数据库操作语句总结(插入：insert, 删除：delete, 更新：update, 查询：select)

Mysql语句后必须添加；符号，否则不执行；

安装mysql服务 ：mysqld –install 启动mysql服务 ：net start mysql 关闭mysql服务：net stop mysql

卸载mysql：mysqld -remove

登录mysql数据库： mysql -u用户名 -p用户密码 退出mysql数据库：exit; 或者 quit;

查看所有数据库 show databases;

查看当前使用的数据库名 select database();

创建数据库 create database 库名；

创建数据库时指定编码方式 create database 库名 character set 编码方式；

创建数据库后修改编码方式 alter database 库名 default character

ste 编码方式 collate 编码方式\_bin;

查看已创建的数据库信息 show create database 库名；

使用数据库 use 库名；

删除数据库 drop database 库名

查看所有的数据表 show tables；

查看单个数据表的信息 show create table 表名 \G;

只查看表中列的相关信息 desc 表名；

创建数据表 create table 表名（

字段名1 数据类型,

字段名2 数据类型,

…,

字段名n 数据类型

);

修改表名 alter table 原表名 rename 新表名；

删除数据表 drop table 表名；

修改字段 alter table 表名 change 原字段名 新字段名 新数据类型；

例：（ALTER TABLE student CHANGE stu\_id stu\_sex VARCHAR(10);

修改字段的数据类型 alter table 表名 modify 字段名 数据类型；

修改字段1至第一位 alter table 表名 modify 字段名1 数据类型 first；

将字段1插到字段2后 alter table 表名 modify 字段名1 数据类型 after字段名2；

添加字段 alter table 表名 add 新字段名 数据类型；

删除字段 alter table 表名 drop 字段名；

添加主键约束 alter table 表名 add primary key(列名); --当列中已存在相同数据时，无法设置主键

添加外键约束 alter table 表名 add foreign key(外键字段名) references 主表表明(主键字段名);

删除外键约束 alter table 表名 drop foreign key(外键名)；

添加唯一约束 alter table 表名 add unique(列名);

非空约束 alter table 表名 modify 字段名 数据类型 not null;

默认值约束 alter table 表名 modify 字段名 数据类型 default 默认值;

设置自动增长列 alter table 表名 modify 字段名 数据类型 primary key auto\_ increment;

为字段创建普通索引 create index 索引名 on 表名(字段名[(长度)]);

为字段创建唯一索引 create unique index 索引名 on 表名(字段名[(长度)]);

删除索引： alter table 表名 drop index 索引名; 或 drop index索引名 on 表名;

为所有字段插入数据 insert into 表名 values(值1,值2，……); --值的顺序必须与字段顺序一致

为指定字段插入数据 insert into 表名(字段名1,字段名2，……) values(值1,值2,……); --顺序前后一致即可

为所有字段批量插入数据 insert into 表名 values(值1,值2,……),……,(值1,值2……);

为指定字段批量插入数据 insert into 表名(字段名1,字段名2,……) values(值1,值2,……),……,(值1,值2……);

更新字段的全部数据 update 表名 set 字段名1=值1 [,字段名2=值2,……];

更新字段的部分数据 update 表名 set 字段名1=值1 [,字段名2=值2,……] where 条件表达式.； --条件表达式例:name=‘lihua’;

删除表中所有数据 delete from 表名； --dml数据操纵语言语句

删除表中所有数据 truncate table 表名； --ddl数据描述语言

删除表中符合条件的数据 delete from 表名 where 条件表达式； --条件表达式例如:gender=‘female‘ and salary<800;

数据的完整性

创建时跟在表的字段数据类型后：

唯一约束： unique, --确保数据的唯一性,可以取空值null，但不能重复。

非空约束： not null, --确保该字段数据不为空值

默认值约束： default 默认值 --在不插入数据时的默认数据值

自动增长列： primary key auto\_increment, --自增字段的类型必须是整型，且必须是主键，否则报错

主键约束：主键的列不能有重复，且不能为空值null，一个表可以有多个主键。

创建表时添加主键两种方式： create table 表名（

字段名1 数据类型 primary key,

……，

字段名n 数据类型,

primary key (字段名1,字段名2…)

);

外键约束(引用完整性): 实体之间关系的描述 若要删除主表中的数据，必须先删除所有引用它的从表的数据

(外键是指引用另一个表的一列或多列，被引用的列应是主键约束或唯一约束;主表中没有数据,从表就无法插入数据(引用错误)

create table 表名(

字段名 数据类型,

……,

foreign key(外键字段名) references 主表表名(主键字段名)

);

创建索引： create table 表名（

字段名 数据类型,

……,

**[**unique**/**fulltext**/**spatial**]** index**/**key **[**索引名**]**(字段名[(长度)] **[**asc**/**desc**]** )

);

说明：

（1）unique：表示索引为唯一索引。 --保证数据的唯一性，拒绝插入重复的数据  
（2）FULLTEXT：表示索引为全文索引。  
（3）SPATIAL：表示索引为空间索引。  
（4）index 和 key：为普通索引,两者选择其一,作用相同,加快对数据的访问速度,索引名是可选值,长度用于表示索引的长度  
（5）索引名：给创建的索引取一个新名称。如果不指定则采用字段名作为索引名。  
（6）字段名：指定索引对应的字段的名称。  
（7）长度：指索引的长度，字符串类型才可以使用。  
（8）asc：表示升序排列。 desc：表示降序排列。

