

数据库

2021年9月6日 11:01

数据库分类

- 关系型数据库：采用关系模型来组织数据的数据库；关系模型指的就是二维表格模型
 - 例如：Oracle, Microsoft SQL Server, MySQL, SQLite
- 非关系型数据库：Not Only SQL，强调KEY-VALUE的方式存储数据
 - 例如：Mongodb, redis

MySQL

MySQL是一个关系型数据库管理系统软件。

关系型数据库管理系统（RDBMS）可以分为

- 1.关系型数据库服务端软件-server
- 2.关系型数据库客户端软件-client

SQL语言的分类

简写	语义	说明
DQL	数据查询语言	Select
DML	数据操作语言	Insert,update,delete
DDL	数据定义语言	Create,drop
TPL	事务处理语言	Begin transaction, commit, rollback
DCL	数据控制语言	Grant, revoke

MySQL数据类型

数据类型	说明
Int, bit	整数
decimal	小数
varchar（可变字符串）, char	字符串
Date, time, date time	日期时间
enum	枚举类型

Mysql字段约束

约束参数	说明
Primary key	主键约束
Not null	非空约束
unique	唯一约束
default	默认约束

Mysql数据库服务端启动

查看MySQL服务状态

```
sudo service mysql status
```

停止MySQL服务

```
sudo service mysql stop
```

启动MySQL服务

`sudo service mysql start`

重启

`sudo service mysql restart`

Mysql配置文件

路径: `cd /etc/mysql/mysql.conf.d/`

`Vim mysqld.cnf`

配置项介绍

- 1.port表示端口号, 默认3306
- 2.bind-address表示服务器绑定的ip, 默认为127.0.0.1
- 3.datadir表示数据库保存路径, 默认为/var/lib/mysql
- 4.log_error表示错误日志, 默认为 /var/log/mysql/error.log

Navicat基本使用 图形化

navicat建立删除数据库 新建删除表 设计表 以及数据库的CRUD

Mysql登陆客户端操作

- 1.连接mysql服务端指令
`mysql -uroot -p`
- 2.显示当前时间
`select now()`
- 3.退出连接
`exit/quit/crtl+d`

mysql数据库操作

- 1.查看所有数据库
`Show database;`
- 2.创建数据库
`Create database 数据库名 charset=utf8;`
- 3.使用数据库
`Use 数据库名`
- 4.查看当前使用的数据库
`Select database();`
- 5.删除数据库
`Drop database 数据库名`

mysql表操作

- 1.查看所有当前库中所有表
`Show tables;`
- 2.创建表
`Create table 表名(字段名称 数据类型 可选的约束条件,);`
- 3.修改表字段类型
`Alter table 表名 modify 列名 类型 约束;`
- 4.删除表
`Drop table 表名`
- 5.查看表结构
`Desc 表名;`

Mysql-CRUD操作

1.查询操作

1.查询列

Select * from 表名;

例子:

Select * from students;

2.查询指定列

Select 列1, 列2, From 表名;

例子

Select id,name from students;

3.distinct关键字: 只返回不同的值

Select name from info;

```
mysql> select name from info;
+-----+
| name |
+-----+
| 张三 |
| lilei |
| xiaoming |
| laotian |
| lily |
| hanmeiemi |
| yi |
| er |
| san |
| yi |
| er |
| san |
+-----+
12 rows in set (0.00 sec)
```

Select **distinct** name from info;

```
mysql> select distinct name from info;
+-----+
| name |
+-----+
| 张三 |
| lilei |
| xiaoming |
| laotian |
| lily |
| hanmeiemi |
| yi |
| er |
| san |
+-----+
9 rows in set (0.01 sec)
```

4.limit关键字: 返回前N行

Select * from info limit 5;

只返回表的前5行

select * from test LIMIT 3 OFFSET 1;

表示舍去一个后取两个

select* from test LIMIT 1,3;

当limit后面跟两个参数的时候, 第一个数表示要跳过的数量, 后一位表示要取的数量。

5.排序检索数据: order by 字句

Select * from info order by id;

将info表格按id升序排列

Select * from info order by id, age;

将info先按id排序, 再按age排序

Select * from info order by age desc;

将info按年龄降序排列

6.过滤数据: where子句 (注意where和order同时使用, order应在where之后)

