

4.软件测试领域知识点

2023年6月27日 15:28

一、软件测试基础知识点

1. 单元测试、模块测试、集成测试
2. 白盒测试、黑盒测试
3. 功能测试、性能测试（压力测试、负载测试）、安全性测试、兼容性测试(不同设备、不同操作系统、不同浏览器)、用户体验测试(界面美观易用友好) 等
4. 手动测试，自动化测试工具(MeterSphere或Python for Web API and Website)
5. 性能测试工具 (MeterSphere或JMeter)
6. Smoke冒烟测试、Regression回归测试、用户验收测试、产品发布验证
7. 设计测试用例（正向、反向、边界值等）
8. 测试覆盖率保证软件产品测试质量
9. 多个测试环境多轮测试（开发环境、测试环境、用户体验环境、真实产品环境）从低环境到高环境
10. 产品缺陷报告/Bug （标题、环境、测试数据、严重度、优先级、重现步骤、实际结果、期望结果）
11. 电脑端测试、移动端测试
12. 前端测试(for website, app)、后端测试 (for Web API, windows service, database)
13. 持续集成工具 (Jenkins, TeamCity) 定时执行自动化测试或版本构建自动触发自动化测试，生成测试报告
14. 测试用例及测试报告管理工具 MeterSphere
15. 缺陷报告/任务需求管理工具 Jira
16. 测试过程：项目需求评审、测试计划、测试用例、测试执行、分析测试结果、报告产品缺陷、修复再验证、高环境部署再测试、产品发布验证

二、测试对象：主要针对Website和Web Service(web API)以及Database进行测试

三、编程基础技能

面向对象的基本概念、常用的数据结构和算法、基本语句

四、数据库相关的基础知识

表的主键和索引的区别：主键一定是唯一性索引，唯一性索引不一定是主键；一个表中可以有多个唯一性索引，但只能有一个主键；主键列不允许空值，而唯一性索引允许空值。创建索引优点是可以提高查询的速度，缺点是会降低更新表的速度。

表的主键和外键差别：

基本的SQL语句：增删改查、Distinct、Join on、where、having、聚合函数、sum/average/count、min、max、

Mysql中having和where的区别:

having是在分组后对数据进行操作，该语句后面可以使用聚合函数

where是在分组前对数据进行操作，该语句后面不可以使用聚合函数

在Mysql查询过程中的执行顺序: from > where > group (含聚合函数) > having >

order > select

Distinct去重函数格式:

```
select
  distinct expression[,expression...]
from
  tableName
[where conditions];
```

expression: 列名

LIKE语句获取指定用户的信息:

查询以java字段为开头的信息

```
SELECT * FROM position WHERE nam
```

limit子句用于限制查询结果返回的数量，常用于分页查询 格式:

```
SELECT * FROM tableName LIMIT [offset,] rows | rows OFFSET offset
```

