链滴

关系型数据库（RDBMS Relational database management system）

作者：020

原文链接：https://ld246.com/article/1610519894595

来源网站：链滴

关系型数据库MySQL的详细学习

<!--more-->

一，数据库的发展历程：

·没有数据库，使用磁盘文件进行存储
·层次结构模型数据库
·网状结构模型数据库
·关系型结构模型数据库：使用二位表格存储数据
·关系：对象模型数据库

1，常见的数据库：

·Oracle：甲骨文
·DB2:IBM
·Sybase:赛尔斯
·Mysql：甲骨文
·SQL Server:微软

2，理解数据库

RDBMS：管理员（manager)+仓库(DB)
DB:多个table
Table：多行数据

二，SQL语言结构化查询语言（Structured Query Language)

1，sql的作用

客户端使用sql俩操作服务器
    ·启动mysql.exe，连接服务器后，就可以使用sql来操作服务器了

2，sql标准

有国际标准化组织ISO指定的，对DBMS的同一操作方式

3，sql方言

某种DBMS不仅支持sql标准，而且还会有一些自己独有的语法，这称之为方言，
例如limit只有mysql可以使用

4，sql语句

1)sql语句可以多行或单行书写，以分号结尾
2. 可使用空格和缩进来增强语句的可读性
3. MySQL不区分大小写，建议关键字使用大写

三，sql语句的分类

1，DDL(Data Definition Language)

数据定义语言，用来定义数据库对象：库，表，列等
·创建，删除，修改：库，表结构

2，DML(Data Manipulation Language)

数据操作语言，用来定义数据库记录（数据）
·增，删，改，更新：表记录

3，DCL(Data control Language)

数据库控制语言，用来定义访问权限和安全级别
·对用户的创建，及授权

4，DQL(Data Query Language)

数据库查询语言，用来查询记录（数据）

四，数据类型

·int 整型
·double 浮点型 double(5,2)表示最多五位，其中必须有2位小数，即最大值位999.99；
·decimal 浮点型， 在表单，钱方面使用该类型，因为不会出现精度确实问题。
·char 固定长度的字符串类型 char(255)//所有字符都为给定长度，最大255,不足则补足到给定长度
·varchar 可变长度的字符串类型 ： varchar(65535)，指定长度。最大值为65535，单独花一个字节录长度
·text（clob):字符串类型，可以存储超大的文本。
·tinytext，2^8 -1B
·text，2^16 -1B
·mediumtext: 2^24 -1B
·longtext:2^32 -1B
·binary 255B
·blob：表示二进制数字
·tinyblob，2^8 -1B
·blob，2^16 -1B
·mediumblob: 2^24 -1B
·longblob:2^32 -1B
·date：日期，只有年月日
·time：时间类型，只有时分秒
·timestamp：时间戳类型，年月日时分秒

五，具体语句

1，登录及使用数据库

·mysql -u '用户' -p '密码' //登录

·show databases; //显示所有数据库
·use '数据库名' //使用指定数据库
·net stop mysql //停止服务
·net start mysql //关闭服务

2，DDL（Data Definition Language）语句

·对数据库的操作
  ·CREATE DATABASE '新建数据库名'[CHARSER=utf8];//创建一个数据库,以默认编码创建
  ·DROP DATABASE '数据库名' //删除数据库
  ·ALTER DATABASE '数据库名' CHARSET SET utf8;//修改数据库编码
·对表的操作：
  ·查看当前数据库中所有的表名称：show tables;
  ·查看指定表的创建语句：SHOW CREATE TABLE 表名
  ·查看表结构：DESC 表名；
  ·删除表:DROP TABLE 表名
  ·创建表
    CREATE TABLE [IF NOT EXISTS]表名(
    列名 列类型,
    列名 列类型,
    列名 列类型,
    列名 列类型,
    ……
    );
    实例：
      CREATE DATABASE mydb3;
      USE mydb3;
      CREATE TABLE tb_stu(
        number CHAR(11),
        name VARCHAR(50),
        age INT,
        gender VARCHAR(10)
        );
  ·修改表
    ·修改表名称
      ALTER TABLE 表名
        RENAME TO 新表名
    ·添加列
      ALTER TABLE 表名
        ADD(
        列名 列类型,
        列名 列类型,
        列名 列类型,
        列名 列类型,
        ……
)
    ·修改列类型
      ALTER TABLE 表名
        MODIFY 列名 类类型
    ·修改列名称
    ALTER TABEL 表名
      CHANGE 原列名 新列名 类型
    ·删除列
    ALTER TABEL 表名