- where操作符 =,<>!=,<,<=,>,>=,between (between 5 and 10, 包含5和10)
Select * from info where age=13;
将所有年龄为13的返回
- 多条件时使用and or连接, 且and优先级高于or, 搭配括号使用
Select * from info where age<13 and gender='男';
返回年龄小于13岁的男性
- 也可以用in操作符
Select * from info where age in (12,13,14)
选出年龄为12或13或14的
- NOT操作符, 在mysql中not支持对in, between和exists子句取反
Select * from info where age not in (12,13,14)
选出年龄不为12或13或14的

7.用通配符进行过滤

- Like 操作符
- %: %通配符表示任何字符出现任意次数, 比如shen%, 表示shen开头的任意字符串
- _ : _只匹配单个字符
例1: select id, name from info where name like 'y%' 找名字y开头的项目, 返回名字和id列
例2: select * from info where name like 'e_' 返回名字为er的

8.正则表达式匹配

- 使用regexp操作符 注: mysql中正则表达式不区分大小写, 要区分则使用regexp binary
- select * from info where 项 regexp '正则表达式'
- 例: Select * from 表 where name regexp '.il.'

▪ mysql正则表达式

or匹配: 1|2|3 相当于 [123] 相当于 [1-3]

特殊字符匹配: 用转义字符\ 例如匹配 '.' 则用 \. 匹配, 特别的 [\\f换页](#) [\\n换行](#) [\\r回车](#) [\\t制表](#) [\\v纵向制表](#)
其他正则表达式见书本地9章

9.创建计算字段

- Concat()拼接字段

Concat()函数用于拼接

例: select concat(name, '(', age, ')') from info order by age;

```
mysql> select concat(name, '(', age, ')') from info order by age;
+-----+
| concat(name, '(', age, ')') |
+-----+
| NULL                         |
| hejiajun(1)                  |
| yi(12)                       |
| er(13)                       |
| yi(13)                       |
| er(14)                       |
| lily(16)                     |
| hanmeiemi(17)                |
| 张三(18)                     |
| xiaoming(18)                 |
| san(20)                      |
| san(21)                      |
| lilei(28)                    |
+-----+
```

- AS使用别名

例: select concat(name, '(', age, ')') as name_age from info order by age;

```
mysql> select concat(name, '(', age, ')') as name_age from info order by age;
+-----+
| name_age |
+-----+
| NULL    |
| hejiajun(1) |
| yi(12)  |
| er(13)  |
| yi(13)  |
| er(14)  |
| lily(16) |
| hanmeiemi(17) |
| 张三(18) |
| xiaoming(18) |
| san(20)  |
| san(21)  |
| lilei(28) |
+-----+
13 rows in set (0.00 sec)
```

- 执行算术计算

Select id*age as id_age from info where age between 10 and 20;

2.增加数据

- 全列插入：值的顺序必须和字段顺序完全一致
Insert into 表名 values(...);
- 部分列插入：值的顺序和给出的列的顺序对应
Insert into 表名(列1...) values(值1...);
- 全列多行插入
Insert into 表名 values(...),(....),(...);
- 部分列多行插入
Insert into 表名(列1...) values(值1...),(值1...),(值1...);

3.修改数据

- Update 表名 set 列1=值1, 列2=值2... where 条件
例子：
Update students set age=18,gender='女' where id=6;

4.删除数据

- Delete from 表名 where 条件
例子：
Delete from students where id=5;

5.分组查询group by

```
Select emp_no,count(*) as t from employee
Group by emp_no
Having t>15;
```

分组查询与函数搭配使用，函数有avg, count, max, min, sum

Mysql联结表

- 内部联结
Select * from vendors,products where vendors.id=products.id;
创建联结两个表，用where规定如何关联
Select * from vendors **inner join** products **on** vendors.id=products.id;
与第一条相同
- 使用别名
Select p1.id,p1.name from products **as p1**,products **as p2** where p1.id=p2.id;
同一张表用两个别名也是允许的，但是对象不一样
- 外部联结
Select customer.cust_id, order.order_num
From customer **left outer join** orders
On customers.cust_id = orders.cust_id;
外部联结和内部联结不同在于，外部联结还包括没有关联行的行，left outer join是从左边表选择所有行，right则从右

查询优先级安排

Select	5
From	1
Where	2
Group by	3
Having	4
Order by	6
Limit	7

Msql数据库备份和恢复

Mysql数据-备份导出:

Mysqldump -u用户名 -p密码 数据库名字 表名字 >data.sql

Mysql数据-恢复导入:

Cd 到数据文件路径下

Msql -u用户名 -p密码

Use 数据库

Source data.sql