**0 相关知识**

* [MYSQL逻辑架构](https://blog.csdn.net/chaoyue1861/article/details/80468773)
* [MYSQL底层原理](https://blog.csdn.net/GitChat/article/details/78787837?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.control&dist_request_id=&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-2.control)
* [MySQL底层架构原理，工作流程和存储引擎的数据结构讲解](https://blog.csdn.net/m0_38075425/article/details/82256315?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_baidulandingword-0&spm=1001.2101.3001.4242)

**1 用户管理操作**

**1.1 添加用户**

* create user username identified by 'password';

参数：用户名：username 密码：password

用户数据存储在mysql.user表内

**1.2 用户授权**

* grant privilegesCode on dbName.tableName to username@host identified by 'password';

将Dbname数据库的所有操作权限都授权给了用户username。

参数：

**privilegesCode** 权限类型  
常用的权限类型

| **参数** | **含义** |
| --- | --- |
| all privileges | 所有权限 |
| select | 读取权限 |
| delete | 删除权限 |
| update | 更新权限 |
| create | 创建权限 |
| drop | 删除数据库、数据表权限 |

**dbName.tableName** 授权的库或特定表

| **参数** | **含义** |
| --- | --- |
| . | 授予该数据库服务器所有数据库的权限 |
| dbName.\* | 授予dbName数据库所有表的权限 |
| dbName.dbTable | 授予数据库dbName中dbTable表的权限 |

**username@’host’** 授予的用户以及允许该用户登录的IP地址  
| 参数 | 含义 |  
| ————– | ———————- |  
| localhost | 只允许该用户在本地登录，不能远程登录 |  
| % | 允许在除本机之外的任何一台机器远程登录 |  
| 192.168.52.32 | 具体的IP表示只允许该用户从特定IP登录 |

* flush privileges;

刷新权限变更

* show grants for 'username';

查看用户的已有权限

用例：

|  |  |
| --- | --- |
| 1 2 Copy | grant all privileges on zhangsanDb.\* to zhangsan@'%' identified by 'zhangsan'; flush privileges; Copy |

上面的语句将zhangsanDb数据库的所有操作权限都授权给了用户zhangsan

* REVOKE DELETE ON \*.\* FROM 'test'@'localhost';

取消该用户的删除权限

用户权限数据存储在mysql.db表内

**1.3 修改密码**

|  |  |
| --- | --- |
| 1 2 Copy | update mysql.user set password = password('newpassword') where user = 'username' and host = '%';  flush privileges; Copy |

**1.4 删除用户**

* drop user zhangsan@'%';

**1.5 常用命令组**

**创建用户并授予指定数据库全部权限：适用于Web应用创建MySQL用户**

|  |  |
| --- | --- |
| 1 2 3 Copy | create user zhangsan identified by 'zhangsan'; grant all privileges on zhangsanDb.\* to zhangsan@'%' identified by 'zhangsan'; flush privileges; Copy |

创建了用户zhangsan，并将数据库zhangsanDB的所有权限授予zhangsan。如果要使zhangsan可以从本机登录，那么可以多赋予localhost权限：

|  |  |
| --- | --- |
| 1 Copy | grant all privileges on zhangsanDb.\* to zhangsan@'localhost' identified by 'zhangsan'; Copy |

**2 数据库操作**

| **命令** | **含义** |
| --- | --- |
| show database; | 查看所有的数据库 |
| create database DBname; | 创建该数据库 |
| drop DBname; | 删除该数据库 |
| use DBname; | 使用调用该数据库 |

**3 表操作**

**3.1 表的基础操作**

* show tables;  
  查看所有的表
* SHOW TABLE STATUS;  
  查看所有的表信息（包括视图）
* create table TBname(mode);

创建一个表

例如  
create table n(id INT, name VARCHAR(10));

* create table TBname select \* from TBname;  
  直接将查询结果导入或复制到新创建的表
* create table TBname like TBname;  
  新创建的表与一个存在的表的数据结构类似
* create temporay table TBname(mode);  
  创建一个临时表

