DDL Data Definition Language 数据定义语言

DML Data Manipulation Language 数据操纵语言

DCL Data Control Language 数据控制语言

DQL Data Query Language 数据查询语言

# 一、知识点

## 1、mysql常用数据类型：

整数：int 、smallint、bigint、tinyint

浮点数;float、double

字符串：char(length)（定长字符串）、varchar(length)（变长字符串，最大长度是65535）

日期：date、datetime、time、timestamp（时间戳类型，年月日时分秒）

Datatime：如果不赋值，值为空；

Timestamp：如果不赋值，使用默认的系统时间；

大数据：text（字符型）、blob（字节型）

bigint：从 -2^63 (-9223372036854775808) 到 2^63-1 (9223372036854775807) 的整型数据（所有数字）。存储大小为 8 个字节。

int:从 -2^31 (-2,147,483,648) 到 2^31 – 1 (2,147,483,647) 的整型数据（所有数字）。存储大小为 4 个字节。int 的 SQL-92 同义字为 integer。

smallint:从 -2^15 (-32,768) 到 2^15 – 1 (32,767) 的整型数据。存储大小为 2 个字节。

tinyint:从 0 到 255 的整型数据。存储大小为 1 字节。

## 2、字符集校对collate;

big5 big5\_chinese\_ci

gbk gbk\_chinese\_ci

utf8 utf8\_general\_ci

## 3、where常用运算符

在where子句中经常使用的运算符：

### 1、Like

like语句中，% 代表零个或多个任意字符，\_ 代表一个字符，例如first\_name like ‘\_a%’;支持转意字符。

例如查找包含%：like “%\%%”escape “\”（标准sql语句）

在mysql中like “%\%%”。或者 like “%#%%”escape “#”。但是mysql不支持like “%\%%”escape “\”。

### 2、注意点s

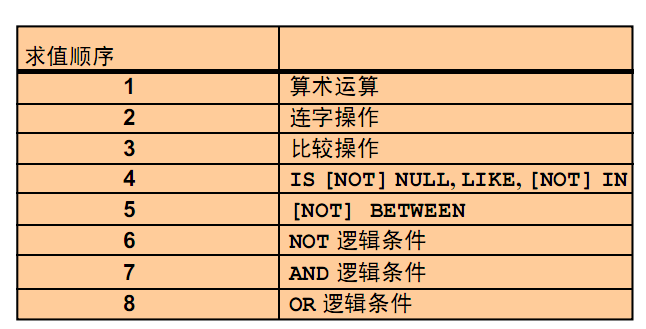
and优先级高于or

空值（null）参与运算结果为空。聚合函数除外。

例如：null+10=null，不是10

|  |  |  |
| --- | --- | --- |
| 比较运算符 | > < <= >= = <>（!=） | 大于、小于、大于(小于)等于、不等于(不等于) |
| between……and…… | 显示在某一区间的值，例如：WHERE column\_name BETWEEN value1 AND value2(包含头尾) |
| in | 显示在in列表中的值，例：in(100,200) |
| like | 模糊查询，字符可以连续使用  %：匹配多个任意字符  \_：匹配一个任意字符 |
| Is null（is not null） | 判断为空（不为空） |
| 逻辑运算符 | and | 多个条件同时成立 |
| or | 多个条件任一成立 |
| not | 不成立，例：where not in(1,5,2,8); |

### 3、运算优先级



# 4、执行语法顺序：

Select------from------where------group by------having------order by------limit

# 二、对数据库的操作

dos进入数据库：

1、创建

Create database 数据库名称：

2、删除

drop database 数据库名称;

例如：drop database users;

3、查看

（1）显示所有数据库：show databases;

（2）show create database 数据库名称;

显示数据库的相关信息，编码等

（3）显示当前使用的数据库：select database();

4、切换：use 数据库名称;

5、修改（字符校验和编码方式）

alter database 数据库名称 character set gbk collate gbk\_chinese\_ci;

字符校验：

big5 big5\_chinese\_ci

gbk gbk\_chinese\_ci

utf8 utf8\_general\_ci

# 三、对表的操作

1. 创建

create table 表名(列名1 数据类型,列名2 数据类型,列名3 数据类型……); -- 如果有同名不能创建

例如：create table \_category(id int primary key auto\_increment, name varchar(50) not null, descrtiption varchar(200));

replace table 表名(列名1 数据类型,列名2 数据类型,列名3 数据类型……); --覆盖同名表

create or replace table 表名(列名1 数据类型,列名2 数据类型,列名3 数据类型……); --如果有同名表覆盖，没有则创建

1. 查看

（1）desc 表名;

或者：describe 表名;

（2）显示所有表： show tables；

3、删除

drop table 表名;

4、重命名表名

rename table 旧表名 to 新表名;

1. 修改列数据

mysql每次只能修改一列，不能加括号。

1. 增加

alter table 表名 add 列名 数据类型

1. 修改

修改列名：alter table 表名 change 旧列名 新列名 数据类型;

修改列数据类型：alter table 表名 modify 列名 数据类型

1. 删除

alter table 表名 drop 列名;

1. 修改字符集：

alter table 表名 character set utf8;

