

MySQL数据库

SQL: 结构化查询语言

- 作用: 是一种定义, 操作, 管理关系数据库的句法。
- 组成: DQL, DML, DDL, TPL, DCL, CCL

数据库分类:

- 关系型数据库: Oracle, MySQL, SQLite等等
- 非关系型数据库(NOSQL): Redis, Memcached, MongoDB等。

DDL数据定义语言

- 作用创建数据库和表的结构。
- 关键字: CREATE, ALTER, DROP, TRUNCATE
 - 创建数据库: `CREATE DATABASE[SCHEMA] databasename;`
 - 使用数据库: `USE databasename;`
 - 创建表: `CREATE TABLE tablename(field1 datatype,...) character set [字符集] collate [校对规则]`
 - 修改表: `ALTER TABLE tablename ADD/MODIFY/DROP column (datatype [DEFAULT...])`
 - `ALTER TABLE class ADD math float DEFAULT 60;`
 - `ALTER TABLE class CHANGE math math int DEFAULT 70;`
 - `ALTER TABLE class MODIFY math float DEFAULT 60;`
 - `ALTER TABLE class DROP math;`

DML数据操纵语言

- 作用: 用于向数据库中插入, 删除, 修改数据
- 关键字: INSERT, UPDATE, DELETE
 - 插入数据: `INSERT INTO tablename (col,col) VALUES (data,data);`
 - 修改表中数据: `UPDATE tablename SET col= value WHERE`
 - 删除数据: `DELETE FROM tablename WHERE`

DQL数据查询语言

- 作用: 查询数据
- 关键字: SELECT
- 简单查询
 - `SELECT [DISTINCT] *(col,col...) FROM tablename WHERE ...`
 - `SELECT col expression col FROM tablename;`
 - `SELECT *|col FROM tablename ORDER BY col asc|desc;`

数据完整性

- 为了保证插入到数据库中的数据是正确的, 防止了用户可能的输入错误。
- 分类
 - 实体完整性: 规定每一行记录在表中是唯一的实体。通过表的主键来实现。
 - 域完整性: 数据库中表的列(即字段)必须符合某种特定的数据类型或约束。
 - 参照完整性: 保证一个表的外键和另一个表的主键对应

多表设计

- 一对一
- 一对多
- 多对多

复杂查询

- 连接查询
 - 交叉连接: `cross join`, 返回连接表中所有数据行的笛卡尔积。
 - 内连接: `inner join` 显式内连接, 逗号隔开查询的表, 隐式内连接。
 - 外连接:
 - 左外连接: `left outer join`
 - 右外连接: `right outer join`
- 分组查询
 - `group by` — `having` 子句设定查询条件
 - 统计函数
 - 计数`count`
 - 求和`sum`
 - 求平均`avg`
 - 最大`max`
 - 最小`min`

数据库索引

- 索引
 - 索引是帮助数据库高效获取数据的数据结构(提升效率)
 - 查找是数据库使用频率最高的操作, 减少查询花费的时间能够极大的提高数据的效率
- 数据结构 — 要求
 - 时间复杂度
 - 查找效率
 - 空间复杂度
 - 磁盘IO效率(磁盘IO尽可能少)
 - 不能占用太大的连续的磁盘空间