

**一、TRIGGER**

create TRIGGER xxx

after/before insert

on 表xx for each row

begin

xxxxxx;

end;

查看触发器：show triggers like “”；

select \* from information\_schema.triggers;

show create grigger

**二、procedure**

create procedure xxx（IN/OUT/INOUT 参数名 参数类型）

begin

xxxxxx;

DECLARE cursor\_name CURSOR FOR

select from where;

DECLARE exit HANDLER FOR NOT FOUND CLOSE cursor\_name;

open cursor\_name;

fetch cursor\_name into A, B, C;

close cursor\_name;

end;

调用时 call xx

**三、case when**

case

when 条件 then 表达式/结果值

when then

else

end;

**四、表中新增一列属性**

ALTER TABLE “表名” ADD“字段名” “...属性”AFTER “字段名”

**五、WITH ROLLUP**

是用来在分组统计数据的基础上再进行统计汇总，即用来得到group by的汇总信息；

**六、截取日期中的年份、月份、日期**

「1」year(日期) / month（日期） / day（日期）

「2」DATE\_FORMAT(TXN\_DATE, "%m")

**七、类型转换 cast**

CAST函数语法规则是：Cast(字段名 as 转换的类型 )，其中类型可以为：

CHAR[(N)] 字符型

DATE 日期型

DATETIME 日期和时间型

DECIMAL float型

SIGNED int

TIME 时间型

**八、IF ELSE**

if xx then ；

elseif xx then;

else xx;

endif;

**九、函数**

create function xxx(参数名 参数类型) returns 类型

begin

end;

调用时 select 函数名

**十、replace 主键冲突时替换原始数据**

replace into 表名（属性1，属性2） values （’xx’，‘yyy’）

**十一、删除操作**

（1）delete

delete from 表名 where

（2）truncate

truncate 表名 = delete \* from 表名

（3）区别

runcate:会重置 auto\_increment (自增)起始值

delete:不会重置

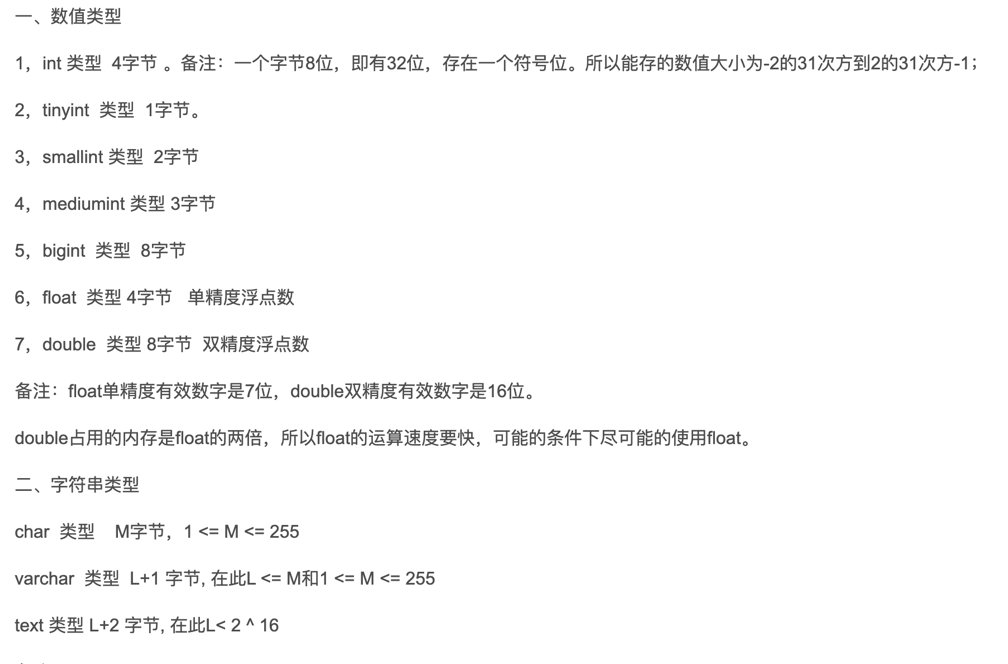
**十二、字符集**

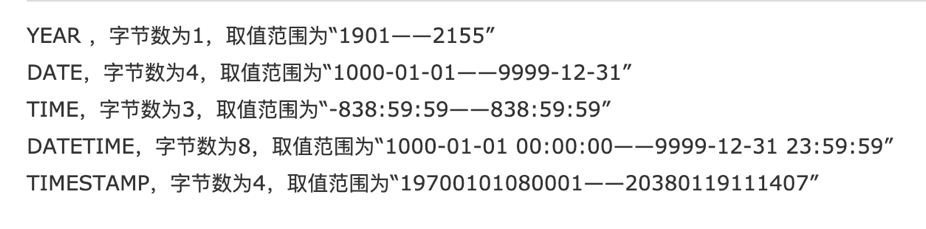
show character set——查看MySQL服务器实例支持的字符集信息

show variables like '%char%';——查看MySQL服务器实例使用的字符集信息

show full columns from 表名;——查看该表各列的字符集

**十三、数据类型**





enum枚举类型和set集合类型。 enum类型的字段类似于单选按钮的功能，一个enum类型的数据最多可以包含65535个元素。 set 类型的字段类似于复选框的功能，一个set类型的数据最多可以包含64个元素。

