student` (
8    `sid` int NOT NULL AUTO_INCREMENT,
9    `name` varchar(20) DEFAULT NULL,
10   `age` int NOT NULL,
11   PRIMARY KEY (`sid`),
12   UNIQUE KEY `name` (`name`)
13 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb3
14
15 -- 查看表结构
16 desc student;
17
18 +-----+-----+-----+-----+-----+-----+
19 | Field | Type          | Null | Key | Default | Extra          |
20 +-----+-----+-----+-----+-----+-----+
21 | sid   | int           | NO   | PRI | NULL    | auto_increment |
22 | name  | varchar(20)   | YES  | UNI | NULL    |                |
23 | age   | int           | NO   |     | NULL    |                |
24 +-----+-----+-----+-----+-----+-----+
25
26 -- 删除表
27 drop table student;
```

对表结构的常用操作-修改表结构格式

```
1  -- 添加列
2  alter table student add dept varchar(20);
3
4  +-----+-----+-----+-----+-----+-----+
5  | Field | Type          | Null | Key | Default | Extra          |
6  +-----+-----+-----+-----+-----+-----+
7  | sid   | int           | NO   | PRI | NULL    | auto_increment |
8  | name  | varchar(20)   | YES  | UNI | NULL    |                |
9  | age   | int           | NO   |     | NULL    |                |
10 | dept  | varchar(20)   | YES  |     | NULL    |                |
11 +-----+-----+-----+-----+-----+-----+
12
13 -- 修改列名和类型
14 alter table student change dept department varchar(30);
15
16 +-----+-----+-----+-----+-----+-----+
17 | Field          | Type          | Null | Key | Default | Extra          |
18 +-----+-----+-----+-----+-----+-----+
19 | sid            | int           | NO   | PRI | NULL    | auto_increment |
20 | name           | varchar(20)   | YES  | UNI | NULL    |                |
21 | age            | int           | NO   |     | NULL    |                |
22 | department     | varchar(30)   | YES  |     | NULL    |                |
23 +-----+-----+-----+-----+-----+-----+
24
25 -- 修改表删除列
26 alter table student drop department;
27
28 +-----+-----+-----+-----+-----+-----+
29 | Field | Type          | Null | Key | Default | Extra          |
30 +-----+-----+-----+-----+-----+-----+
31 | sid   | int           | NO   | PRI | NULL    | auto_increment |
32 | name  | varchar(20)   | YES  | UNI | NULL    |                |
33 | age   | int           | NO   |     | NULL    |                |
34 +-----+-----+-----+-----+-----+-----+
35
36 -- 修改表名
37 rename table student to stu;
38
39 +-----+-----+
40 | Tables_in_database1 |
41 +-----+-----+
42 | stu                  |
43 +-----+-----+
```

数据库操作DML

```
1  -- 数据插入
2  insert into student (sid,name,age) values(5,'老七',19),(6,'老八',20);
3
4  insert into student(sid,age) values(100,20);
5
6
7  insert into student values(7,'老九',21),(8,'老十',22);
```

```

8
9  sid name age
10 1 张三 20
11 2 李四 20
12 3 王五 24
13 4 老六 18
14 5 老七 19
15 6 老八 20
16 7 老九 21
17 8 老十 22
18 100 20
19 -----
20
21 -- 数据修改
22 -- 将所有学生的年纪修改为20
23 update student set address = '武汉';
24
25 -- 将4的地址改为葛店
26 update student set address = '葛店' where sid =4;
27
28 update student set address = '长职' where sid >4;
29
30 -- 将id为3的学生地址修改为光谷 年龄修改为18
31 update student set address = '光谷',age=18 where sid =3;
32
33 -----
34 -- 数据删除
35 -- 删除sid为4的学生数据
36 delete from student where sid =4;
37
38 -- 删除表所有数据
39 delete from student;
40
41 -- 清空表数据
42 truncate table student;
43 truncate student;
44
45 -----
46 -- 案例
47 -- 创建员工表employee
48 -- id name gender salary
49 create table if not exists database1.employee(
50 id int,
51 name varchar(20),
52 gender varchar(10),
53 salary double
54 );
55
56 -- 插入数据
57 -- 1 张三 男 2000
58 -- 2 李四 男 1000
59 -- 3 王五 女 4000
60 insert into employee values(1,'张三','男',2000),(2,'李四','男',1000),(3,'王
五','女',4000);
61
62 -- 修改表数据

```

```

63  -- 将所有员工薪水修改为5000元
64  update employee set salary=5000;
65
66  -- 将姓名为张三的员工薪水修改为3000元
67  update employee set salary=3000 where name = '张三';
68
69  -- 将姓名为李四的员工薪水修改为4000元 gender改为女
70  update employee set salary=4000,gender='女' where name = '李四';
71
72  -- 将王五的薪水在原有的基础上增加1000元
73  update employee set salary=salary+1000 where name = '王五';

```

Mysql约束

主键约束

方式1-语法:

```

-- 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:

```

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

```

方式2-语法:

```

--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:

```

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

```

```

1  -- 联合主键
2  -- 所谓的联合主键 就是这个主键是由一张表中多个字段组成的
3  -- primary key (字段1, 字段2....., 字段n)
4  create table emp3(
5  name varchar(20),
6  deptId int,
7  salary double,
8  constraint pk2 primary key(name,deptId)
9  );

```

```

10
11 insert into emp3 values('张三',10,5000);
12 insert into emp3 values('张三',20,5000);
13
14 -- 联合主键的各列 每一列都不能为空
15 insert into emp3 values(NULL,30,5000);
16 insert into emp3 values('赵六',NULL,5000);
17 insert into emp3 values(NULL,NULL,5000);
18 -----
19
20 -- 添加单列主键
21 create table emp4(
22     eid int,
23     name varchar(20),
24     deptId int,
25     salary double
26 );
27
28 alter table emp4 add primary key(eid);
29
30
31 -- 创建多列主键
32 create table emp5(
33     eid int,
34     name varchar(20),
35     deptId int,
36     salary double
37 );
38
39 alter table emp5 add primary key(name,deptId);
40 -----
41 -- 删除主键约束
42 alter table emp5 drop primary key;

```

自增长约束

```

1  -- 自增长约束
2  use database1;
3  create table t_user1(
4  id int primary key auto_increment,
5  name varchar(20)
6  );
7
8  INSERT into t_user1 values(null,'张三');
9  INSERT into t_user1(name) values('李四');
10
11 -- 创建表时指定
12 create table t_user2(
13 id int primary key auto_increment,
14 name varchar(20)
15 )auto_increment=100;
16
17 insert into t_user2 values(null,'张三');
18 insert into t_user2 values(null,'李四');
19 -----

```



```

20
21 -- 创建表之后指定
22 create table t_user3(
23 id int primary key auto_increment,
24 name varchar(20)
25 );
26
27 insert into t_user3 values(null,'张三');
28 insert into t_user3 values(null,'李四');
29 insert into t_user3 values(null,'王五');
30
31 alter table t_user3 auto_increment=100;
32 id name
33 1 张三
34 2 李四
35 100 王五
36 -----
37 delete和truncate在删除后自增列的变化
38
39 delete
40 use database1;
41 create table t_user1(
42 id int primary key auto_increment,
43 name varchar(20)
44 );
45
46 INSERT into t_user1 values(null,'张三');
47 delete from t_user1; -- delete删除数据之后 自增长还是在最后一个值基础上+1
48 id name
49 3 张三
50
51
52 truncate
53 create table t_user2(
54 id int primary key auto_increment,
55 name varchar(20)
56 )auto_increment=100;
57
58 insert into t_user2 values(null,'张三');
59 insert into t_user2 values(null,'李四');
60
61 truncate t_user2; -- truncate删除数据之后 自增长从1开始+1
62 id name
63 1 张三

```

非空约束(not null)

```

1 -- 非空约束
2 -- 创建表时指定
3 create table t_user6(
4 id int,
5 name varchar(20) not null,
6 address varchar(20) not null
7 );
8 +-----+-----+-----+-----+-----+

```

```

9 | Field | Type | Null | Key | Default | Extra |
10 +-----+-----+-----+-----+-----+-----+
11 | id | int | YES | | NULL | |
12 | name | varchar(20) | NO | | NULL | |
13 | address | varchar(20) | NO | | NULL | |
14 +-----+-----+-----+-----+-----+
15
16 -- 创建表之后指定
17 alter table t_user3 modify name varchar(20) not null;
18 +-----+-----+-----+-----+-----+-----+
19 | Field | Type | Null | Key | Default | Extra |
20 +-----+-----+-----+-----+-----+-----+
21 | id | int | NO | PRI | NULL | auto_increment |
22 | name | varchar(20) | NO | | NULL | |
23 +-----+-----+-----+-----+-----+
24
25 -- 删除非空约束
26 alter table t_user6 modify address varchar(20);
27 +-----+-----+-----+-----+-----+-----+
28 | Field | Type | Null | Key | Default | Extra |
29 +-----+-----+-----+-----+-----+-----+
30 | id | int | YES | | NULL | |
31 | name | varchar(20) | NO | | NULL | |
32 | address | varchar(20) | YES | | NULL | |
33 +-----+-----+-----+-----+-----+

```

唯一约束

