**SQL**

第一部分 SQL基础

一、数据类型

1、整数数据类型

2、浮点数据类型

3、二进制数据类型

4、逻辑数据类型

5、字符数据类型

6、文本和图形数据类型

7、日期和时间数据类型

8、货币数据类型

9、特定数据类型

10、用户自定义数据类型

第二部分 语法

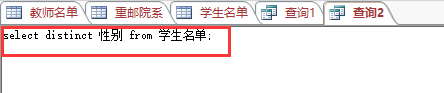
注意，在数据库中是没办法直接同时执行两条以上的SQL代码的，但是，在一条语句中的多个操作是可以同时执行的。但是使用access VBA代码以及在oracle中使用存储过程是可以实现的。

一、关于Select

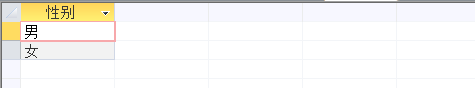
1、select的作用就是从数据库中选择数据；格式就是在select后面添加自己需要选择的内容的key值，如name、sex等；例如，select \* from table\_name就是从数据库中获取表，要是select A,B \* from table\_name就是从数据库中获取表table\_name中的A,B的信息。

二、关于SELECT DISTINCT 语句

Select distinct用于返回唯一不同的值。在一个表中，有可能会有很多重复的值，有时候只需要返回那些不一样的值，这时候就可以使用这个语句来获取数据。



结果如下：



1. where子句

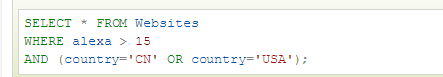
1、Where子句就是用于过滤记录，筛选出那些符合指定标准的数据。

2、对于文本，需要用单引号括起来，但是对于数值字段就不需要用引号来进行处理。

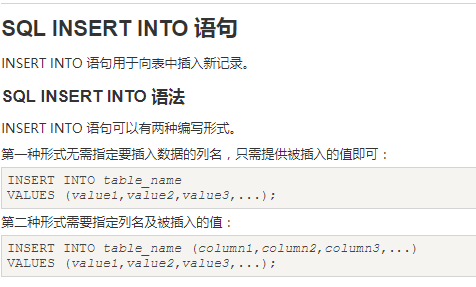


1. AND & OR 运算符

这两个运算符用于基于一个以上的条件对进行过滤。此处，二者的意义就是“和”以及“或”的意思。并且这两个运算符是可以同时一起使用的，也就是进行组合使用。



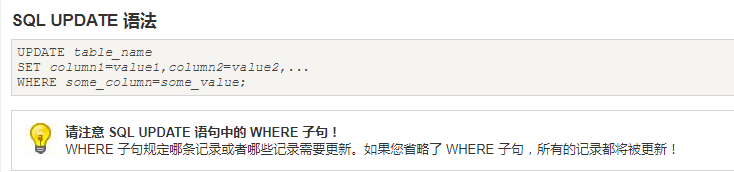
1. insert into
2. 用于向表中插入新的数据，就是新增加一行。要是插入的数据和表格中原有的数据是一一对应的，那么可以省略列名，直接写values的列表。



注意在插入的时候最好还是使用第二种方法，这样更加清楚。在access中，现在可以使用追加按钮直接添加实现这个插入的功能。

1. update语句

用于更新表中已经存在的数据。



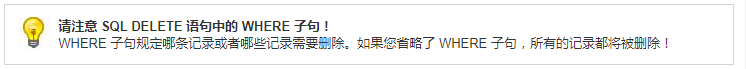
Update语句的使用就是先用update指明是进行数据更新的操作，然后用set指明是更新哪一些数据以及更新之后的值是什么，最后在使用where进行定位。

有一点特别重要的是，如果没有使用where子句，那么上面的更新是针对整个表格中符合相关标准的所有的列的替换，因此需要特别注意。

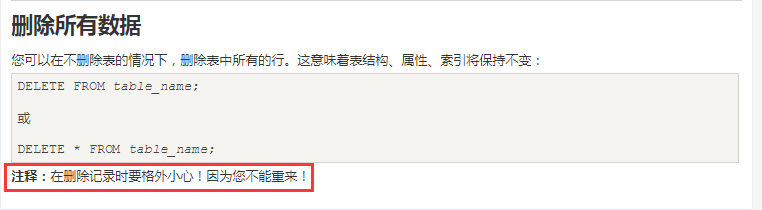
注意：在SQL语句的使用中，数字类型的值是不需要也不能使用引号的，其余的文本类的使用单引号。