**十四、通配符**

通配符:

%:代表任意多个字符

\_(下划线):代表任意的一个字符

**十五、转义字符** escape

如果要查询的字符串本身就含有通配符，此时就需要用ESCAPE关键字，对通配符进行转义

转义后：select \* from user where username like '/\_nihao' escape '/',意思就是说/之后的\_不作为通配符

**十六、group by**

（1）如果是GROUP BY ROLLUP(A,B,C)，首先会对(A,B,C)进行GROUP BY，然后对(A,B)进行GROUP BY，然后是(A) 进行GROUP BY，最后对全表进行GROUP BY操作

（2）如果是GROUP BY CUBE(A,B,C)，首先会对(A,B,C)进行 GROUP BY，然后是(A,B)，(A,C)，(B,C)，(A)， (B)， (C)，最后对全表进行GROUP BY操作

（3）

rollup(a,b) 统计列包含:(a,b)、(a)、() rollup(a,b,c) 统计列包含:(a,b,c)、(a,b)、(a)、() ......以此类推ing......

cube(a,b) 统计列包含:(a,b)、(a)、(b)、() cube(a,b,c) 统计列包含:(a,b,c)、(a,b)、(a,c)、(b,c)、 (a)、(b)、(c)、()......以此类推ing......

̈ CUBE在ROLLUP的基础上进一步从各种维度上给出细 化的统计汇总结果(可以用ROLLUP查询表示CUBE)

**十七、存储过程优点**

（1）通常存储过程有助于提高应用程序的性能。当创建，存储过程被编译之后，就存储在数据库中。 但是，MySQL实现的存储过程略有不同，MySQL存储过程按需编译。

（2）存储过程有助于减少应用程序和数据库服务器之间的流量，因为应用程序不必发送多个冗长的SQL语句，而只能发送存储过程的名称和参数。

（3）存储的程序对任何应用程序都是可重用的和透明的。存储过程将数据库接口暴露给所有应用程序，以便开发人员不必开发存储过程中已支持的功能。

（4）存储的程序是安全的。 数据库管理员可以向访问数据库中存储过程的应用程序授予适当的权限，而不向基础数据库表提供任何权限。

**十八、InnoDB引擎支持外键级联选项 **

CASCADE；SET NULL；NO ACTION；RESTRICT

**十九、隐式执行commit的语句**



**二十、事务四大特性**

原子性、一致性、隔离性、永久性

**二十一、事务隔离级别 从上到下隔离级别升高-必考—研究作业7！**

read uncommitted读取未提交的数据

可读取其他事务未提交的数据

脏读、不可重复读、幻读

read committed读取提交的数据

可读取已提交的数据

可避免脏读，但可能出现:不可重复读、幻读

repeatable read 可重复读

确保同一事务内相同查询语句执行结果总是相同(默认)

可以避免脏读、不可重复读，可能出现幻读

serializable 串行化

强制事务排序，事务须依次顺序执行

可以避免幻读



查看当前会话的事务隔离级别 select @@session.tx\_isolation;

查看全局事务隔离级别 select @@global.tx\_isolation;

**二十二、**

脏读：一个事务读取到另一个事务未提交的数据

不可重复读：同一事务中，两条相同的查询语句查询结果不一致；当一个事务访问数据时，另一个事务对该数据进行修改并提交，导致第一个事务两次读到的数据不一样。

幻读：当前事务读不到其他事务已提交的修改

**二十三、JDBC**

【1】．加载JDBC驱动程序

加载驱动程序的方法是使用java.lang.Class类的静态方法forName(String className)。

Class.forName("com.mysql.jdbc.Driver");

若加载成功，系统将加载的驱动程序注册到DriverManager类中。如果加载失败，将抛出ClassNotFoundException异常，即未找到指定的驱动类。

try {

Class.forName(“com.mysql.jdbc.Driver”);

} catch (ClassNotFoundException ex) {

System.out.println("加载数据库驱动时抛出异常!");

ex.printStackTrace();

}

【2】创建数据库连接

与数据库建立连接的方法是调用DriverManager类的getConnection()方法。

getConnection()方法的返回值类型为java.sql.Connection，如果连接数据库失败，将抛出SQLException异常，其方法调用如下：

Connection conn = DriverManager.getConnection(String url, String userName, String password);

