

API（Application Programming Interface,应用程序编程接口）是一些预先定义的函数

## RDBMS 术语

* **主键**：主键是唯一的。一个数据表中只能包含一个主键。你可以使用主键来查询数据。
* **索引：**使用索引可快速访问数据库表中的特定信息。索引是对数据库表中一列或多列的值进行排序的一种结构。类似于书籍的目录。
* **数据库:** 数据库是一些关联表的集合。
* **数据表:** 表是数据的矩阵。在一个数据库中的表看起来像一个简单的电子表格。

## 登录 MySQL

当 MySQL 服务已经运行时, 我们可以通过 MySQL 自带的客户端工具登录到 MySQL 数据库中, 首先打开命令提示符, 输入以下格式的命名:

mysql -h 主机名 -u 用户名 -p

## 启动及关闭 MySQL 服务器

**关闭MySQL服务器:**

1、**通过 "服务" 管理工具：**同样打开"运行"对话框，输入 services.msc，找到 "MySQL" 服务，右击选择"停止"。

mysql客户端连接:

1. 在系统菜单中,找到mysql-unicode,输入密码
2. 在命令提示符中

mysql数据库的sql语句不区分大小写,关键字建议大写

SQL分类:

1. DDL:数据定义语言,用来定义数据库对象(数据库,表,字段)
2. DML:数据操作语言,用来对数据库表中的数据进行增删改
3. DQL:数据查询语言,用来查询数据库中表的记录
4. DCL:数据控制语言,用来创建数据库用户,控制数据库的访问权限

DDL:

查询所有数据库:SHOW DATEBASES;

查询当前数据库:SELECT DATEBASE();

创建:CREATE DATABASE [IF NOT EXISTS] //[ ]指的是数据库名存在创建,不存在不创建 [ DEFAULT CHARSET字符集] //比如UTF8mb4(四字节) [COLLATE 排序规则]

删除:drop database if exists数据库名;

使用: use 数据库名;

DDL-表查询-查询:

1. 查询当前数据库所有表:show tables;
2. 查询表结构:desc 表名;
3. 查询指定表的建表语句:show create table 表名;

DDL-表操作-创建

Create table 表名(

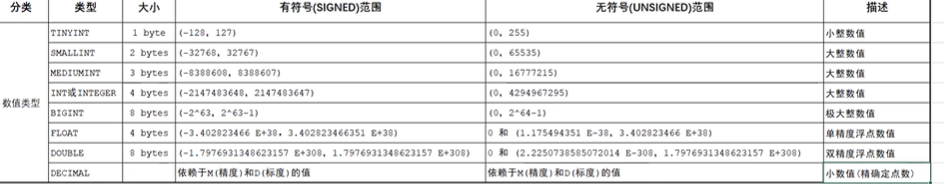
字段1 字段1类型 [comment 字段1注释 ] , ////有逗号

............

字段n 字段n类型 [comment 字段n注释 ] ///无逗号

)[ comment 表注释];

DDL-表操作-数据类型



Decimal依赖M(精度)和D(标度)的值 例如123.45 精度5;标度2;

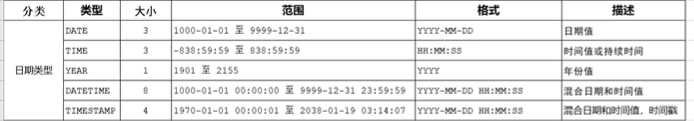
Eg: age TINYINT UNSIGNED //取值取无符号范围

Eg: score double(4,1) ///4代表整体长度,1代表位数

字符串类型: char (10) ///10定义字符串字节



日期类型:



DDL-表修改-修改

添加字段: alter tables 表名 add 字段名 类型(长度) [ comment注释] [约束];

修改数据类型: alter tables 表名 modify 字段名 新数据类型(长度);

修改字段名和字段类型:

alter table 表名 change 旧字段名 新字段名 类型(长度) [comment注释] [约束];

DDL-表操作-修改

删除字段:alter table 表名 drop 字段名;

修改表名:alter table 表名 rename to 新表名;

DDL-表操作-删除

删除表: drop table [ IF EXISTS]表名;

删除指定表,并重新创建该表: truncate table 表名; ////数据已经被删除

DML-表操作-添加数据

1. 给指定字段添加数据:insert into 表名 (字段名1,字段名2....) values(值1,值2....);
2. 给全部字段添加数据:insert into 表名 values (值1,值2...);
3. 批量添加数据

insert into 表名 (字段名1,字段名2....) values(值1,值2....),(值1,值2....),(值1,值2....);

insert into 表名 values (值1,值2...),(值1,值2...),(值1,值2...);

注意:1.插入数据时,指定的字段顺序需要与值的顺序是一一对应的

字符串和日期型数据应该包含在引号内

插入的数据大小,应该在字段的规定范围内

DML-修改数据

Update 表名 set 字段名1=值1,字段名2=值2,....[where 条件];

注意:修改语句的条件可以有,也可以没有,如果没有条件,则会修改整张表的所有数据

DML-删除数据

Delete from 表名 [where 条件];

注意:  
1.delete 语句的条件可以有,也可以没有,如果没有条件,则会删除整张表的所有数据;

2.delete语句不能删除某一个字段的值(可以使用UPDATE)