1. DELETE语句

Delete语句用于删除表中的记录，每次删除是以行为单位的。



同样需要注意的是，delete和之前的更新是一样的，要是没有使用where子句规定删除哪的是哪一条或者那几条需要删除，那么就会删除所有数据。

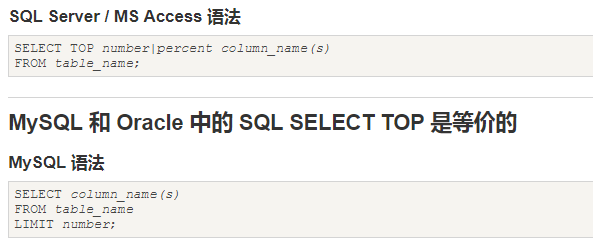


第三部分 SQL高级教程

一、 SELECT TOP, LIMIT, ROWNUM 子句

1、SELECT TOP 子句

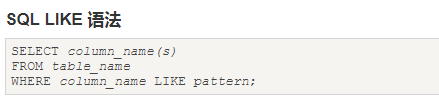
用于规定要返回的记录的数目，这个子句对于拥有数千条记录的大型表来说是比较有用的。但是，这个语句并不是所有的数据库都支持，在使用的时候需要先进行验证。





1. LIKE操作符

Like操作符用于在where子句中搜索列中的指定模式。



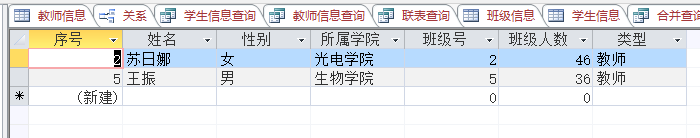
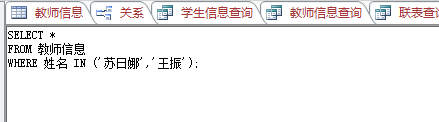
1. 通配符

可用于替代字符串中的任何其他字符，在SQL中是和SQL LIKE 操作符一起使用的，通配符用于搜索表中的数据。

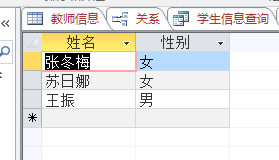
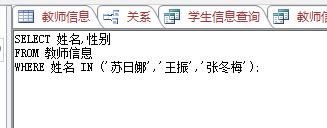


1. IN操作符

允许在where子句中规定多个值。



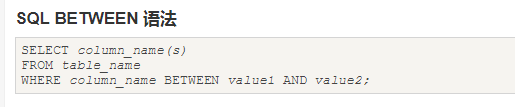
上面的写法是将含有in后面括号中信息的所有的行都选择出来，也可以像下面一样指明选择符合条件的那几列：

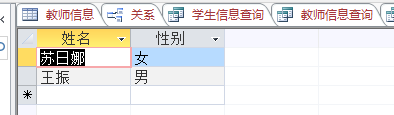
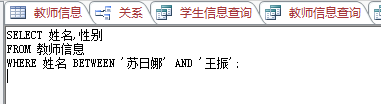


这个例子的意思就是：选择出in中姓名所在行的姓名和性别。

1. BETWEEN操作符

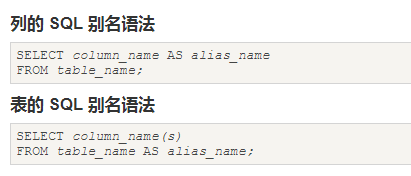
Between操作符用于选取介于两个值之间的数据范围的值，这些值可以是数字、文本、日期。





1. 别名

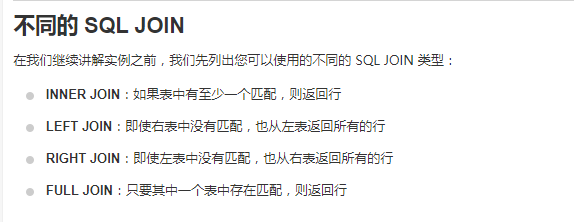
使用SQL可以为表名称和列名称指定别名，目的是使列或者行的可读性更好。



关于联表查询

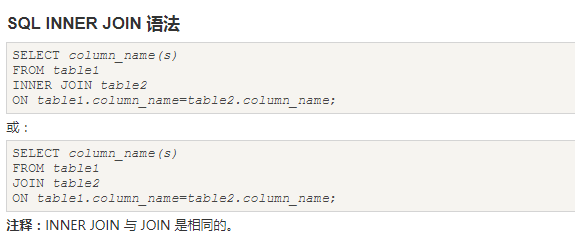
1. 连接join

Join子句用于把来自两个或多个表的行结合起来，基于这些表的共同字段。



1. inner join关键字

Inner join关键字在表中存在至少一个匹配的时候返回行。



Innerjoin和join是没有区别的，join默认的就是内连接。上面所说的匹配就是指的是数据类型和数据的值完全相同才算是匹配。

1. left join

以左右两张表为例，left join就是以左边的表为基础，然后在此基础上将右边的表中符合on后面条件的列返回添加到左边的表上，要是右边表中没有的内容就是null。其中的on就是讲表连接起来的固定用法。



1. full outer join关键字

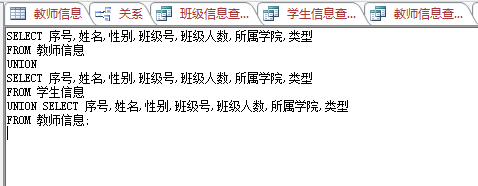
Full outer join关键字可以看做是取的表的并集。也就是说，在这个语句会返回表中所有的行，不管是是否匹配。在返回的结果中是显示出表的所有行，如果相对应的位置没有匹配，也就是hi没有值，那么就会被null填充。