依次指定要连接数据库的路径、用户名及密码，即可创建数据库连接对象。

try {

String url ="jdbc:mysql://localhost:3306/bookstore";

String user="root"; //访问数据库的用户名

String password="123456"; //访问数据库的密码

Connection conn= DriverManager.getConnection(url,user,password);

System.out.println("连接数据库成功！");

} catch (SQLException ex) {

System.out.println("连接数据库失败！");

ex.printStackTrace();

}

【3】创建statement对象

（1）创建Statement对象

利用Connection接口的createStatement()方法可以创建Statement对象，用来执行静态的SQL语句。

代码如下：

Statement stmt=conn.createStatement(); //conn为数据库连接对象

（2）创建PreparedStatement对象

利用Connection接口的prepareStatement(String sql)方法可以创建PreparedStatement对象，用来执行动态的SQL语句。

创建PreparedStatement对象pstmt的代码如下：

String sql = "select \* from users where u\_id>? and u\_sex=?";

PreparedStatement pstmt = conn.prepareStatement(sql);

在执行该SQL语句前，需要对每个输入参数进行设置，设置参数的语法格式如下：pstmt.setXxx(position,value);

【4】执行SQL语句

创建Statement对象后，就可以利用该对象的相应方法来执行SQL语句，实现对数据库的具体操作。

Statement对象的常用方法有executeQuery()、executeUpdate()等。

（1）ResultSet executeQuery(String sql)方法：该方法用于执行产生单个结果集的SQL语句，如SELECT语句，该方法返回一个结果集ResultSet对象。

（2）int executeUpdate(String sql)方法：该方法用于执行 INSERT、UPDATE 或 DELETE 语句以及SQL DDL（数据定义语言）语句。该方法的返回值是一个整数，表示受影响的行数。对于 CREATE TABLE 或 DROP TABLE 等不操作数据行的语句，返回值为0。

【5】处理返回结果

使用Statement对象的executeQuery()方法执行一条SELECT语句后，会返回一个ResultSet对象。

ResultSet对象保存查询的结果集，调用ResultSet对象的相应方法就可以对结果集中的数据行进行处理。

（1）boolean next()：ResultSet对象具有指向当前数据行的指针，指针最初指向第一行之前，使用next()方法可以将指针移动到下一行。如果没有下一行时，则返回false。

（2）getXxx(列名或列索引)：该方法可获取所在行指定列的值。其中，Xxx指的是列的数据类型。若使用列名作为参数，则getString(“name”)，表示获取当前行列名为“name”的列值。列索引值从1开始编号，如第2列对应的索引值为2。

【6】批处理

（1）使用addBatch(sql)方法，将需要执行的SQL命令添加到批处理中。

（2）使用executeBatch()方法，执行批处理命令。

（3）使用clearBatch()方法，清空批处理队列。

批量执行静态SQL

使用Statement对象的addBatch()方法可以批量执行静态SQL。

优点是可以向数据库发送多条不同的SQL语句。

缺点是SQL语句没有预编译，执行效率较低，并且当向数据库发送多条语句相同，但仅参数不同的SQL语句时，需重复使用多条相同的SQL语句。

批量执行动态SQL

批量执行动态SQL，需要使用PreparedStatement对象的addBatch()方法来实现批处理。

优点是发送的是预编译后的SQL语句，执行效率高。

缺点是只能应用在SQL语句相同，但参数不同的批处理中。因此此种形式的批处理经常用于在同一个表中批量更新表中的数据。

混合模式

使用PreparedStatement对象的addBatch()方法还可以实现混合模式的批处理，既可以执行批量执行动态SQL，同时也可以批量执行静态SQL。

【7】游标

在数据查询、数据更新事务中，如果需要在结果集中前后移动或显示结果集指定的一条记录的时候，就要用到游动查询，这时应使用带参数的createStatement()方法创建语句对象，其语法格式如下：

Statement stmt=conn.createStatement(int type，int concurrency);

①type的取值决定滚动方式，其取值为：

ResultSet.TYPE\_FORWORD\_ONLY：结果集的游标只能向下滚动。

ResultSet.TYPE\_SCROLL\_INSENSITIVE：结果集的游标可以上下移动，当数据库变化时，当前结果集不变。

ResultSet.TYPE\_SCROLL\_SENSITIVE：返回可滚动的结果集，当数据库变化时，当前结果集同步改变。

②concurrency的取值决定是否可以用结果集更新数据库，其取值为：

ResultSet.CONCUR\_READ\_ONLY：不能用结果集更新数据库中的表。

ResultSet.CONCUR\_UPDATETABLE：能用结果集更新数据库中的表。

【8】数据源

数据源（Data Source）是目前Web开发中获取数据库连接的首选方法。这种方法是首先创建一个数据源对象，由数据源对象事先建立若干连接对象，通过连接池管理这些连接对象。