DROP 列名

3，DML(Data Definition Language)数据库操作语言-对记录进行操作

·插入语句
//给定字段插入，没有赋值的字段为NULL
INSERT INTO 表名
(插入的字段)
VALUES
(赋值字段的值)
//默认插入所有字段，且按表中的顺序给值
INSERT INTO 表名
VALUES
(赋值字段的值)
//同时插入多条语句
INSERT INTO 表名
VALUES
(赋值字段的值),(赋值字段的值),(赋值字段的值),.....
·修改语句，更新语句
//将所有的选定列名的值改为给定的列值
UPDATE 表名
SET
列名="列值",列名="列值",.....;
//加上条件，则按条件选择指定记录
UPDATE 表名
SET
列名="列值",列名="列值",.....WHERE 条件;
·条件
·必须是Boolean类型的值或表达式
·运算符：=，!=,<>,>,<,>=,<=,BETWEEN.....AND,IN(.....) IS NULL,NOT,OR,AND
·BETWEEN AND 在之间的数值条件
·IN("","") 集合，在集合中查找条件值
·IS NULL :满足为空则执行，不能用 字段=null（值为false）
·删除语句
//不加条件则将所有记录全部删除
DELETE FORM 表名 WHERE 条件

4，DCL(Data Control Language)数据控制语句--对用户和权限的控制

一个项目创建一个用户，一个项目对应的数据库只有一个，项目创建的用户只能对项目对应的数据库操作权限，其他数据库
无权限操作。
1，创建用户
CREATE USER 用户名@IP地址 IDENTIFIED BY '密码'
指定用户的用户名，密码和IP地址，若IP地址设为'%',则表示用户的IP任意
2，给用户授权
GRANT 权限1，.....,权限n ON 数据库.* TO 用户名@IP地址
给用户授予对数据库的n个权限
指定权限为all表示开放所有权限，数据库.*表示对所有内容给定权限
3，删除用户
·DROP USER 用户名@IP地址

5，DQL(Data Query Language) 数据库查询语句

所有的操作都是对结果集的操作，这是一个很重要的理解

1. 字段控制查询
·查询所有列
    ·SELECT *FROM 表名
·查询指定列
    SELECT 列1，列2，列3,……FROM 表名
·去除重复行查询
    SELECT DISTINCT *FROM 表名 //田间关键字
·模糊查询
    使用like关键字配合模式匹配进行符合条件的查询
        ·_：表示一个字符，如"张_"匹配张姓两个字的
        ·%：表示多个字符，0~n
·排序查询
    关键字：ORDER BY 排序列 排序模式 //默认升序
        模式：ASC 升序
            DESC 降序
    多重比较排序：
        ORDER BY 排序列1 排序模式1,排序列2 排序模式2
        若按排序列1排序后有记录的排序列相同，则相同排序列的记录按排序列2进行排序模式2排序，次类推，直到没有
        相同排序列的记录为止。
·聚合函数
    ·count()//计算个数
        count(*) //表示计算不为null的记录（若某记录的字段全为null，则该条记录不累加
        count(字段)//累加所有该字段不为null的记录
    ·sum(字段)//累加所有部位null的该字段值
    ·min(字段) //返回所有该字段值中的所有值
    ·avg(字段) //返回指定字段的平均值
    ·IFNULL(字段，'指定显示内容') //如果指定字段为null，则按指定内容显示
·分组查询
    ·SELECT 分组字段 ,
    聚合函数1(字段1), 聚合函数n(字段n)
    FROM 表名
    GROUP BY 分组字段
    ·带分组前条件的分组查询
        SELECT 分组字段 ,
        聚合函数1(字段1), 聚合函数n(字段n)
        FROM 表名
        WHERE 条件
        GROUP BY 分组字段;
    ·带分组后条件的分组查询
        SELECT 分组字段 ,
        聚合函数1(字段1), 聚合函数n(字段n)
        FROM 表名
        WHERE 条件
        GROUP BY 分组字段;
        HAVING 组内条件
        //组内条件，例如(组员人数<3)
·limit子句（方言）
    ·用来限制查询的结果
        LIMIT start,numberLine //从第start开始（下标以0起始），查询最多numberLine行
    分页查询中，查询第n页的m条记录，start=（n-1)*m

六，字符集问题

<pre spellcheck="false" class="md-fences mock-cm md-end-block" lang="" cid="n42" mdtype="fences">·character_set_client：该属性的值为mysql服务器接收客户端数据的编码格式。
    ·在客户端发送给mysql服务器数据时，mysql数据库会把数据当作character_set_client设定的编格式存储数据
·character_set_results：该属性的值为mysql服务器发送给客户端的数据编码格式。
    ·在客户端查询数据库时，mysql数据库会把结果按character_set_results的编码格式发送数据
这样势必会导致乱码问题。
    ·解决方案1：（暂时修改）
        set character_set_client=指定编码格式
        set character_set_results=指定编码格式
    解决方案2：
        在总配置文件中进行配置，永久设置
        [client]
        [mysqld]
        port =3306
        default-character-set=指定编码格式
        //改变client，results，connection的编码格式
</pre>