```

1 -- 唯一约束
2 -- 创建表时添加
3 create table t_user8(
4 id int,
5 name varchar(20),
6 phone_number varchar(20) unique
7 );
8 +-----+-----+-----+-----+-----+-----+
9 | Field | Type | Null | Key | Default | Extra |
10 +-----+-----+-----+-----+-----+-----+
11 | id | int | YES | | NULL | |
12 | name | varchar(20) | YES | | NULL | |
13 | phone_number | varchar(20) | YES | UNI | NULL | |
14 +-----+-----+-----+-----+-----+-----+
15
16 insert into t_user8 values(1001,'张三',null);
17 insert into t_user8 values(1002,'张三1',null);
18 id      name      phone
19 1001    张三      (NULL)
20 1002    张三1     (NULL)
21 在mysql中NULL和任何值都不相同 NULL不等于NULL
22
23 -- 创建表之后指定
24 alter table t_user6 add constraint UNI unique(address);
25 +-----+-----+-----+-----+-----+-----+
26 | Field | Type | Null | Key | Default | Extra |
27 +-----+-----+-----+-----+-----+-----+

```

```

28 | id      | int      | YES |      | NULL |      |
29 | name    | varchar(20) | NO  |      | NULL |      |
30 | address | varchar(20) | YES | UNI | NULL |      |
31 +-----+-----+-----+-----+-----+-----+
32
33 -- 删除唯一约束
34 alter table t_user6 drop index UNI;
35 +-----+-----+-----+-----+-----+-----+
36 | Field | Type      | Null | Key | Default | Extra |
37 +-----+-----+-----+-----+-----+-----+
38 | id     | int       | YES  |     | NULL    |      |
39 | name   | varchar(20) | NO   |     | NULL    |      |
40 | address | varchar(20) | YES  |     | NULL    |      |
41 +-----+-----+-----+-----+-----+-----+

```

默认约束

```

1  -- 默认约束
2  -- 创建表时添加
3  create table t_user10(
4  id int,
5  name varchar(20),
6  address varchar(20) default '鄂州'
7  );
8  +-----+-----+-----+-----+-----+-----+
9  | Field | Type      | Null | Key | Default | Extra |
10 +-----+-----+-----+-----+-----+-----+
11 | id     | int       | YES  |     | NULL    |      |
12 | name   | varchar(20) | YES  |     | NULL    |      |
13 | address | varchar(20) | YES  |     | 鄂州    |      |
14 +-----+-----+-----+-----+-----+-----+
15
16 -- 创建表之后指定
17 alter table t_user10 modify address varchar(20) default '长职';
18 +-----+-----+-----+-----+-----+-----+
19 | Field | Type      | Null | Key | Default | Extra |
20 +-----+-----+-----+-----+-----+-----+
21 | id     | int       | YES  |     | NULL    |      |
22 | name   | varchar(20) | YES  |     | NULL    |      |
23 | address | varchar(20) | YES  |     | 长职    |      |
24 +-----+-----+-----+-----+-----+-----+
25
26 -- 删除默认约束
27 alter table t_user10 modify column address varchar(20) default null;
28 +-----+-----+-----+-----+-----+-----+
29 | Field | Type      | Null | Key | Default | Extra |
30 +-----+-----+-----+-----+-----+-----+
31 | id     | int       | YES  |     | NULL    |      |
32 | name   | varchar(20) | YES  |     | NULL    |      |
33 | address | varchar(20) | YES  |     | NULL    |      |
34 +-----+-----+-----+-----+-----+-----+

```

零填充约束

```
1  -- 零填充约束
2  create table t_user12(
3  id int zerofill, -- zerofill默认为int(10) 当插入字段的值小于定义的长度时 会在该值
   的前面补上相应的0
4  name varchar(20)
5  );
6
7  +-----+-----+-----+-----+
8  | Field | Type                | Null | Key | Default | Extra |
9  +-----+-----+-----+-----+
10 | id    | int(10) unsigned zerofill | YES  |     | NULL    |      |
11 | name  | varchar(20)           | YES  |     | NULL    |      |
12 +-----+-----+-----+-----+
13
```

DOS

```
1  mysql> show databases;
2  +-----+
3  | Database |
4  +-----+
5  | database1 |
6  | information_schema |
7  | mysql |
8  | performance_schema |
9  | sys |
10 +-----+
11 5 rows in set (0.00 sec)
12
13 mysql> use database1
14 Database changed
15 mysql> show tables;
16 +-----+
17 | Tables_in_database1 |
18 +-----+
19 | student |
20 +-----+
21 1 row in set (0.00 sec)
22
23 mysql> desc student;
24 +-----+-----+-----+-----+-----+-----+
25 | Field | Type        | Null | Key | Default | Extra          |
26 +-----+-----+-----+-----+-----+-----+
27 | sid   | int         | NO   | PRI | NULL    | auto_increment |
28 | name  | varchar(20) | YES  | UNI | NULL    |                |
29 | age   | int         | NO   |     | NULL    |                |
30 +-----+-----+-----+-----+-----+-----+
31
32
```