十一、关于UNION组合查询

合并两个或多个select语句的结果集。

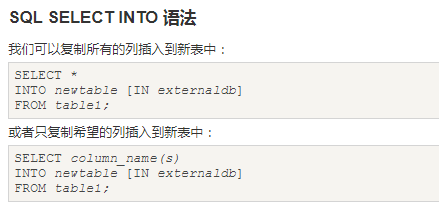
联合查询语句要求查询select语句的列数相同，并且每一列的数据类型也要相似才行。这是属于组合二查询的限制，更加确切的说应该是叫做合并查询结果在一个结果集中。这里的合并就是单纯的将查询结果合并到了一起，并不能称作联表查询。



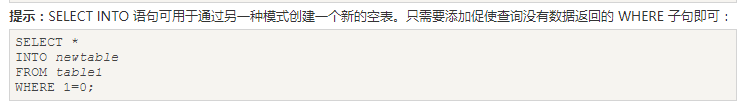
上面的这些SQL语句只是将这几个select的查询结果合并到了一起，并不是按照我们自己想要的标准从不同的表中获取到我们需要的数据。

1. select into 语句

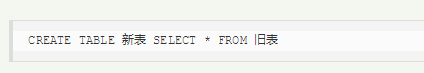
就是从一个表复制数据，然后把数据插入到另一个新表中。



在from语句后面一样可以使用where类似进行条件限制，指定我们需要选择的内容。



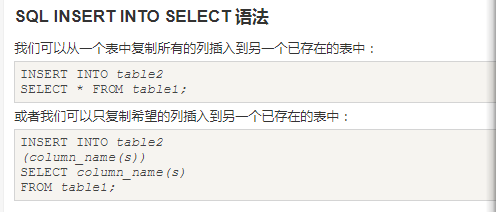
也可以像下面一样来拷贝表结构和数据：



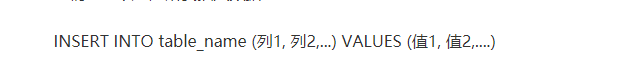
上面就是创建一个新表，然后将数据和结构拷贝到新表中。

1. insert into select

这个语句的作用就是从一张表中复制数据，然后把数据插入到另一张已经存在的表中，目标表中已经存在的行不会受到任何影响。



这个语句就是将从表一选择出来的数据，插入到表的某个指定的地方。



这就是才向一张空表写入数据的语法。

1. create database语句

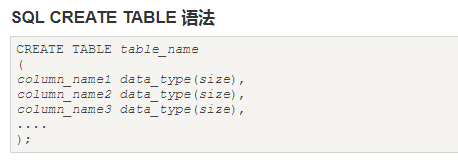
用于创建数据库。语法如下：

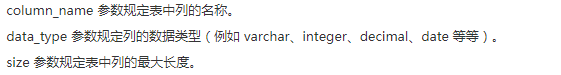


就是：创建数据库 数据库名称。其中的数据库表可以通过create table语句；来添加。

1. CREATE TABLE 语句

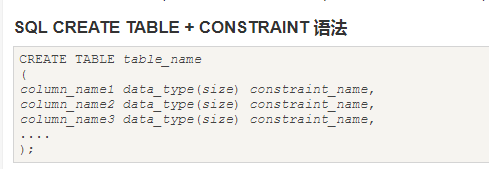
这个语句用于创建数据库中的表。表是由行和列组成的，每个表都必须有一个表名。

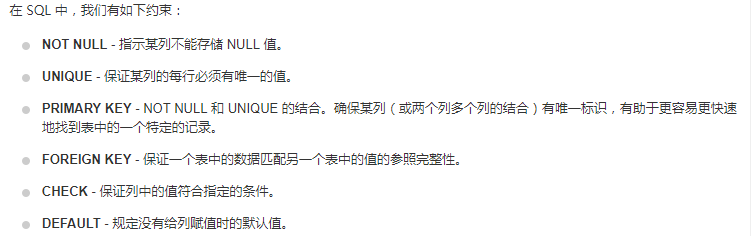




1. 约束constraints

用于规定表中的数据规则，如果存在违反约束规则的行为，那么该行为就会被终止。这个约束可以在创建表的时候规定，也可以在创建好之后通过alert table来规定。





约束条件是写在数据类型之后的。

1. NOT NULL

约束强制列不接受null值，也就是说请回要求字段始终包含值，不能为空，要是不向字段添加值，那么就无法向表插入新纪录或者是更新表的记录。

1. UNIQUE

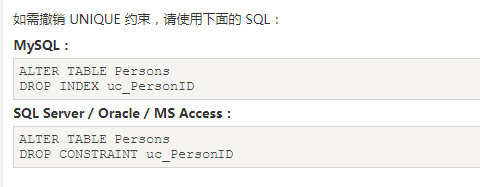
约束唯一标识数据库中的每条记录，UNIQUE和PRIMARY KEY均为列或者猎德集合提供了唯一性保证。每个表可以拥有多个UNIQUE约束，但是每个表只能有一个primary 可以约束。