# 四、对数据的操作

## DML部分

Data Manipulation Language 数据操纵语言

### 1 、insert

（1）指定列插入数据

Insert into 表名 (列1，列2……) values (值1，值2……);

（2）所有列插入数据

Insert into 表名 values（值1，值2……值last）;

必须按照表中列的顺序，主键即使是自动增长，也不能省略

例如：insert into student values(20040002,'刘德华',18631100000,'外语系',23,'南京');

### 2、update

（1）统一修改表中某列的值

UPDATE 表名称 SET 列名称 = 新值；所有行该列的值都会被修改

（2）统一修改表中多列的值

UPDATE 表名称 SET 列1 = 新值1，列2 = 新值2……；

（3）带条件的更新列值

UPDATE 表名称 SET 列名称 = 新值 WHERE……;

### 3、delete

（1）删除表中全部数据

Delete from 表名；

（2）带条件的删除

Delete from 表名 [where ……]；

例如： Delete from users where id in（1,2,3,4,7）；id包含1,2,3,4,7的被删除.

**注意：**mysql中的字符串也不区分大小写。例如：delete from employee where name="XIAOQIANG";数据库中name=”xiaoqiang”的也会被删除。

### 4、truncate

截断表，删除表中所有数据，并且自增长的列重新计数，效率比delete高。

格式：Truncate 表名;

## DQL部分

Data Query Language 数据查询语言

### select

（1）查询所有列

Select \* from 表名;

（2）查询指定列

Select 列1,列2…… from 表名;

可以使用列名做字段运算：



select id,name,(chinese+english+math) as "总分" from student;

结果：



### as

select 列1 as columnname\_other from 表名;

例如：select id,name,salary,avg(salary) as "平均工资" FROM employee GROUP BY name HAVING AVG(salary)>99 ORDER BY 平均工资,id;

注意：as后的别名可以当成列名直接使用.where除外，order by和group by均可。

### 3、distinct

去除重复。当显示多列去重复时，只有当显示的列都相同才会去重复，否则无效。

select distinct 列名 from 表名;

可以在\*前使用：select distinct \* from 表名;

多行去重复：select distinct 列1，列2 from 表名;

注意：只有列1和列2都一样才会有效果。

### 4、order by

（1）按照单列排序

Select 列1,列2,……from 表名 order by 列1;

（2）按照多列排序

Select 列1,列2,……from 表名 order by 列1,列2;

（3）指定顺序

升序asc；倒序 desc。默认asc

Select 列1,列2,……from 表名 order by 列1 desc;

（4）多列指定顺序

Select 列1,列2,……from 表名 order by 列1 desc,列2 desc;

（5）按照列号排序

Select 列1,列2,……from 表名 order by **2** desc;

Order by后面加数字，指的是按照select后面显示的第几列排序。

### 5、group by

（1）按照单列分组

Select 列1 from 表名 group by 列1

（2）按照多列分组

Select 列1，列2 from 表名 group by 列1，列2

注意：select后面的列一定要都出现在group by 后面。出现在group by后面的不一定在select后面写出。聚合函数除外。

例如：Select 列1，列2 from 表名 group by 列1;是错误的。

Where后面不能跟聚合函数，如果有需要使用having;

### 6、limit

显示指定的行数，放在where的后面。

Select 列1,列2…… from 表名 limit 3,5;

注意：limit后面第一个数字3表示从第四行开始，5表示显示后面的五行（不是数到第5行）。编号是从0开始的。

## 函数

### 单行函数：

一个输入一个输出，多个输入多个输出



Select concat(“asd”,”qwe”,”zxc”) from dual;

dual是一个虚拟表。结果asdqwezxc。在mysql中也可以Select concat(“asd”,”qwe”,”zxc”);

### 聚合函数（组函数）：

聚合函数忽略空值，例如count()不计空值；avg:(有数据的值之和)/有数据的行数

• AVG 平均值

遇到空值，也想计算在内。例如salary有空值，考虑使用avg(ifnull(salary,0));-----🡪这里空值为0

• COUNT 计数

• MAX 最大值

• MIN 最小值

• STDDEV 标准差

• SUM 合计

• VARIANCE 方差

## 多表查询

### 1、关联关系：

（1）一对多、多对一

例如，员工（employee）和部门（department）

多方（employee）建立外键 参照一方（department）的主键。

（2）多对多

例如：学生（student）和课程（course）

添加一个中间关系表sc（包含source\_id和student\_id两列），source\_id和student\_id联合做主键。

sc表做两个外键：source\_id做外键参照source的id列，student\_id做外键参照student的id列。

（3）一对一

1 主键的自然匹配

2 唯一外键（建立其中一个表的外键，并把该列设置为唯一）

### 2、多表连接查询

（1）内连接：等值连接、非等值连接、自连接

（2）外连接：左外链接、右外连接、全外连接（mysql不支持）

### 3、多表sql语句

Employee和departm两张表，employee表外键department\_id匹配department的id列。

#### （1）等值连接sql

select employee.id,employee.name,department.name from employee,department where employee.departm\_id=department.id;

select employee.id,employee.name,department.name from employee,department where employee.departm\_id=department.id and employee.name=”xiaoqiang”;

