

MySQL-Day02必须掌握

外键

原理

- 1 让当前表字段的值在另一张表的范围内去选择

使用规则

- 1 数据类型要一致
- 2 主表被参考字段必须为KEY的一种 : PRI

级联动作

- 1 cascade : 删除 更新同步(被参考字段)
- 2 restrict(默认) : 不让主表删除 更新
- 3 set null : 删除 更新,从表该字段值设置为NULL

嵌套查询 (子查询)

定义

- 1 把内层的查询结果作为外层查询的条件

多表查询

笛卡尔积

- 1 多表查询不加where条件,一张表的每条记录分别和另一张表的所有记录分别匹配一遍

连接查询

分类

- 1、内连接 (表1 `inner join` 表2 `on` 条件)
- 2、外连接 (表1 `left|right join` 表2 `on` 条件)
 - 1、左连接 : 以左表为主显示查询结果
 - 2、右连接 : 以右表为主显示查询结果

语法

- 1 `select` 表名.字段名 `from` 表1 `inner join` 表2 `on` 条件;

锁

1、目的：解决客户端并发访问的冲突问题

2、锁分类

- 1、锁类型：读锁 写锁
- 2、锁粒度：行级锁 表级锁

数据导入

方式一（使用source命令）

- 1 `mysql> source /home/tarena/xxx.sql`

方式二（使用load命令）

- 1、将导入文件拷贝到数据库搜索路径中
- 2、在数据库中创建对应的表
- 3、执行数据导入语句

索引

定义

- 1 对数据库表中一列或多列的值进行排序的一种结构

优点

- 1 加快数据的检索速度

缺点

- 1 1、占用实际物理存储空间
- 2 2、索引需动态维护，消耗资源，降低数据的维护速度

分类及约束

- 1 1、普通索引 (MUL)：无约束
- 2 2、唯一索引 (UNI)：字段值不允许重复，但可为NULL
- 3 3、主键 (PRI)：字段值不允许重复，不可为NULL
- 4 4、外键：让当前表字段的值在另一张表的范围内选择

MySQL-Day03笔记

存储引擎

定义

- 1 处理表的处理器

基本操作

- 1 1、查看所有存储引擎
- 2 `mysql> show engines;`
- 3 2、查看已有表的存储引擎
- 4 `mysql> show create table 表名;`
- 5 3、创建表指定
- 6 `create table 表名(...)engine=MyISAM;`
- 7 4、已有表指定
- 8 `alter table 表名 engine=InnoDB;`

常用存储引擎及特点

■ InnoDB

- 1 1、支持行级锁
- 2 2、支持外键、事务操作
- 3 3、表字段和索引同存储在一个文件中
- 4 1、表名.frm：表结构及索引文件
- 5 2、表名.ibd：表数据

■ MyISAM

- 1 1、支持表级锁
- 2 2、表字段和索引分开存储
- 3 1、表名.frm : 表结构
- 4 2、表名.myi : 索引文件
- 5 3、表名.myd : 表记录

■ MEMORY

- 1 1、表记录存储在内存中，效率高
- 2 2、服务或主机重启，表记录清除

如何选择存储引擎

- 1 1、执行查操作多的表用 MyISAM(使用InnoDB浪费资源)
- 2 2、执行写操作多的表用 InnoDB

MySQL的用户账户管理

开启MySQL远程连接

- 1 更改配置文件，重启服务！

添加授权用户

- 1 1、用root用户登录mysql
- 2 mysql -uroot -p123456
- 3 2、授权
- 4 grant 权限列表 on 库.表 to "用户名"@"%" identified by "密码" with grant option;

权限列表

- 1 all privileges 、 select 、 insert
- 2 库.表 : * . * 代表所有库的所有表

示例

- 1 1、添加授权用户work,密码123,对所有库的所有表有所有权限
- 2 2、添加用户duty,对db2库中所有表有所有权限

事务和事务回滚

事务定义

- 1 一件事从开始发生到结束的过程

作用

- 1 确保数据的一致性、准确性、有效性

事务操作

- 1 1、开启事务
- 2 2、开始执行事务中的1条或者n条SQL命令
- 3 3、终止事务

事务四大特性 (ACID)

■ 1、原子性 (atomicity)

- 1 一个事务必须视为一个不可分割的最小工作单元，整个事务中的所有操作要么全部提交成功，要么全部失败回滚，对于一个事务来说，不可能只执行其中的一部分操作

■ 2、一致性 (consistency)