Unique的写法有两种，要是无需命名，那就是直接写在数据类型后面就可以了，但是如果需要给约束命名，那么就写在后面专门把constraint列出来的地方。

alert table是的unique约束

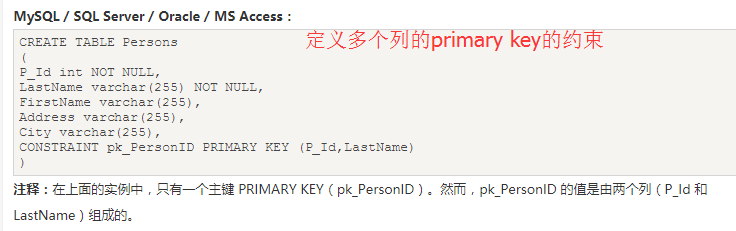


撤销unique约束



1. primary key约束



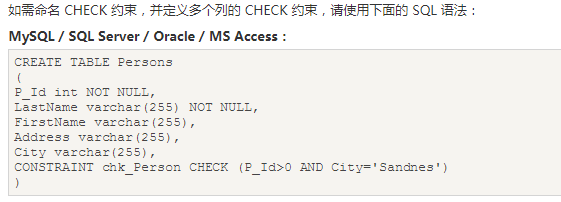
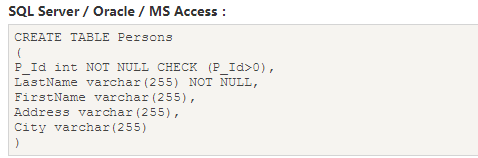


1. foreign key

用于预防破坏表结构之间连接的行为，同时也能防止数据插入外键列，因为她必须是指向那个表中的值之一。

1. Check

用于限制列中的值的范围。如果对单个列定义check约束，那么该列只允许特定的值；如果对一个表蒂尼check约束，那么此约束会基于行中其他列的值在特定的列中对值进行限制。

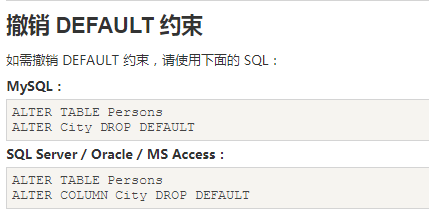


1. Default约束

用于向列中插入默认的值。如果没有其他的值，那么该默认值将会添加到所有的新纪录。写法就是在数据类型后面加上默认的default声明以及默认值；有些函数，列入GETDATE（）也可以通过default约束插入。

当表已经被创建的时候：如果需要在已经创建好的表中添加default约束，那么就使用下面的语法：





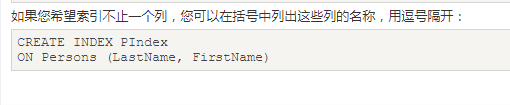
1. CREATE INDEX语句

用于在表中创建索引，在不读取整个表的情况下，索引使数据库引用程序可以更快的查找数据。用户是不能看见索引的，它只能用来加快数据的搜索和查询。

在更新一个含有索引的表的时候，花费的时间比不包含索引的表要长，因为在根性数据的时候也会同时更新索引，因此只对常用的数据添加索引。



同时，创建索引的时候不仅仅是值针对一个列，也可以将列包含在括号中，同时对多个列添加索引。

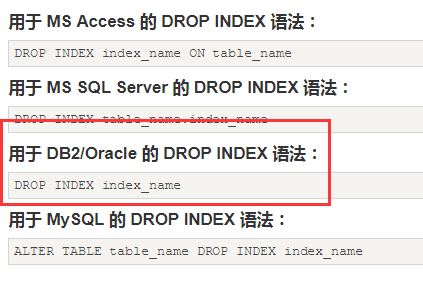


1. 撤销索引、撤销表以及撤销数据库

通过drop语句，可以轻松地删除索引、表以及数据库。

1. DROP INDEX语句

用于删除表中的索引。



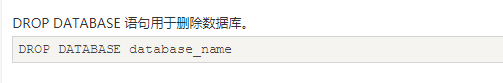
1. DROP TABLE 语句

用于删除表。



1. DROP DATABASE语句

用于删除数据库



1. TRUNCATE TABLE 语句

这个语句的作用就是只删除表中的数据，但是不删除表本身，相当于就是清空表，但是依然保存表的结构。

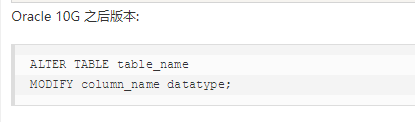
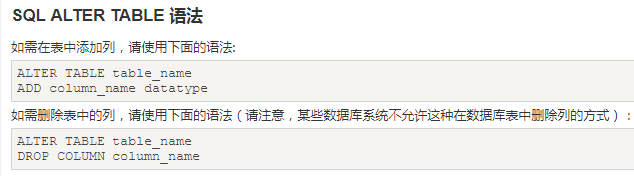
小结：

truncate table 在功能上与不带 WHERE 子句的 delete语句相同：二者均删除表中的全部行。但 truncate 比 delete速度快，且使用的系统和事务日志资源少。delete 语句每次删除一行，并在事务日志中为所删除的每行记录一项。所以可以对delete操作进行roll back。