可以使用表的别名(employee e,department d):

select e.id,e.name,d.name from employee e, department d where e.departm\_id=d.id;

#### （2）自连接语句

select e1.name,e1.salary from employee e1,employee e2 where e2.name="小强" and e1.salary>e2.salary;

#### （3）左外、右外连接

左外连接：表1 left outer join 表2 on 条件

表1全部显示出来，表2显示和表1有关连的数据

select e.id,e.name,d.name from employee e left outer join department d on e.departm\_id=d.id;

右外连接：表1 right outer join 表2 on 条件

表2全部显示出来，表1显示和表2有关连的数据

全外连接：表1 full outer join 表2 on 条件（mysql不支持）

在on后面等值条件的基础上，再把左右两个表未匹配的内容填补进来

### 子查询

select e.id,e.name,e.salary,d.name from employee e,department d where e.departm\_id=d.id and salary>(select salary from employee where name="小强");

子查询也叫内查询：

放在小括号里面；

放在运算符的右面（mysql可以放左面）；

子查询不要使用order by（没有意义）；

在单行子查询中用单行运算符，在多行子查询中用多行运算符

单行子查询：子查询语句返回一条结果

例如：select id,name,salary from employee where salary<(select salary from employee where id=2);

在上面的语句中，子句（绿色部分）返回的结果只有一条，所以叫单行子查询。主查询的where要使用单行运算符（< = > != 等）。

多行子查询：子查询语句返回多条结果（多行运算符：in、any、all;）

例如：select id,name,salary from employee where salary < any(select salary from employee where id>2);

在上面的语句中，子句（绿色部分）返回的结果有多条，所以叫多行子查询。主查询的where要使用多行运算符（any、all等）。

in 直接使用：

select name,age from users where age in(select name,age from users where age>10);

any和all配合单行运算符使用：

select name,age from users where age> any (select name,age from users where age>10);

//上面语句中age只要大于最小值即可

select name,age from users where age> all (select name,age from users where age>10);

//上面语句中age需要大于最大值

# 五、视图

由一个或多个表的部分数据组成的对象。

作用：

1、限制数据的访问权限（安全性）

2、使查询变得容易（可以把复杂的查询结果存到视图里面）

如果视图中包含下面的部分就不能通过视图添加数据：

• 组函数

• GROUP BY 子句

• DISTINCT 关键字

## 1、创建视图

格式：create view 视图名 as 查询数据

例如：

create view emp\_v as select id,name,salary from employee;

## 2、查询视图

Select \* from emp\_v;

## 3、修改视图

引用表和视图内的数据都会改变：

insert into emp\_v(id,name) values(19,"wangcai1");

## 4、内建视图

内建视图可以有多表。

例如，绿色部分：

SELECT t.id,t.name,t.salary from(select \* from employee where salary>1200) t;

# 六、约束

## 1、约束种类

主键约束；

外键约束；

唯一性约束；

非空约束；

check约束（mysql不支持）；

## 2、数据库完整性

实体完整性；（主键）

参照完整性；（外键）

用户自定义完整性；（check）

## 3、约束语句

– NOT NULL 非空

– UNIQUE 唯一

– PRIMARY KEY 主键

– FOREIGN KEY 外键

– CHECK check约束

AUTO\_INCREMENT 自动增长

要让 AUTO\_INCREMENT 序列以其他的值(100)起始，请使用下列 SQL 语法：

ALTER TABLE 表名 AUTO\_INCREMENT=100;

not null 在MySQL中必须在列类型后面指定，使用constraint无效。

1. 创建表时添加约束

Constraint使用：

Constraint 约束名称（自己设定） 约束关键字（列名）

constraint约束名称（自己设定）foreign key (外键列名) REFERENCES 表名(外键参照的列名)

例1：

create table users(

id int primary key AUTO\_INCREMENT,

name varchar(255) not null,

birthday date,

email varchar(255) unique

)

例2：使用constraint

create table users(

id int ,

name varchar(255) not null,

birthday date,

email varchar(255),

class\_id int,

constraint user\_pk primary key(id),

constraint user\_email\_unique unique(email),

constraint user\_fk1 foreign key (class\_id) REFERENCES class (id)

)

1. 在原表上添加约束

alter table 表名 add constraint约束

例如：

alter table users add constraint users\_email\_unique(email);

1. 删除约束

alter table 表名 drop constraint约束

例如：

ALTER TABLE Persons

DROP CONSTRAINT uc\_PersonID

# 七、数据库的备份和恢复

例如在C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin目录下，执行如下命令行

导出：mysqldump –u 数据库登录用户名 –p 数据库名称 > 绝对路径.导出文件名.sql

例如：mysqldump –u root –p users > d:\ddl.sql

导入：mysql –u 数据库登录用户名 –p 数据库名称 < sql文件所在的绝对路径

例如：mysql –u root –p users < d:\ddl.sql

工具导出的sql文件中包含自动创建数据库的语句，直接导入就行。

语句导出的sql文件不包含自动创建数据库语句，需要手动创建数据库，然后再导入。