临时表将在你连接MySQL期间存在。当断开连接时，MySQL将自动删除表并释放所用的空间。也可手动删除。

* create temporary table TBname select \* from TBname;  
  直接将查询结果导入或复制到新创建的临时表
* drop table if exists TBname;  
  删除一个存在表
* alter table TBname rename TBname;
* 或
* rename TBname to TBname;

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 Copy | 更改存在表的名称      - ```   desc TBname;  describe TBname;  show columns in TBname;  show columns from TBname;  explain TBname; Copy |

* 查看表的结构(以上五条语句效果相同）
* Copy
* show create table TBname;  
  查看表的创建语句

**3.2 表的结构操作**

| **语句** | **含义** |
| --- | --- |
| alter table TBname add Fieldname mode; | 添加字段 |
| alter table TBname drop Fieldname; | 删除字段 |
| alter table TBname change Fieldname mode; | 更改字段属性 |
| alter table TBname change Fieldname Fieldname mode; | 更改字段名与属性 |

**3.3 表的数据操作**

* 增加数据

INSERT INTO n VALUES (1, 'tom', '23'), (2, 'john', '22');

INSERT INTO n SELECT \* FROM n; 把数据复制一遍重新插入

* 删除数据

DELETE FROM n WHERE id = 2;

* 更改数据

UPDATE n SET name = 'tom' WHERE id = 2;

* 数据查找

SELECT \* FROM n WHERE name LIKE '%h%';

* 数据排序(反序)

SELECT \* FROM n ORDER BY name, id DESC ;

增删改查请看：<https://www.cnblogs.com/heyangblog/p/7624645.html>

**4 键**

**4.1 添加主键**

* ALTER TABLE TBname ADD PRIMARY KEY (id);
* ALTER TABLE TBname ADD CONSTRAINT pk\_n PRIMARY KEY (id);

添加主键的同时自定义键名

**4.2 删除主键**

* ALTER TABLE TBname DROP PRIMARY KEY ;

**4.3 添加外键**

* ALTER TABLE TBname ADD FOREIGN KEY (id) REFERENCES TBname(id);

自动生成键名m\_ibfk\_1

* ALTER TABLE TBname ADD CONSTRAINT fk\_id FOREIGN KEY (id) REFERENCES TBname(id);

使用定义的键名fk\_id

**4.4 删除外键**

* ALTER TABLE TBname DROP FOREIGN KEY fk\_id;

**4.5 修改外键**

* ALTER TABLE TBname DROP FOREIGN KEY fk\_id;

ADD CONSTRAINT fk\_id2 FOREIGN KEY (id) REFERENCES TBname(id);

删除之后从新建

**4.6 添加唯一键**

* ALTER TABLE TBname ADD UNIQUE (name);
* ALTER TABLE TBname ADD UNIQUE u\_name (name);
* ALTER TABLE TBname ADD UNIQUE INDEX u\_name (name);
* ALTER TABLE TBname ADD CONSTRAINT u\_name UNIQUE (name);
* CREATE UNIQUE INDEX u\_name ON TBname(name);

**4.7 添加索引**

* ALTER TABLE TBname ADD INDEX (age);
* ALTER TABLE TBname ADD INDEX i\_age (age);
* CREATE INDEX i\_age ON TBname(age);

**4.8 删除索引或唯一键**

* DROP INDEX u\_name ON n;
* DROP INDEX i\_age ON n;

**5 视图**

**5.1 创建视图**

* CREATE VIEW v AS SELECT id, name FROM n;
* CREATE VIEW v(id, name) AS SELECT id, name FROM n;

**5.2查看视图**

* SELECT \* FROM v;
* DESC v;

与表操作类似

**5.3查看创建视图语句**

* SHOW CREATE VIEW v;

**5.4 更改视图**

* CREATE OR REPLACE VIEW v AS SELECT name, age FROM n;
* ALTER VIEW v AS SELECT name FROM n ;