查询语句：

第一行： 单表查询：

查看表中所有数据 select \* from 表名;

查看表中指定数据 select 字段名1,字段名2,……,字段名n from 表名; --显示顺序与指定字段顺序一致

查看数据并去除重复数据 select distinct字段名 from 表名; --只显示单列(指定的字段列)

聚合函数：可为返回列起别名(as可省略) 例如: select conut (\*|1|列名) [as] 别名 from 表名;

ifnull(列名,0),当值为空(null)时自动替换为0 例如: where 列名1+ifnull(列名2,0) 条件语句;

--若不使用ifnull时null值和其它数的任何运算结果都为0

返回某列的行数 select conut (\*|1|列名) [as] 别名 from 表名

-- \*和1:返回表的行数 列名:返回指定列的行数不包含null(空值)

select conut (\*|1|列名1) conut (\*|1|列名2)…… from 表名;

--可同时返回多个行数

返回某列或多列值的和 select sum(字段名1+…+字段名n) from 表名; --如果不是数值类型返回0

select sum(字段名1),…,(字段名n) from 表名;

返回某列的平均值 select avg(字段名) from 表名; --如果不是数值类型返回0

返回某列的最大值 select max(字段名) from 表名; --如果指定字符串类型,则使用字符排序运算

返回某列的最小值 select min(字段名) from 表名; --如果指定字符串类型,则使用字符排序运算

多表查询： --要求多个表的列数必须一致，且以第一个表的字段名为标准；

使用union关键字合并 select \* from 表名1 union select \* from 表名2; --合并结果集并过滤掉重复数据

使用union all 关键字合并 select \* from 表名1 union all select \* from 表名2; --合并结果集,保留重复数据

交叉查询(内连接)： select 查询字段 from 表1 cross join 表2; --笛卡尔积在sql中实现方式是交叉连接,每行数据任意组合

内连接(标准): select 查询字段 from 表1 [inner] join 表2 on 表1.关系字段=表2.关系字段 where 查询条件;

--内连接指:返回值只包含符合条件的数据 --inner可省略，mysql默认的连接方式就是内连接; on用来指定连接条件;

--表1若为emp, 此语句中表1写为emp e, 则表示为emp起别名e; 可用e.字段名 表示 表1.关系字段;

外连接: --还包含没有关联的数据；

左外连接: --以左表数据为基准,若左表有数据而右表没有的数据,右表的数据显示为空(null)

select 查询字段 from 表1 left [outer] join 表2 on表1.关系字段=表2.关系字段 where 查询条件;

右外连接: --以右表数据为基准,若右表有数据而左表没有的数据,左表的数据显示为空(null)

Select 查询字段 from 表1 right [outer] join 表2 on 表1.关系字段=表2.关系字段 where 查询条件;

多表查询: --内/外连接仅限于两个表之间的查询, --多个表通过join关键字连接,on关键字后面是表与表之间的关系字段;

select 查询字段 from 表1 [别名] join 表2 [别名] on表1.关系字段=表2.关系字段 join 表m on……;

自然连接: --自动匹配表与表之间列名和数据类型相同的数据,不需要指定连接字段,默认内连接方式查询;

select 查询字段 from 表1 [别名] natural [left/right] join 表2 [别名]; --可以指定左/右连接

自连接: select 查询字段 from 表1 [别名],表名 [别名] where 查询条件;

子查询(嵌套查询): --即select 中包含select 查询条件；

作为查询条件: --select子查询在where关键字后，先执行子查询将结果作为外层查询条件执行;

例子: select \* from emp where sal>(select sal from emp where ename=’lihua’); --查询所有工资高于lihua的员工信息:

子查询作为表: --放在from后面, 先执行子查询将结果作为外层查询条件执行;

例子: select e.name,e.sal,d.dame,d.loc from emp e,(select dame,loc,deptno. from dept) d

where e.deptno=d.deptno and e.empno=7788;

第二行; 查看数据条件

= 等于 != 不等于(部分数据库不支持) <> 不等于

< 小于 <= 小于等于 > 大于 >= 大于等于

And：需要同时满足多个查询条件之间用and连接 交集∩ --where 条件1 and 条件2;

Or：只需满足其中之一用or连接 并集∪ --where 条件1 or 条件2;

(not) In： 筛选出(不包含)包含in后内容的数据行 --where 字段名 (not) in (元素1,元素2,元素3… );

Is (not) null：筛选出指定字段是(不是)’空值’的数据行 --where 字段名 is (not) null;

(not) Between and: 筛选出值在(不在)范围内的数据行 --where 字段名 (not) between 值1 and 值2;

模糊查询： --where 字段名 (not) like ‘匹配字符串’;

%：百分号表示任意0~n个任意字符

\_:下划线表示一个任意字符

排序查询： order by 字段名1[asc|desc],字段名2[asc|desc]……;

asc升序 desc降序 不写默认升序 (多个字段约束时按先后，当上一个相等时按下一个)

分组查询: group by 字段名1,字段名2，…[having 条件表达式]; --通常与聚合函数一起使用

-- Where在分组前进行数据过滤不可出现在分组查询后; Having在分组后进行数据过滤

分页查询： limit [m,]n; --m=(要显示第几页-1)\*每页分多少个,默认为0, n代表从m+1条记录开始取n条记录；

**Select+常用函数;**

表名设置中文会出现其他字；

字段名设置中文可能会出现错误；