MYSQL

基础

常见命令

查看当前所有的数据库：show databases;
打开指定的库：use 库名
查看当前库的所有表：show tables;
查看当前其他库的所有表：select * from 库名
创建表 ：create table 表名（字段名 类型）;
查看表结构：desc 表名;
查看服务器版本: select version(); || mysql -V
规范：关键字大写，表名列名小写

crud

select

常量

select 100;

表达式

select 要查询的字段||* from 表名 where 条件

函数

select version(); length() concat() ifnull()

去重

distinct

concat()

查询员工名和姓连接成一个字段: select concat('first_name','last_name') as 字段 from 表名

+

数值型的计算，字符串会转化为数值型进行计算，如果没有数值就是0，还有就是只要计算的一方为null,结果也为null

IFNULL(字段，值)

函数判断是否为空，如果为空就返回自定义值，不为空就返回数据：select IFNULL(字段名,0) as new字段 from 表名 where 条件

运算符

简单条件运算符

> < = != <> >= <=

逻辑运算符

&& || ! and or not

模糊查询

like（%为任意多个，_为任意一个字符） between and in is null

排序查询

select* from 表名 where 条件 order by (要比较的值,可填可不填)(desc||asc)

多条件排序,用逗号分割：工资升序,员工号降序 select * from 表名 order by salary asc,id desc;

insert

insert into 表（字段）values (值)

update

update 表名 set 字段= 值

delete

delete from 表名 where 条件

DQL：数据查询语言

联合查询union

select *from 表 where 条件 union select * from 表where 条件

应用场景：要查询的结果来自于多个表，切多个表之间没有直接的关联关系，会去重，如果不去重，使用union all

分组查询

group by

select avg(salary),dp_id from表 where 条件 group by 分组依据(dp_id) having 筛选条件 order by avg(salary) asc

常见函数

单行函数

字符函数

concat

concat("a","b")

length

length("joon")

ifnull

判断是否为空

substr,substring

select substring('assddfd',6) ||select substring('assddfd',2,6)

trim

去除字符串前后符合表达式的字符

lpad

用指定字符实现左填充指定长度 lpad('aaa',2,'*')

rpad

用指定字符实现右填充指定长度 lpad('aaa',2,'*')

replace

替换：repace('aaaae','ae','bb')

数学函数

ceil

上

round

四舍五入
floor
下
truncate 截断
日期函数
now
curdate 包含日期，不包含时间
curtime
year
select year('1998-01-01')
month
str_to_date
date_format
流程控制函数
if
select 字段1，字段2,if(条件，'表达式1','表达式2')
case
select salary,dp_id， case dp_id when 30 then salary *1.2 when 40 then salary *1.3 else salary end as 新工资 from 表
其他函数
version
user
database
分组函数
sum
select sum(salary) from 表
avg
select avg(salary) from 表
max
min
count
多表连接查询
内连接 inner
等值
sql92:select * from 表1，表2 where 表1.字段=表2.字段
sql99: select * from 表1 inner join 表2 where 表1.字段=表2.字段
非等值
自连接

语法
select 查询列表 from 表1 [连接类型] join 表2 on 连接条件 【where 条件】【group by 分组条件】【having 筛选】【order by 排序列表】
外连接:一个表有另外一个表没有
左外连接 left
sql99: select * from beauty
右外连接 right
全外连接 full
特点：外连接的查询结果为主表中的所有记录，如果从表中有和它匹配的，显示匹配的值，没有匹配的，显示null，外连接查询结果=内连接结果+主表中有而从表没有的记录 左外连接：left join 左边的是主表，右外连接 ：right join右边的是主表
交叉连接
笛卡尔乘积
子查询
分页查询
slelect 查询列表 from 表 【join type join 表2 on条件 where条件 group 分组 having分组后的筛选 order by 排序 limit offset,size】(page-1)*size,size

DML：数据操纵语言

插入语句
insert into 表名(列名,...) values (值1,....)
insert into 表名 set 列名=值，列名=值,...
更新语句
update 表名set 列名=新值 where 条件
删除语句
单表删除delete from 表 where 条件，delete删除在插入，自增长的列的值从断点开始
表全部信息删除：truncate table 表名，truncate删除在插入，自增长的列额值从1开始
多表删除：连接
sql92:delete 别名1，别名2 from 表1 别名1，表2 别名2 where 连接条件 and 筛选条件
sql99：:delete 别名1，别名2 from 表1 别名1 inner|right|left join 表2 别名2 on 连接条件 where 筛选条件

DDL：数据定义语言

库的管理
表的管理
约束
notnull 非空
default 默认
primarykey 主键
unique 唯一
foreign key
添加表级约束
语法：在各个字段的最下面（constraint 约束名 约束类型（字段名））

alter table 表名 modify column 列名 类型 约束

索引

主键

外键

唯一

自建索引

tcl：事务控制语言

事务:一个或一组sql语句组成一个执行单元，这个执行单元要么全部执行，要么全部不执行

事务

特点ACID

A原子性：事务是一个不可分割的工作单位，要么全部发生，要么全部不发生

C一致性：事务必须使数据库从一个一致性状态转换到另一个一致性状态

i隔离性：一个事务的执行不受其他事务干扰，一个事务内部的操作及使用的数据对并发的其他事务是隔离的，并发执行的各个事务之间不能相互干扰

D持久性：事务一旦提交，他对数据库中数据的改变是永久性的接下来的其他操作和数据故障不会对其有任何影响

隔离级别

赃读

读未提交

读未提交，顾名思义，就是一个事务可以读取另一个未提交事务的数据。 事例：老板要给程序员发工资，程序员的工资是3.6万/月。但是发工资时老板不小心按错了数字，按成3.9万/月，该钱已经打到程序员的户口，但是事务还没有提交，就在这时，程序员去查看自己这个月的工资，发现比往常多了3千元，以为涨工资了非常高兴。但是老板及时发现了不对，马上回滚差点就提交了的事务，将数字改成3.6万再提交。分析：实际程序员这个月的工资还是3.6万，但是程序员看到的是3.9万。他看到的是老板还没提交事务时的数据。这就是脏读。

不可重复读

读提交

读提交，顾名思义，就是一个事务要等另一个事务提交后才能读取数据。 事例：程序员拿着信用卡去享受生活（卡里当然是只有3.6万），当他埋单时（程序员事务开启），收费系统事先检测到他的卡里有3.6万，就在这个时候！！程序员的妻子要把钱全部转出充当家用，并提交。当收费系统准备扣款时，再检测卡里的金额，发现已经没钱了（第二次检测金额当然要等待妻子转出金额事务提交完）。程序员就会很郁闷，明明卡里是有钱的…分析：这就是读提交，若有事务对数据进行更新（UPDATE）操作时，读操作事务要等待这个更新操作事务提交后才能读取数据，可以解决脏读问题。但在这个事例中，出现了一个事务范围内两个相同的查询却返回了不同数据，这就是不可重复读。

幻读

同一事务中，使用相同的查询语句，第二次查询时，莫名的多出了一些之前不存在数据，或者莫名不见了之前存在的数据

Serializable 序列化

Serializable 是最高的事务隔离级别，在该级别下，事务串行化顺序执行，可以避免脏读、不可重复读与幻读。但是这种事务隔离级别效率低，比较耗数据库性能，一般不使用。

高级