**5.5 删除视图**

* DROP VIEW IF EXISTS v;

**6 联接**

**6.1 内联接**

* SELECT \* FROM m INNER JOIN n ON m.id = n.id;

**6.2 外连接**

* SELECT \* FROM m LEFT JOIN n ON m.id = n.id;左外连接
* SELECT \* FROM m RIGHT JOIN n ON m.id = n.id;右外连接

**6.3 交叉联接**

* SELECT \* FROM m CROSS JOIN n; 标准写法
* SELECT \* FROM m, n;

**6.4 类似全连接full join的联接用法**

* SELECT id,name FROM m
* UNION
* SELECT id,name FROM n;

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 Copy | ## 7 函数  ### 7.1 聚合函数  | 语句 | 含义 | | ---------------------------------- | ------ | | SELECT count(id) AS total FROM n; | 总数 | | SELECT sum(age) AS all\_age FROM n; | 总和 | | SELECT avg(age) AS all\_age FROM n; | 平均值 | | SELECT max(age) AS all\_age FROM n; | 最大值 | | SELECT min(age) AS all\_age FROM n; | 最小值 |  ### 7.2 数学函数  | 语句 | 含义 | | --------------------------------- | ------------------------------------------------------ | | SELECT abs(-5); | 绝对值 | | SELECT bin(15), oct(15), hex(15); | 二进制，八进制，十六进制 | | SELECT pi(); | 圆周率3.141593 | | SELECT ceil(5.5); | 大于x的最小整数值6 | | SELECT floor(5.5); | 小于x的最大整数值5 | | SELECT greatest(3,1,4,1,5,9,2,6); | 返回集合中最大的值9 | | SELECT least(3,1,4,1,5,9,2,6); | 返回集合中最小的值1 | | SELECT mod(5,3); | 余数2 | | SELECT rand(); | 返回０到１内的随机值，每次不一样 | | SELECT rand(5); | 提供一个参数(种子)使RAND()随机数生成器生成一个指定的值 | | SELECT round(1415.1415); | 四舍五入1415 | | SELECT round(1415.1415, 3); | 四舍五入三位数1415.142 | | SELECT round(1415.1415, -1); | 四舍五入整数位数1420 | | SELECT truncate(1415.1415, 3); | 截短为3位小数1415.141 | | SELECT truncate(1415.1415, -1); | 截短为-1位小数1410 | | SELECT sign(-5); | 符号的值负数-1 | | SELECT sign(5); | 符号的值正数1 | | SELECT sqrt(9); | 平方根3 |  ### 7.3 字符串函数  | 语句 | 含义 | | ------------------------------------------- | ------------------------------------------------------- | | SELECT concat('a', 'p', 'p', 'le'); | 连接字符串-apple | | SELECT concat\_ws(',', 'a', 'p', 'p', 'le'); | 连接用','分割字符串-a,p,p,le | | SELECT insert('chinese', 3, 2, 'IN'); | 将字符串'chinese'从3位置开始的2个字符替换为'IN'-chINese | | SELECT left('chinese', 4); | 返回字符串'chinese'左边的4个字符-chin | | SELECT right('chinese', 3); | 返回字符串'chinese'右边的3个字符-ese | | SELECT substring('chinese', 3); | 返回字符串'chinese'第三个字符之后的子字符串-inese | | SELECT substring('chinese', -3); | 返回字符串'chinese'倒数第三个字符之后的子字符串-ese | | SELECT substring('chinese', 3, 2); | 返回字符串'chinese'第三个字符之后的两个字符-in | | SELECT trim(' chinese '); | 切割字符串' chinese '两边的空字符-'chinese' | | SELECT ltrim(' chinese '); | 切割字符串' chinese '两边的空字符-'chinese ' | | SELECT rtrim(' chinese '); | 切割字符串' chinese '两边的空字符-' chinese' | | SELECT repeat('boy', 3); | 重复字符'boy'三次-'boyboyboy' | | SELECT reverse('chinese'); | 反向排序-'esenihc' | | SELECT length('chinese'); | 返回字符串的长度-7 | | SELECT upper('chINese'), lower('chINese'); | 大写小写 CHINESE chinese | | SELECT ucase('chINese'), lcase('chINese'); | 大写小写 CHINESE chinese | | SELECT position('i' IN 'chinese'); | 返回'i'在'chinese'的第一个位置-3 | | SELECT position('e' IN 'chinese'); | 返回'i'在'chinese'的第一个位置-5 | | SELECT strcmp('abc', 'abd'); | 比较字符串，第一个参数小于第二个返回负数- -1 | | SELECT strcmp('abc', 'abb'); | 比较字符串，第一个参数大于第二个返回正数- 1 |  ### 7.4 时间函数  | 语句 | 含义 | | ------------------------------------------------------------ | -------------------------------------------- | | SELECT current\_date, current\_time, now(); | 2018-01-13 12:33:43 2018-01-13 12:33:43 | | SELECT hour(current\_time), minute(current\_time), second(current\_time); | 12 31 34 | | SELECT year(current\_date), month(current\_date), week(current\_date); | 2018 1 1 | | SELECT quarter(current\_date); | 1 | | SELECT monthname(current\_date), dayname(current\_date); | January Saturday | | SELECT dayofweek(current\_date), dayofmonth(current\_date), dayofyear(current\_date); | 7 13 13 |  ### 7.5 控制流函数  - `SELECT if(3>2, 't', 'f'), if(3<2, 't', 'f');`    t f  - `SELECT ifnull(NULL, 't'), ifnull(2, 't');`    t 2  - `SELECT isnull(1), isnull(1/0);`    0 1 是null返回1，不是null返回0  - `SELECT nullif('a', 'a'), nullif('a', 'b');`    null a 参数相同或成立返回null，不同或不成立则返回第一个参数  - ```  SELECT CASE 2  WHEN 1 THEN 'first'  WHEN 2 THEN 'second'  WHEN 3 THEN 'third'  ELSE 'other'  END ;  Copy |