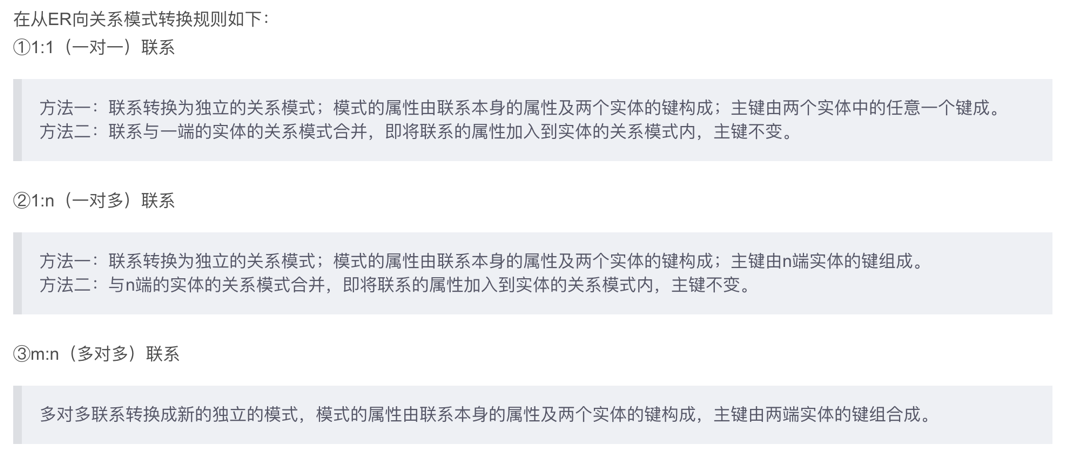
JNDI是一种将名称和对象绑定的技术，对象工厂负责创建对象，这些对象都和唯一的名称绑定，应用程序可以通过名称来获得某个对象的访问。

MySQL游标为只读，不可滚动和敏感。

除了Count(\*)之外聚合函数忽略列值为NULL的行

Union ALL选项表示保留结果集中的重复记录，默认时系统自动删除重复记录

提高运行效率：降低范式，增加冗余，少用触 发器，多用存储过程



**做题整理**

【1】RANK()

    在计算排序时，若存在相同位次，会跳过之后的位次。

    例如，有3条排在第1位时，排序为：1，1，1，4······

【2】DENSE\_RANK()

    这就是题目中所用到的函数，在计算排序时，若存在相同位次，不会跳过之后的位次。

    例如，有3条排在第1位时，排序为：1，1，1，2······

【3】ROW\_NUMBER()

    这个函数赋予唯一的连续位次。

    例如，有3条排在第1位时，排序为：1，2，3，4······

窗口函数用法：

<窗口函数> OVER ( [PARTITION BY <列清单> ]

ORDER BY <排序用列清单> ）

【4】字符函数

[1]char\_length(str)

（1）计算单位：字符

（2）不管汉字还是数字或者是字母都算是一个字符

[2]length(str)

（1）计算单位：字节

（2）utf8编码：一个汉字三个字节，一个数字或字母一个字节。

（3）gbk编码：一个汉字两个字节，一个数字或字母一个字节。

concat group\_concat concat\_was

（1）concat(str1, str2,...)

返回结果为连接参数产生的字符串，如果有任何一个参数为null，则返回值为null。

（2）concat\_ws(separator, str1, str2, ...)

第一个参数指定分隔符。需要注意的是分隔符不能为null，如果为null，则返回结果为null。

（3）group\_concat()函数

在有group by的查询语句中，select指定的字段要么就包含在group by语句的后面，作为分组的依据，要么就包含在聚合函数中。

先group 再 concat

lpad/rpad

lpad用制定字符实现左填充长度

rpad用制定字符实现左填充长度

lpad(“被填充的字符串”,返回被填充后的长度,”用来填充的字符”)

instr

返回子串第一次出现的索引位置

tirm

去除字符串左右两侧的制定字符

trim(“需要被去除的字符”，”字符串”)

replace

replace(“字符串”，“被替换的元素”，“替换后的元素”)

【5】数学函数

（1）round四舍五入

round（数字，保留的小数位数）

（2）ceil向上取整

（3）floor向下取整

（4）truncate截断

truncate（小数，保留位数）

（5）mod取模

【6】日期函数

（1）now（）当前系统日期、时间

（2）curdate（）当前日期

（3）curtime()当前时间

（4）获取指定部分

年year(日期)-月month()-日

返回英文：yearname()

（5）str\_to\_date()字符串转换为日期

（6）date\_to\_str()日期转换为字符串

（7）