参考文章：<http://www.cr173.com/html/40708_1.html；>

1. ALTER LABLE语句

用于在已有的表中添加、删除或修改列。



二十、AUTO INCREMENT字段

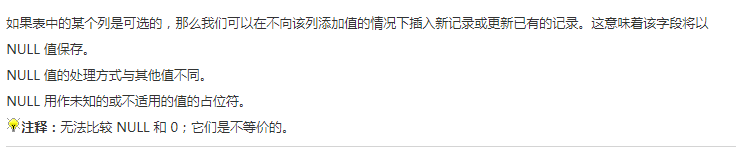
会在新纪录插入表中时生成一个唯一的数字。这个数字的起始值默认是1，每次插入新的数据的时候会自动递增1，但是若果需要以其他的数字开始的话，那么在适应这个字段的的时候就将其赋值为需要的数字。

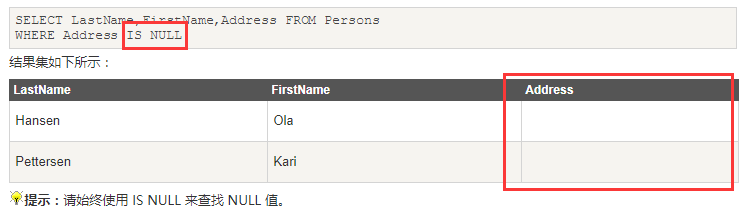


二十一、视图

视图是可视化的表，它是基于结果集的。视图包含行和列，就像一个真实的表，其中的字段就是来自于一个或多个数据库的真是的表中的字段，可以想我视图中添加SQL函数、WHERE以及join语句，也可以呈现数据，就像这些数据来自于某一个单一的表一样。

二十二、NULL值

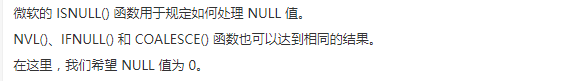


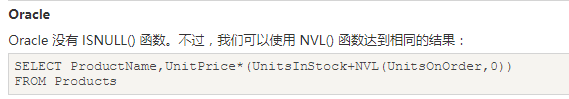


由于不能使用比较操作符来对null进行比较，因此要对null来进行查找就需要用咋混用的IS NULL 和IS NOT NULL来查找。

二十三、NULL函数







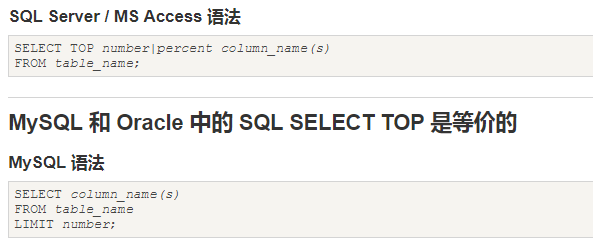
注意，在plsql developer中是不能自己去新建一个空间的，并且我们所看到的那些都不是一个文件夹而是一种分类，table，constraints等目录下面就是专门存放相关类别的空间。

第三部分 SQL高级教程

一、 SELECT TOP, LIMIT, ROWNUM 子句

1、SELECT TOP 子句

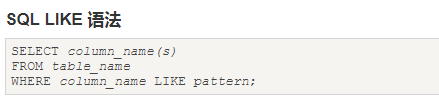
用于规定要返回的记录的数目，这个子句对于拥有数千条记录的大型表来说是比较有用的。但是，这个语句并不是所有的数据库都支持，在使用的时候需要先进行验证。





1. LIKE操作符

Like操作符用于砸where子句中搜索列中的指定模式。



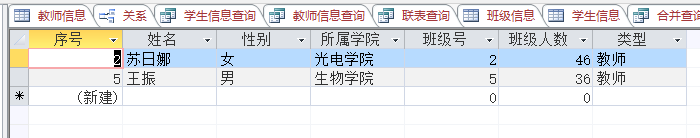
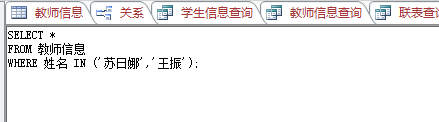
1. 通配符

可用于替代字符串中的任何其他字符，在SQL中是和SQL LIKE 操作符一起使用的，通配符用于搜索表中的数据。

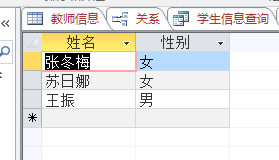
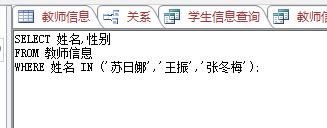


1. IN操作符

允许在where子句中规定多个值。



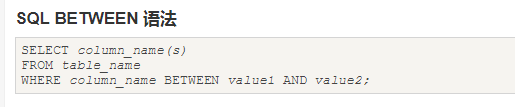
上面的写法是将含有in后面括号中信息的所有的行都选择出来，也可以像下面一样指明选择符合条件的那几列：