* second
* > 这一块不是完全看不懂吗！
* Copy

**系统信息函数**

| **语句** | **含义** |
| --- | --- |
| SELECT database(); | 当前数据库名-test |
| SELECT connection\_id(); | 当前用户id-306 |
| SELECT user(); | 当前用户-root@localhost |
| SELECT version(); | 当前mysql版本 |
| SELECT found\_rows(); | 返回上次查询的检索行数 |

**8 存储过程**

**8.1 创建存储过程**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 Copy | DELIMITER // # 无参数 CREATE PROCEDURE getDates()  BEGIN  SELECT \* FROM test ;  END // CREATE PROCEDURE getDates\_2(IN id INT) # in参数  BEGIN  SELECT \* FROM test WHERE a = id;  END // CREATE PROCEDURE getDates\_3(OUT sum INT) # out参数  BEGIN  SET sum = (SELECT count(\*) FROM test);  END // CREATE PROCEDURE getDates\_4(INOUT i INT) # inout参数  BEGIN  SET i = i + 1;  END // DELIMITER ; Copy |

**8.2 删除存储过程**

* DROP PROCEDURE IF EXISTS getDates;

**8.3 修改存储过程的特性**

* ALTER PROCEDURE getDates MODIFIES SQL DATA ;

**8.4 查看存储过程**

* SHOW PROCEDURE STATUS LIKE 'getDates';

状态

* SHOW CREATE PROCEDURE getDates\_3;

语句

**8.5 调用存储过程**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 Copy | CALL getDates(); CALL getDates\_2(1); CALL getDates\_3(@s); SELECT @s; SET @i = 1; CALL getDates\_4(@i); SELECT @i; # @i = 2 Copy |

**9 数据库安全**

**9.1 数据库备份**

* mysqldump -u root -p db\_name > file.sql
* mysqldump -u root -p db\_name table\_name > file.sql
* Copy

**9.2数据库还原**

* mysql -u root -p < C:\file.sql