DQL-基本查询

1. 查询多个字段

Select 字段1,字段2,字段3...from

Select\*from 表名; /////查询所有字段,尽量不使用,不直观///

1. 设置别名:select 字段1 [AS 别名1],字段2[AS 别名2].....from 表名;
2. 去除重复记录

Select distinct 字段列表 from 表名;

DQL-条件查询:



Select\*from 表名 where 字段名 is NULL; //无信息

Select\*from 表名 where 字段名 is not NULL; //排除无信息

Select\*from 表名 where 字段名 in(18,20,40....); //多个信息中满足任一

Select\*from 表名 where 字段名 like ‘\_ \_’; //查询两个字符的信息

Select\*from 表名 where 字段名 like ‘%X’; //查询最后一个字符为X的信息

DQL-聚合函数

介绍:将一列数据作为一个整体,纵向计算(列)

常见聚合函数:  
1.count //统计数量; 2.max //最大值 3.min //最小值 4.avg //平均值 5.sum //求和

语法: SELECT 聚合函数(句段列表) from 表名;

//NULL值不参与所有聚合函数运算

DQL-分组查询

语法:

EG1:查询年龄小于45的员工,并更具工作地址分组,获取员工数量大于等于3的工作地址

Select workaddress, count(\*) address\_count from emp where age <45group by workaddress having address\_count > = 3;

select 字段列表 聚合函数 from 表名[WHERE 条件] group by 分组字段名 [having 分组后过滤条件]

where与having区别:

执行时机不同:where是分组之前进行过滤,不满足where 条件,不参与分组;而having是分组之后对结果进行过滤

判断条件不同:where不能对聚合函数进行判断,而having可以

注意: 执行顺序:where > 聚合 > having

分组之后,查询的字段一般为聚合函数和分组字段,查询其他字段无任何意义

DQL--排序查询

1. 语法:

Select 字段列表 from 表名 order by 字段1 排序方式1,字段2,排序方式 2;

1. 排序方式

ASC:升序(默认值)

DESC:降序

注意:如果是多字段排序,当地一个字段值相同时,才会根据第二个字段进行排序

DQL--分页查询

1. 语法:

Select 字段列表 from 表名 limit 起始索引,查询记录数 ;

注意:

1. 起始索引从0开始,起始索引=(查询页码-1)\*每页显示记录数
2. 分页查询是数据库的方言,不同的数据库有不同的实现,MYSQL中是LIMIT
3. 如果查询的是第一页数据,其实索引可以省略,直接简写为limit 10

案例

统计员工表中,年龄小于60岁的,男性员工和女性员工的人数

Select gender,count(\*) from emp where age <60 group by gender ;

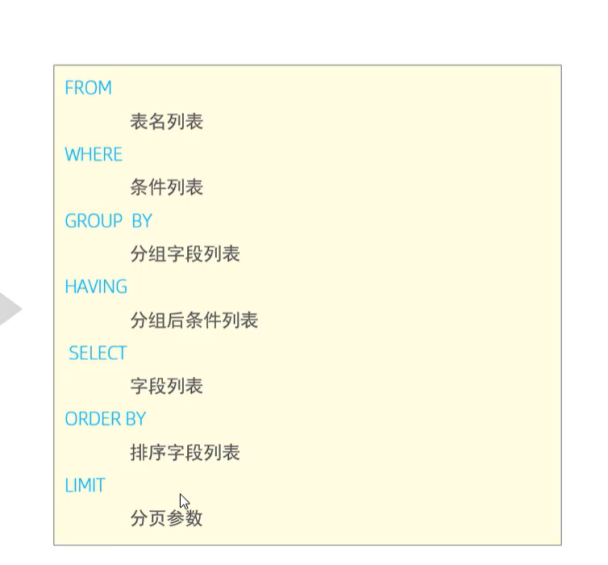
查询所有年龄小于等于35岁员工的姓名和年龄,并对查询结果按年龄升序排序,如果年龄相同按入职时间降序排序

Select name,age from emp where age <= 35 order by age asc,entrydate desc;

查询性别为男,且年龄在20-40岁(含)以内的前五个员工信息,对查询的结果按年龄升序排序,年龄相同按入职时间升序排序

Select \*from where gender ‘男’ and age between20 and 40 order by age asc, entrydate asc limit 5;

DQL-执行顺序



DCL-管理用户

1. 查询用户

Use mysql;

Select \*from user;

1. 创建用户

Create user ‘用户名’@’主机名’ identified by ‘密码’;

1. 修改用户密码

Alter user ‘用户名’@’主机名’ indentified by ‘密码’;  
4.删除用户

Drop user ‘用户名’@’主机名’;

注意:

1. 主机名可以使用%表示通配 2.这类SQL开发人员操作的比较少,主要是数据库管理员

DCL-权限控制



1. 查询限制:  
   show grants for ‘用户名’@’主机名’;
2. 授予权限:

Grant 权限列表 on 数据库名.表名 to ‘用户名’@’主机名’;

1. 撤销权限:

Revoke 权限列表 On 数据库名.表名 from ‘用户名’@’主机名’;

注意:1.多个权限之间,使用逗号分割

2.授权时,数据库名和表名可以使用\*进行通配,代表所有