- 1 数据库总是从一个一致性的状态转换到另一个一致性的状态

■ 3、隔离性 (isolation)

- 1 一个事务所做的修改在最终提交以前，对其他事务是不可见的

■ 4、持久性 (durability)

- 1 一旦事务提交，则其所做的修改就会永久保存到数据库中。此时即使系统崩溃，修改的数据也不会丢失

注意

- 1 1、事务只针对于表记录操作(增删改)有效,对于库和表的操作无效
- 2 2、事务一旦提交结束，对数据库中数据的更改是永久性的

E-R模型(Entry-Relationship)

定义

- 1 E-R模型即 实体-关系 数据模型,用于数据库设计
- 2 用简单的图(E-R图)反映了现实世界中存在的事物或数据以及他们之间的关系

实体、属性、关系

■ 实体

- | | |
|---|------------------------|
| 1 | 1、描述客观事物的概念 |
| 2 | 2、表示方法：矩形框 |
| 3 | 3、示例：一个人、一本书、一杯咖啡、一个学生 |

■ 属性

- | | |
|---|-------------------------|
| 1 | 1、实体具有的某种特性 |
| 2 | 2、表示方法：椭圆形 |
| 3 | 3、示例 |
| 4 | 学生属性：学号、姓名、年龄、性别、专业 ... |
| 5 | 感受属性：悲伤、喜悦、刺激、愤怒 ... |

■ 关系 (重要)

- | | |
|----|------------------------------|
| 1 | 1、实体之间的联系 |
| 2 | 2、一对一关联(1:1)：老公对老婆 |
| 3 | A中的一个实体,B中只能有一个实体与其发生关联 |
| 4 | B中的一个实体,A中只能有一个实体与其发生关联 |
| 5 | 3、一对多关联(1:n)：父亲对孩子 |
| 6 | A中的一个实体,B中有多个实体与其发生关联 |
| 7 | B中的一个实体,A中只能有一个与其发生关联 |
| 8 | 4、多对多关联(m:n)：兄弟姐妹对兄弟姐妹、学生对课程 |
| 9 | A中的一个实体,B中有多个实体与其发生关联 |
| 10 | B中的一个实体,A中有多个实体与其发生关联 |

ER图的绘制

矩形框代表实体,菱形框代表关系,椭圆形代表属性

■ 课堂示例 (老师研究课题)

- | | |
|---|------------------------------|
| 1 | 1、实体：教师、课题 |
| 2 | 2、属性 |
| 3 | 教师：教师代码、姓名、职称 |
| 4 | 课题：课题号、课题名 |
| 5 | 3、关系 |
| 6 | 多对多 (m:n) |
| 7 | # 一个老师可以选择多个课题，一个课题也可以被多个老师选 |

■ 练习

设计一个学生选课系统的E-R图

- | | |
|---|------|
| 1 | 1、实体 |
| 2 | 2、属性 |
| 3 | 3、关系 |

关系映射实现 (重要)

- | | |
|---|-----|
| 1 | 1:1 |
| 2 | 1:n |
| 3 | m:n |

多对多实现

■ 老师研究课题

- 1 表1、老师表
- 2 表2、课题表
- 3 问题？如何实现老师和课程之间的多对多映射关系？

■ 后续

- 1 1、每个老师都在研究什么课题？
- 2 2、程序员姐姐在研究什么课题？

MySQL调优

存储引擎优化

- 1 1、读操作多：
- 2 2、写操作多：

索引优化

- 1 在 select、where、order by常涉及到的字段建立索引

SQL语句优化

- 1 1、单条查询最后添加 `LIMIT 1`，停止全表扫描
- 2 2、where子句中不使用 `!=`，否则放弃索引全表扫描
- 3 3、尽量避免 `NULL` 值判断, 否则放弃索引全表扫描
优化前: `select number from t1 where number is null;`
优化后:
在number列上设置默认值0, 确保number列无NULL值
- 4 4、尽量避免 `or` 连接条件, 否则放弃索引全表扫描
优化前: `select id from t1 where id=10 or id=20;`
优化后:
- 5 5、模糊查询尽量避免使用前置 `%`，否则全表扫描
`select name from t1 where name like "%c%";`
- 6 6、尽量避免使用 `in` 和 `not in`, 否则全表扫描
优化前: `select id from t1 where id in(1,2,3,4);`
优化后:
- 7 7、尽量避免使用 `select * ...`; 用具体字段代替 `*`，不要返回用不到的任何字段