MYSQL语法总结版本

A.SQL

DDL(definition)

- 数据库操作
  1.查询所有数据库 show databases;
  2.查询当前数据库 select database();
  3.创建数据库 create database 【if not exists】数据库名【default charset字符集】【collate 排序规则】;
  4.删除数据库 drop database 【if exists】数据库名;
  5.切换数据库 use 数据库名;
- 表操作
  查询创建
  1.查询当前数据库所有的表 show tables;
  2.查看指定的表结构 desc 表名;
  3.查询指定表的建表语句 show create table 表名;
  4.创建表结构 create table 表名（
  字段1 字段1类型【comment 1注释】,
  字段2 字段2类型【comment 2注释】）【注释】;

修改表
1.添加字段 ALTER TABLE 表名 ADD 字段名 类型（长度）【注释】【约束】;
2.修改数据类型 ALTER TABLE 表名 MODIFY 字段名 类型（长度）;
3.修改字段名和字段类型 ALTER TABLE 表名 CHANGE 旧字段名 新字段名 类型 (长度)【注释】
4.删除字段 ALTER TABLE 表名 DROP 字段名
5.修改表名 ALTER TABLE 表名 RANAME TO 新表名

删除表
1.删除表 DROP TABLE【if exists】表名
2.删除指定表并重新创建表 TRUNCATE TABLE 表名

DML(manipulation)

- 添加数据
  1.给指定字段添加数据 INSERT INTO 表名（字段名1，字段名2）VALUES（值1，值2）
  2.给全部字段添加数据 INSERT INTO 表名 VALUES(值1，值2)
  3.批量添加数据 在上两条的VALUE基础之上，加上 ，(值1，值2)
- 删除数据
  DELETE FROM 表名 【WHERE 条件】

- 修改数据
  UPDATE 表名 SET 字段名1 = 值1，字段名2 = 值2，...【WHERE 条件】
- 查询数据
  select * from 表名

DQL(query)

- 基本查询
  1.查询多个字段 SELECT 字段1，字段2，字段3...FROM表名；
  SELECT * FROM 表名；//尽量少用，不直观；
  2.字段设置别名 SELECT 字段1 【AS 别名1】字段2 【AS 别名2】 ... FROM 表名；
  SELECT 字段1 【别名1】，字段2 【别名2】... FROM 表名；
  3.去除重复记录 SELECT DISTINCT 字段列表 FROM 表名；
- 条件查询（where）
  SELECT 字段列表 FROM 表名 WHERE 条件列表
- 聚合查询（count、max、min、avg、sum）
  SELECT 聚合函数(字段列表) FROM 表名 //NULL不参与所有聚合函数的运算
- 分组查询（group by）
  SELECT 字段列表 FROM 表名 【WHERE条件】 GROUP BY 分组字段名【HAVING(聚合函数和分组字段）】
- 排序查询（order by）
  SELECT 字段列表 FROM 表名 ORDER BY 字段1 排序方式1 , 字段2 排序方式2 （ASC升 DESC降）
- 分页查询（limit）
  SELECT 字段列表 FROM 表名 LIMIT 起始索引, 查询记录数
  （起始索引从0开始，起始索引 = （查询页码 - 1）* 每页显示记录数。）

执行顺序
from -> where -> group by ->having -> select ->order by -> limit

DCL(control) 数据库权限

- 管理用户
  1.查询用户 select * from mysql.user
  2.创建用户 CREATE USER '用户名'@'主机名' IDENTIFIED BY '密码';
  3.修改用户密码ALTER USER '用户名'@'主机名' IDENTIFIED WITH mysql_native_password BY '新密码' ;
  4.删除用户DROP USER '用户名'@'主机名' ;
- 权限控制
  1.查询权限 SHOW GRANTS FOR '用户名'@'主机名' ;
  2.授予权限 GRANT 权限列表 ON 数据库名.表名 TO '用户名'@'主机名';
  3.撤销权限 REVOKE 权限列表 ON 数据库名.表名 FROM '用户名'@'主机名';

函数

1. 字符串函数

函数	功能
CONCAT(S1,S2,...Sn)	字符串拼接，将S1，S2，... Sn拼接成一个字符串
LOWER(str)	将字符串str全部转为小写
UPPER(str)	将字符串str全部转为大写
LPAD(str,n,pad)	左填充，用字符串pad对str的左边进行填充，达到n个字符串长度
RPAD(str,n,pad)	右填充，用字符串pad对str的右边进行填充，达到n个字符串长度
TRIM(str)	去掉字符串头部和尾部的空格
SUBSTRING(str,start,len)	返回从字符串str从start位置起的len个长度的字符串

2. 数值函数

函数	功能
CEIL(x)	向上取整
FLOOR(x)	向下取整
MOD(x,y)	返回x/y的模
RAND()	返回0~1内的随机数
ROUND(x,y)	求参数x的四舍五入的值，保留y位小数

3. 日期函数

函数	功能
CURDATE()	返回当前日期
CURTIME()	返回当前时间
NOW()	返回当前日期和时间
YEAR(date)	获取指定date的年份
MONTH(date)	获取指定date的月份
DAY(date)	获取指定date的日期
DATE_ADD(date, INTERVAL expr type)	返回一个日期/时间值加上一个时间间隔expr后的时间值
DATEDIFF(date1,date2)	返回起始时间date1 和 结束时间date2之间的天数

