Mysql 备忘

- 编辑时间:2021年8月16日
- 最近的需求涉及到对数据库的操作比较多，所以在这里脑子里对mysql的印象加以总结，备忘

> 登录:

- mysql -uroot -p123456

> 数据类型:

- 大致可以分为三类：数值、日期/时间和字符串(字符)类型。

> 反引号的作用:

- 如果我们不加反引号，直接建立名称为select的表，因为select 默认是 Mysql的关键字，所以会报错。
- 如果我们加上反引号``就创建成功了。
- **所以为了安全起见可以在表名和字段名上都加上``。**

> 数据库:

- 打印所有数据库: `show databases;`
- 进入某个数据库：use （数据库名）; 例如：`use dlan`
- 创建数据库：`create database dlan;`
- 删除数据库：`drop database dlan;`
- 数据库备份：`mysqldump -uroot -p123456 database_name > ~/2021.2021_0528.sql`
- 数据库导入：`mysql -uroot -p123456 database_name < 2021_0528.sql`
- 数据库备份参考：https://www.cnblogs.com/letcafe/p/mysqlautodump.html

> 数据表:

- 进入数据库后，展示这个数据库下的所有表： `show tables;`
- 展示表结构：`desc table_name`
- 查询表：`select * form table_name`
- 删除表：`drop table all_usb_info`
- 修改表名：`rename table all_usb_info to all_usb`

> 对数据表的修改涉及到 1、表结构 和 2、表内容

- 1、对表结构的修改

```
主要是 : alter table table_name + 操作

1、添加字段  增加age列，在age后面添加这列的属性)
alter table t1 add age int;

2. 修改字段属性  修改age列的属性
alter table t1 modify age int not null default 20;

3.删除字段   删除age列
```

```
alter table t1 drop age;

4. 修改字段名    修改name列列名，最后必须跟上修改后列的数据类型
alter table t1 change name username varchar(30);

5. 给某列添加普通索引(主键索引一般建表时添加)
alter table t1 add index in_named(name);

6. 删除某列的普通索引
alter table t1 drop index in_named;
```

- 2、对表内容的修改

```
主要是 :
- 增: insert
- 删: delete
- 改: update
- 查: select

增 是直接增加了新的一行
删 改 查 用 where 确定具体的行，然后对其内容进行修改
删 改 查 where 后的条件可以通用

1、增加全新的一行
insert into t1(name) values("f");

2. 删除某行 主键自增是一直在自增，不管前面的记录有没有被删除
delete from t1 where id=6;

3. 改某行中的某个字段内容 用where确定行，注意: mysql中没有 == 只有 =
update t1 set username='g' where id=6;

4. 查询 这个比较复杂，需要展开来讲
select * from user where id=6
```

- 3、对数据表的查询 `select` 接上一部分
  - 这一部分比较复杂，整理需要时间，找到了很早之前的记录，点击查看
  - 单表查询
  - 多表查询
  - 左连接查询
  - 嵌套查询（省略，几乎不用）

select * from user where id=6
select [1、field_name] from [2、tabele_name] where [3、查询条件] [4、分组] [5、排序]

```
1、  field_name
     这里可以使用函数进行聚合 count sum avg min max
2、tabele_name
```

```
    单表查询只有一个表名，多表查询可以添加多个表
3、查询条件
    可以进行精准查询，也可以使用like进行模糊匹配，并且支持正则
4、分组
    使用group by 对结果进行分组，分组一定要搭配聚合使用,只分组不聚合没有意义。
5、排序
    对查询结果进行排序
```

- 多表查询
- 左连接查询
- 嵌套查询（省略，几乎不用）

> 创建数据表：

- 主键索引和非主键索引的区别
  https://www.cnblogs.com/heishuichenzhou/p/10813463.html
- mysql 详细建表与添加索引语句
  https://blog.csdn.net/justry_deng/article/details/81458470
- 主键索引一般建表的时候就要设置好
- 普通索引可以根据后期的使用情况，比如后期查询价格特别多，就可以给价格加上普通索引。

简约模板

```
CREATE TABLE projectfile (
    id INT  unsigned AUTO_INCREMENT COMMENT '附件id',
    f_name VARCHAR (512),
    fileurl VARCHAR (512) not null default "www.baidu.com",
    filesize BIGINT COMMENT ,
    PRIMARY KEY (id),
    FOREIGN KEY (projectid) REFERENCES project (id),
    UNIQUE INDEX (projectid),
    INDEX (fileuploadercode)
    INDEX (fileuploadercode,projectid)
) ENGINE = INNODB DEFAULT CHARSET = utf8 COMMENT '项目附件表';
```

完整模板

```
CREATE TABLE projectfile (
    id INT AUTO_INCREMENT COMMENT '附件id',
    fileuploadercode VARCHAR(128) ,
    projectid INT ,
    filename VARCHAR (512) ,
    fileurl VARCHAR (512) ,
    filesize BIGINT COMMENT ,
    -- 主键本身也是一种索引（注:也可以在上面的创建字段时使该字段主键自增）
    PRIMARY KEY (id),
    -- 主外键约束（注:project表中的id字段约束了此表中的projectid字段）
    FOREIGN KEY (projectid) REFERENCES project (id),
    -- 唯一索引 给projectid字段创建了唯一索引(注:也可以在上面的创建字段时使用unique来创
```

```
建唯一索引)
    UNIQUE INDEX (projectid),
    -- 普通索引 给fileuploadercode字段创建普通索引
    INDEX (fileuploadercode)
    -- 复合索引
    INDEX (fileuploadercode,projectid)
    -- 指定使用INNODB存储引擎(该引擎支持事务)、utf8字符编码
) ENGINE = INNODB DEFAULT CHARSET = utf8 COMMENT '项目附件表';
```