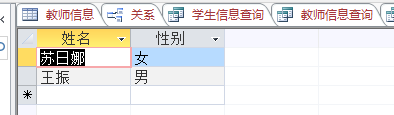
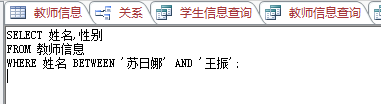


这个例子的意思就是：选择出in中姓名所在行的姓名和性别。

1. BETWEEN操作符

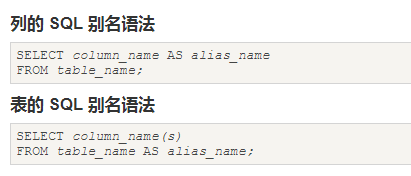
Between操作符用于选取介于两个值之间的数据范围的值，这些值可以是数字、文本、日期。





1. 别名

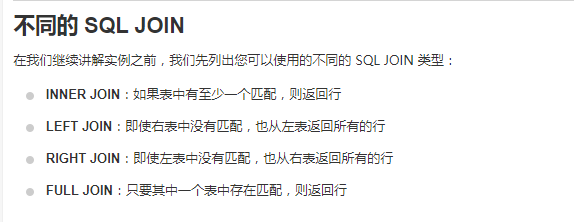
使用SQL可以为表名称和列名称指定别名，目的是使列或者行的可读性更好。



关于联表查询

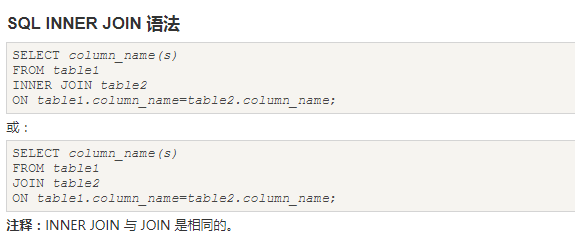
1. 连接join

Join子句用于把来自两个或多个表的行结合起来，基于这些表的共同字段。



1. inner join关键字

Inner join关键字在表中存在至少一个匹配的时候返回行。



Innerjoin和join是没有区别的，join默认的就是内连接。上面所说的匹配就是指的是数据类型和数据的值完全相同才算是匹配。

1. left join

以左右两张表为例，left join就是以左边的表为基础，然后在此基础上将右边的表中符合on后面条件的列返回添加到左边的表上，要是右边表中没有的内容就是null。其中的on就是讲表连接起来的固定用法。



1. full outer join关键字

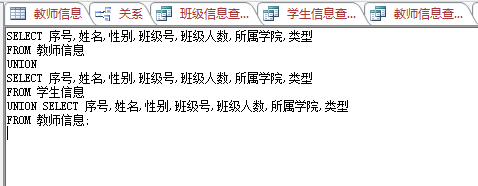
Full outer join关键字可以看做是取的表的并集。也就是说，在这个语句会返回表中所有的行，不管是是否匹配。在返回的结果中是显示出表的所有行，如果相对应的位置没有匹配，也就是hi没有值，那么就会被null填充。



十一、关于UNION组合查询

合并两个或多个select语句的结果集。

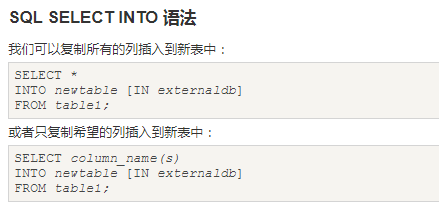
联合查询语句要求查询select语句的列数相同，并且每一列的数据类型也要相似才行。这是属于组合二查询的限制，更加确切的说应该是叫做合并查询结果在一个结果集中。这里的合并就是单纯的将查询结果合并到了一起，并不能称作联表查询。



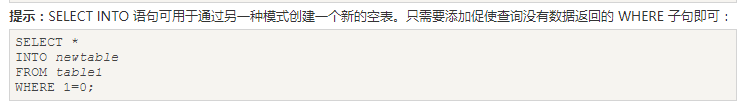
上面的这些SQL语句只是将这几个select的查询结果合并到了一起，并不是按照我们自己想要的标准从不同的表中获取到我们需要的数据。