数据库的备份和恢复

<pre spellcheck="false" class="md-fences mock-cm md-end-block" lang="" cid="n44" mdtype="fences">在未登录模式下进行
·备份数据：
mysqldump -u root -p 数据库名>sql脚本保存路径
·恢复数据
mysql -u root -p 数据库名<sql脚本保存路径
</pre>

七，约束

1，主键约束（唯一标识）

    ·特点：
        ·非空
        ·唯一
        ·被引用（外键引用主键）
    ·指定主键
        ·方式1：谁是主键
            CREATE TABLE [IF NOT EXISTS]表名(
                列名1 列类型 PRIMARY KEY,
                列名2 列类型,
                列名3 列类型,
                列名4 列类型,
                 ……
                );
        方式2：主键是谁
            CREATE TABLE [IF NOT EXISTS]表名(
                列名 列类型,

```
        列名 列类型,
        列名 列类型,
        列名 列类型,
        ...................,
        PRIMARY KEY(列名)
        );
        //把PRIMARY KEY(列名)当作列，修添加和删除一个表的主键
```

·主键自增长

```
· CREATE TABLE [IF NOT EXISTS]表名(
        列名1 列类型 PRIMARY KEY AUTO_INCREMENT,
        列名2 列类型,
        列名3 列类型,
        列名4 列类型,
         ...................
        );
```

AUTO_INCREMENT只能用在int类型列上

不适合使用自然数据作为主键，不适用于群集。使用uuid

2，非空约束和唯一约束

```
CREATE TABLE [IF NOT EXISTS]表名(
        列名1 列类型 NOT NULL,
        列名2 列类型 UNIQUE,
        列名3 列类型,
        列名4 列类型,
         ...................
        );
```

3，外键约束

·外键必须是另一个表中的主键,也可以是本表中的主键

·外键可重复

·外键可以为空

·外键名为外键对象名，用处在于查错和删除外键

·外键取值为主表主键的取值范围

添加外键约束的方式：

```
·CREATE TABLE [IF NOT EXISTS]表名(
        列名 列类型,
        列名 列类型,
        列名 列类型,
        列名 列类型,
        ...................,
        列名n 列类型,
        CONSTRAINT 外键名 FOREIGN KEY(列名n)) REFERENCES dept(主表主键列名))
        );
```

·修改表的方式添加外键

```
  ·CREATE TABLE [IF NOT EXISTS]表名(
        列名 列类型,
        列名 列类型,
        列名 列类型,
        列名 列类型,
        ...................,
        列名n 列类型
```

```
        );
    ALTER TABLE 表名
    ADD CONSTRAINT 外键名 FOREIGN KEY(列名n)) REFERENCES dept(主表主键列名));
```

八，概念模型

1，对象模型：

<pre spellcheck="false" class="md-fences mock-cm md-end-block" lang="" cid="n54" mdtype="fences">在java中是domain，例如User Student
    数据库中的表看成对象模型
  可以双向关联，而且引用的是对象（一条记录），而不是一个主键
  ·is a
  ·has a
    ·1对1
    ·1对多
    ·多对多
  ·use a
</pre>

2，关系模型：

<pre spellcheck="false" class="md-fences mock-cm md-end-block" lang="" cid="n56" mdtype="fences">   ·多对一：
      ·只能多方引入一方，即从表引入主表。而且引用的是主表主键，不是一整行记录（对象）
      ·数据库实现方式
        添加外键约束，将引用字段设为外键
    ·一对一：
      ·只能一方引入一方。
      ·实现方式：
        将从表的主键同时设为外键，这样从表的外键（即主键）有以下特点
          ·非空
          ·唯一
          ·引用主表主键（即从表主键的取值范围为主表主键的范围）
    ·多对多：
      ·实现方式：
        创建中间表（或叫关联表），专门用于存放关系，且字段全部为对象的主键，这样就可以现多对多的关系模型了
</pre>

3，关系模型图

(1)多对一关系模型

(2)一对一关系模型：

4，关系模型-表的创建

(1)多对一关系模型

```
//创建主表
CREATE TABLE dept(
    deptno INT PRIMARY KEY,
    dname VARCHAR(10) NOT NULL
)
//创建从表
CREATE TABLE emp(
    empo INT PRIMARY KEY,
    name varchar(10) NOT NULL,
    job VARCHAR(10),
    deptno INT,
    CONSTRAINT fk_emp_dept FOREIGN KEY(deptno) REFERENCES dept(deptno)
);
```