tableName: 表名

offset: 可选项，偏移量，指定了结果集的起始位置(从0开始)，为0时可省略

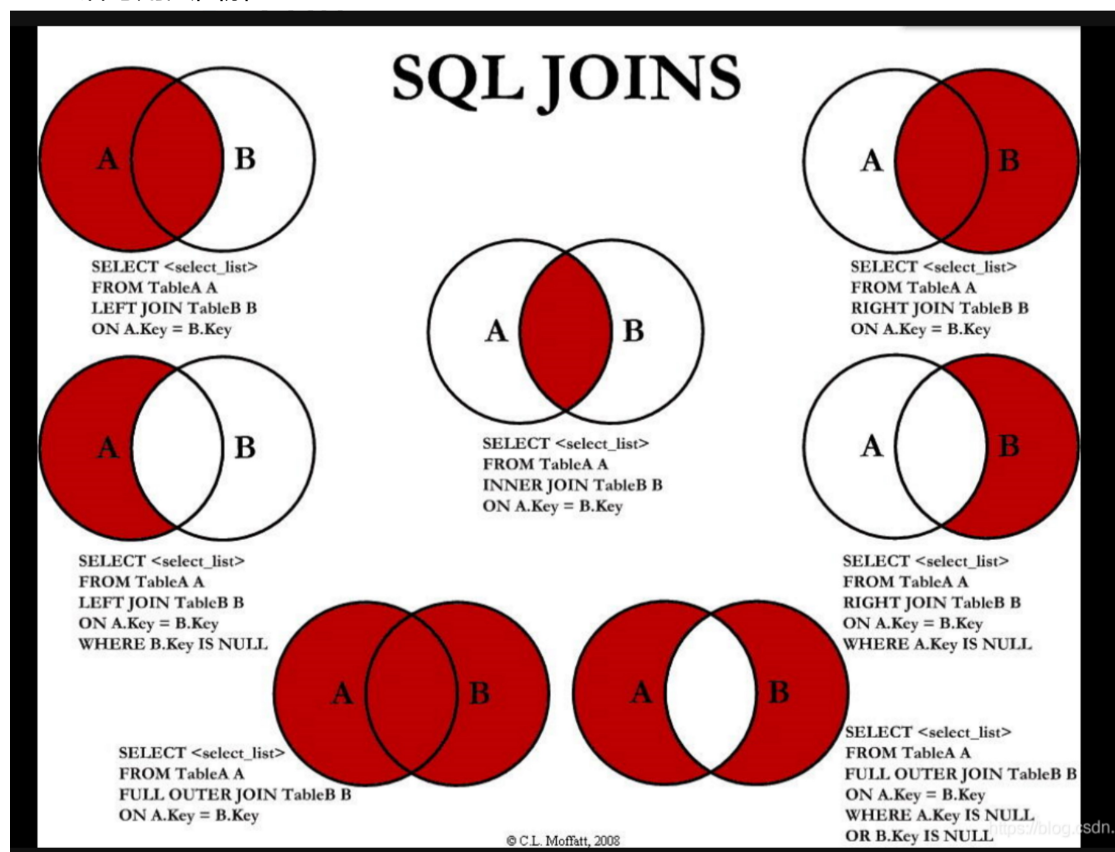
rows: 行数，指定了返回结果集的行数

在无GROUP BY的情况下直接使用HAVING语句的问题:

当GROUP BY NULL的时候，MAX/MIN函数是取所有数据里的最大和最小值！

"SELECT * FROM t HAVING id=MIN(id)"本质上是"SELECT * FROM t HAVING id=1"，就能返回一条记录，而"SELECT * FROM t HAVING id=MAX(id)"本质上是"SELECT * FROM t HAVING id=3"，当然没有返回记录，这就是问题的根源。

JOIN语句用法图解



内连接(获取两张表的共同数据):

```
SELECT * FROM a [INNER] JOIN b ON a.key = b.key;
```

左外连接(获取a和b的共同数据加上a的单独数据):

```
SELECT * FROM a LEFT JOIN b ON a.key = b.key;
```

右外连接(获取a和b的共同数据加上b的单独数据):

```
SELECT * FROM a RIGHT JOIN b ON a.key = b.key;
```

全连接(获取表a和b上所有的数据):

联合查询(union):

```
SELECT * FROM a LEFT JOIN b ON a.key = b.key
```

```
UNION
```

```
SELECT * FROM a RIGHT JOIN b ON a.key = b.key;
```

表a或表b独有的数据:

```
SELECT * FROM a LEFT JOIN b ON a.key = b.key WHERE b.key IS NULL;
```

```
SELECT * FROM a RIGHT JOIN ON a.key = b.key WHERE a.key IS NULL;
```

关系型数据库事务的四大特性: 原子性、一致性、隔离性、持久性(ACID特性)

关系型数据库: SQL Serve、MySQL、Oracle

非关系型数据库: Redis、MongoDB、CouchDB

MySQL索引的相关信息:

索引的数据结构: 二叉树、红黑树、Hash表、B-树和B+树等等