1. select into 语句

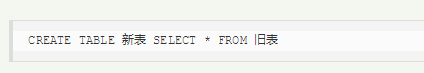
就是从一个表复制数据，然后把数据插入到另一个新表中。



在from语句后面一样可以使用where类似进行条件限制，指定我们需要选择的内容。



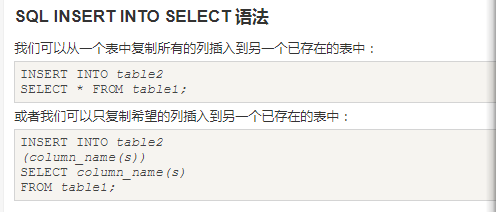
也可以像下面一样来拷贝表结构和数据：



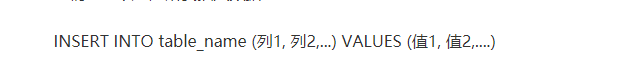
上面就是创建一个新表，然后将数据和结构拷贝到新表中。

1. insert into select

这个语句的作用就是从一张表中复制数据，然后把数据插入到另一张已经存在的表中，目标表中已经存在的行不会受到任何影响。



这个语句就是将从表一选择出来的数据，插入到表耳的某个指定的地方。



这就是才向一张空表写入数据的语法。

1. create database语句

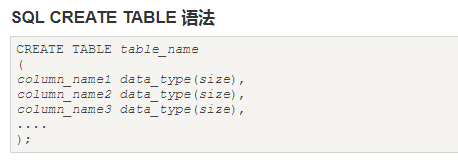
用于创建数据库。语法如下：

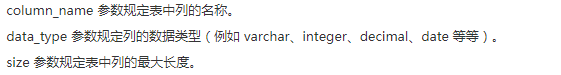


就是：创建数据库 数据库名称。其中的数据库表可以通过create table语句；来添加。

1. CREATE TABLE 语句

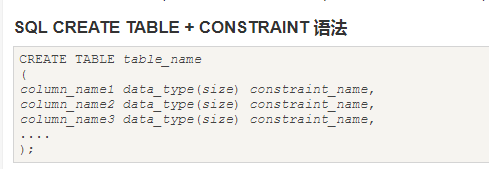
这个语句用于创建数据库中的表。表是由行和列组成的，每个表都必须有一个表名。

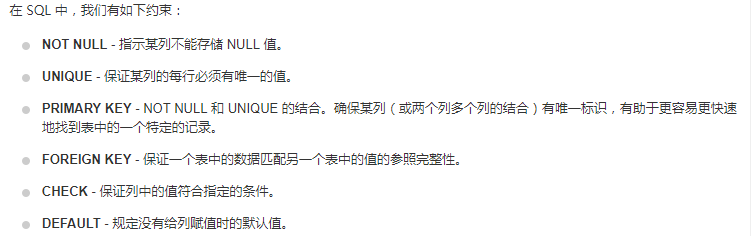




1. 约束constrains

用于规定表中的数据规则，如果存在违反约束规则的行为，那么该行为就会被终止。这个约束可以在创建表的时候规定，也可以在创建好之后通过alert table来规定。





约束条件是写在数据类型之后的。

1. NOT NULL

约束强制列不接受null值，也就是说请回要求字段始终包含值，不能为空，要是不向字段添加值，那么就无法向表插入新纪录或者是更新表的记录。

1. UNIQUE

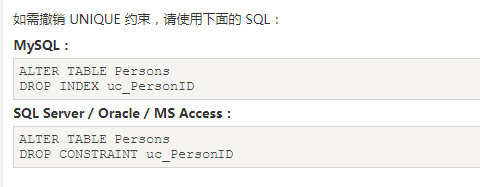
约束唯一标识数据库中的每条记录，UNIQUE和PRIMARY KEY均为列或者猎德集合提供了唯一性保证。每个表可以拥有多个UNIQUE约束，但是每个表只能有一个primary 可以约束。

Unique的写法有两种，要是无需命名，那就是直接写在数据类型后面就可以了，但是如果需要给约束命名，那么就写在后面专门把constraint列出来的地方。

alert table是的unique约束

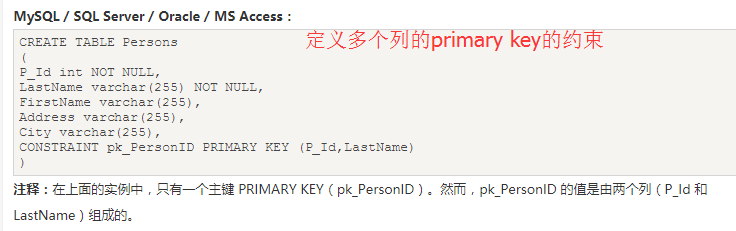


撤销unique约束



1. primary key约束



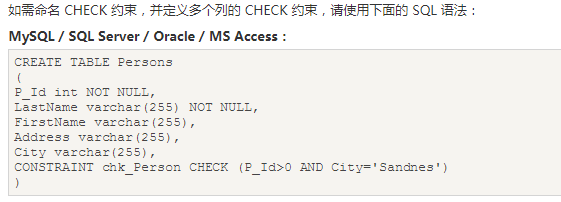
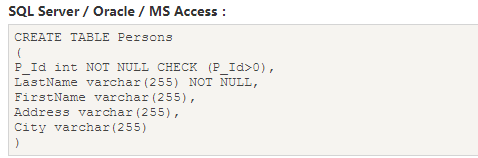


1. foreign key

用于预防破坏表结构之间连接的行为，同时也能防止数据插入外键列，因为她必须是指向那个表中的值之一。

1. Check

用于限制列中的值的范围。如果对单个列定义check约束，那么该列只允许特定的值；如果对一个表蒂尼check约束，那么此约束会基于行中其他列的值在特定的列中对值进行限制。