(2)一对一关系模型

```
CREATE TABLE husband(
    hid INT PRIMARY KEY,
    hname VARCHAR(20)
)

CREATE TABLE wife(
    wid INT PRIMARY key,
    wname varchar(20),
    CONSTRAINT fk_wife_husband FOREIGN KEY(wid) REFERENCES husband(hid)
)
```

(3)多对多关系模型

```
CREATE TABLE student(
    sid INT PRIMARY KEY ,
    sname VARCHAR(20)
);
CREATE TABLE teacher(
    tid INT PRIMARY KEY,
    sname VARCHAR(20)
);

CREATE TABLE stu_tea(
    sid INT,
    tid INT,
    CONSTRAINT fk_student FOREIGN KEY(sid) REFERENCES student(sid),
    const fk_teacher FOREIGN KEY(tid) REFERENCES teacher(tid)
);
```

九，多表查询

1，合并结果集

·语法： （select （字段，……） from 表1） union all (select (字段，……) from 表2)；
·两个查询语句查询出的结果集结构必须完全相同
·对于select *而言，多个表的结构必须完全相同，即也是结果集相同
·使用union all 连接两个查询语句，将查询结果合并，且列名为最前面的查询语句所查询表的字段名
·使用union ， 合并的结果集没有重复记录

2，连表查询

·分类
·内连接
·方言：
·select 字段
from 表1，表2
WHERE 条件；
·结果集为表1，表2结果集的笛卡儿积，列为查询的所有列名
·条件筛选：将有用的记录筛选出
·进行连表查询，结果集的生成步骤：多表的结果集的笛卡儿积-->按条件筛选先前结果集成新的结果集（若有条件）
-->在结果集按查询字段生成结果集
·示例sql语句
SELECT s.sid 学生编号,s.sname 学生姓名,t.tname 执教老师姓名
FROM student s,teacher t,stu_tea st
WHERE s.sid=st.sid AND t.tid=st.tid
·标准：
·SELECT 字段
FORM 表1
inner join 表二 inner join 表三
ON 条件
·自然：
·SELECT 字段
FORM 表1 natural join 表二 natural join 表三
自动匹配，根据查询，将返回结果集中形同列匹配，即筛选出形同字段值的记录，返回新结果集
过程：多表的结果集的笛卡儿积-->筛选笛卡尔结果集中重复列名的字段的字段值相等的记，得到新的结果集
-->在新结果集按查询字段生成结果集
·外连接
·左外连接
·语法
SELECT 字段
FORM 表1 LEFT OUTER JOIN 表二 LEFT OUTER JOIN 表三
ON 条件
·外连接有一主一次，左外连接即左表为主
·外连接的主表中所有的记录无论是否满足条件，都全部进入结果集，当不满足条件时，右使用null补位字段
·右外连接
·语法
SELECT 字段
FORM 表1 RIGHT OUTER JOIN 表二 RIGHT OUTER JOIN 表三
ON 条件
·外连接有一主一次，右外连接即右表为主
·外连接的主表中所有的记录无论是否满足条件，都全部进入结果集，当次表中的数据有不足条件的记录时： （主表的字段必须全部显示，分两种现象）
· 主表有记录在此纪录前没有出现，则主表显示记录，右表null补位

·主表有记录在此纪录前有出现，则主表的此条记录不显示

·全外连接(mysql不支持)

SELECT 字段

FORM 表1 FULL OUTER JOIN 表二 LEFT OUTER JOIN 表三

ON 条件

采用合并结果集语句将左连接和右连接查询语句结合集合实现全外连接的效果

3，子查询

（1）子查询的概念

查询语句中包含查询语句，被包含的查询语句作为需要的结果集给包含查询当作条件使用

（2）被包含查询出现的位置

·SELECT

FROM（位置2）

WHERE (位置1)

·位置1：结果集作为条件

·位置2：结果集作为表

（3）结果集的类型

·单行单列：作为条件出现，如where 列[>,<,>=,<=,=,!=](select 语句)

·多行单列：作为条件出现时，需要使用关键字 ANY ALL IN

·ANY，将结果集中的一项数据作为条件判断依据，有任意一个符合即可

·ALL，将结果集中的一项数据作为条件判断依据，所有符合即可

·IN， 将结果集中的一项数据作为条件判断依据，有任意一个符合即可

·多行多列：

·将多个字段作为条件使用

·SELECT 字段 FROM 表 WHERE (字段1，字段2，……) IN（SELECT 字段1，字段2，……WHERE 条件)

·用多个子查询分解条件

WHERE

字段1=（SELECT 字段1 WHERE 条件）

字段2=（SELECT 字段1 WHERE 条件）

将结果集作为表在from后使用