4.流程函数

函数	功能
IF(value , t , f)	如果value为true，则返回t，否则返回f
IFNULL(value1 , value2)	如果value1不为空，返回value1，否则返回value2
CASE WHEN [val1] THEN [res1] ... ELSE [default] END	如果val1为true，返回res1，... 否则返回default默认值
CASE [expr] WHEN [val1] THEN [res1] ... ELSE [default] END	如果expr的值等于val1，返回res1，... 否则返回default默认值

约束

可以在创建表或者修改表的时候**添加约束**
字段名称 数据类型 【约束类型】【注释】
1.非空约束 NOT NULL（非空）
2.唯一约束 UNIQUE（唯一，不重复）
3.主键约束 PRIMARY KEY（唯一标识，非空且唯一）

4.默认约束 DEFAULT
5.检查约束 CHECK（保证字段满足某一个条件）
6.外键约束 FOREIGN KEY （两张表建立连接，保证数据的一致性和完整性）

外键约束
1.添加外键 CREATE TABLE 表名(字段名 数据类型, ... 【约束】【外键名称】FOREIGN KEY (外键字段名) REFERENCES 主表 (主表列名));
ALTER TABLE 表名 ADD CONSTRAINT 外键名 FOREIGN KEY (外键字段名) REFERENCES 主表 (主表列名);
2.删除外键 ALTER TABLE 表名 DROP FOREIGN KEY 外键名称;
3.删除/更新行为
ALTER TABLE 表名 ADD CONSTRAINT 外键名称 FOREIGN KEY (外键字段) REFERENCES 主表名 (主表字段名) ON UPDATE CASCADE ON DELETE CASCADE

行为	说明
NO ACTION	当在父表中删除/更新对应记录时，首先检查该记录是否有对应外键，如果有则不允许删除/更新。 (与 RESTRICT 一致) 默认行为
RESTRICT	当在父表中删除/更新对应记录时，首先检查该记录是否有对应外键，如果有则不允许删除/更新。 (与 NO ACTION 一致) 默认行为
CASCADE	当在父表中删除/更新对应记录时，首先检查该记录是否有对应外键，如果有，则也删除/更新外键在子表中的记录。
SET NULL	当在父表中删除对应记录时，首先检查该记录是否有对应外键，如果有则设置子表中该外键值为null (这就要求该外键允许取null) 。
SET DEFAULT	父表有变更时，子表将外键列设置成一个默认的值 (Innodb不支持)

多表查询

- 连接查询
  1.内连接（相当于两张表的交集部分数据）
  a.隐式内连接 SELECT 字段列表 FROM 表1 , 表2 WHERE 条件 ...
  b.显示内连接 SELECT 字段列表 FROM 表1 [INNER] JOIN 表2 ON 连接条件 ...
  （比如emp.dept_id = dept.id）

  2.外连接
  左外连接：SELECT 字段列表 FROM 表1 LEFT [OUTER] JOIN 表2 ON 条件 ... ;

（更偏向）
右外连接：SELECT 字段列表 FROM 表1 RIGHT [OUTER] JOIN 表2 ON 条件 ... ；

3.自连接
自连接查询：SELECT 字段列表 FROM 表A 别名A JOIN 表A 别名B ON 条件 ... ；
联合查询：（多次查询的结果合并在一起）
（对于联合查询的多张表的列数必须保持一致，字段类型也需要保持一致。）
SELECT 字段列表 FROM 表A ...
UNION [ALL]
SELECT 字段列表 FROM 表B ....;
（union all 会将全部的数据直接合并在一起，union 会对合并之后的数据去重）

- 子查询 （嵌套查询，SQL嵌套select语句，外部可以是INSERT/UPDATE/DELETE/SELECT）

标量子查询（子查询结果为单个值）= != > < <=
列子查询(子查询结果为一列) IN ANY SOME ALL

操作符	描述
IN	在指定的集合范围之内，多选一
NOT IN	不在指定的集合范围之内
ANY	子查询返回列表中，有任意一个满足即可
SOME	与ANY等同，使用SOME的地方都可以使用ANY
ALL	子查询返回列表的所有值都必须满足

行子查询(子查询结果为一行) IN , NOT IN
表子查询(子查询结果为多行多列) IN

事务

一组操作的集合
控制事务1：
查看/设置事务提交方式：SELECT @@autocommit ;
SET @@autocommit = 0 ;
提交事务 COMMIT;
回滚事务 ROLLBACK;
//此时要手动的执行COMMIT进行提交

控制事务2：
开启事务：START TRANSACTION 或 BEGIN ；
提交事务：COMMIT;
回滚事务：ROLLBACK;

6.5 事务隔离级别

为了解决并发事务所引发的问题，在数据库中引入了事务隔离级别。主要有以下几种：

隔离级别	脏读	不可重复读	幻读
Read uncommitted	√	√	√
Read committed	×	√	√
Repeatable Read(默认)	×	×	√
Serializable	×	×	×

1). 查看事务隔离级别

```
SELECT @@TRANSACTION_ISOLATION;
```

2). 设置事务隔离级别

```
SET [ SESSION | GLOBAL ] TRANSACTION ISOLATION LEVEL { READ UNCOMMITTED |
READ COMMITTED | REPEATABLE READ | SERIALIZABLE }
```

> 注意：事务隔离级别越高，数据越安全，但是性能越低。