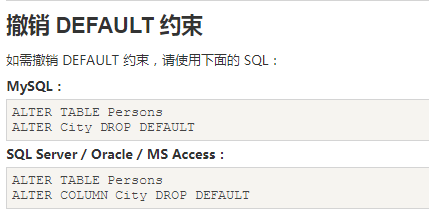


1. Default约束

用于向列中插入默认的值。如果没有其他的值，那么该默认值将会添加到所有的新纪录。写法就是在数据类型后面加上默认的default声明以及默认值；有些函数，列入GETDATE（）也可以通过default约束插入。

当表已经被创建的时候：如果需要在已经创建好的表中添加default约束，那么就使用下面的语法：





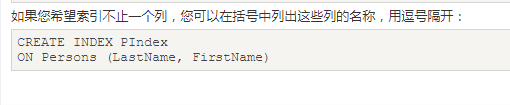
1. CREATE INDEX语句

用于在表中创建索引，在不读取整个表的情况下，索引使数据库引用程序可以更快的查找数据。用户是不能看见索引的，它只能用来加快数据的搜索和查询。

在更新一个含有索引的表的时候，花费的时间比不包含索引的表要长，因为在根性数据的时候也会同时更新索引，因此只对常用的数据添加索引。



同时，创建索引的时候不仅仅是值针对一个列，也可以将列包含在括号中，同时对多个列添加索引。

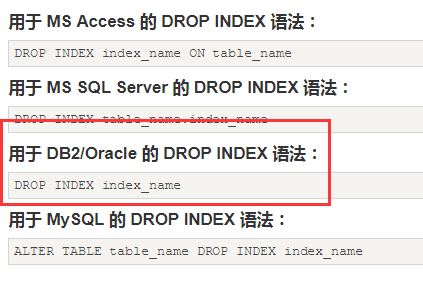


1. 撤销索引、撤销表以及撤销数据库

通过drop语句，可以轻松地删除索引、表以及数据库。

1. DROP INDEX语句

用于删除表中的索引。



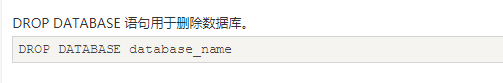
1. DROP TABLE 语句

用于删除表。



1. DROP DATABASE语句

用于删除数据库



1. TRUNCATE TABLE 语句

这个语句的作用就是只删除表中的数据，但是不删除表本身，相当于就是清空表，但是依然保存表的结构。

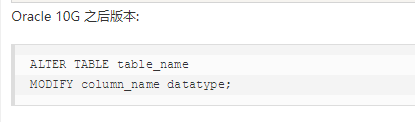
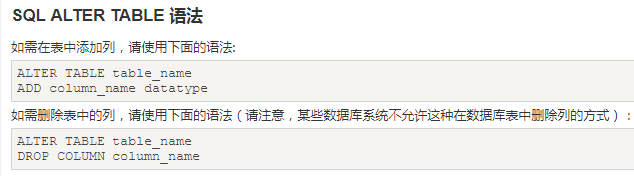
小结：

truncate table 在功能上与不带 WHERE 子句的 delete语句相同：二者均删除表中的全部行。但 truncate 比 delete速度快，且使用的系统和事务日志资源少。delete 语句每次删除一行，并在事务日志中为所删除的每行记录一项。所以可以对delete操作进行roll back。

参考文章：<http://www.cr173.com/html/40708_1.html；>

1. ALTER LABLE语句

用于在已有的表中添加、删除或修改列。



二十、AUTO INCREMENT字段

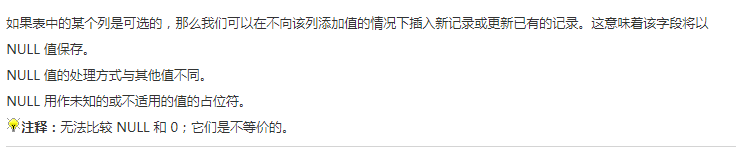
会在新纪录插入表中时生成一个唯一的数字。这个数字的起始值默认是1，每次插入新的数据的时候会自动递增1，但是若果需要以其他的数字开始的话，那么在适应这个字段的的时候就将其赋值为需要的数字。

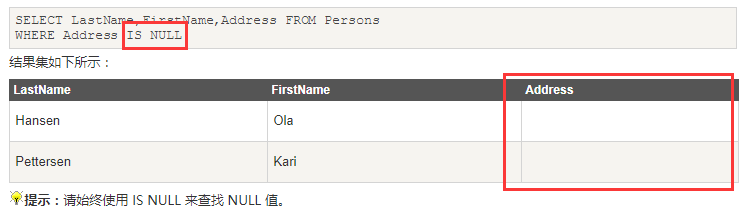


二十一、视图

视图是可视化的表，它是基于结果集的。视图包含行和列，就像一个真实的表，其中的字段就是来自于一个或多个数据库的真是的表中的字段，可以想我视图中添加SQL函数、WHERE以及join语句，也可以呈现数据，就像这些数据来自于某一个单一的表一样。

二十二、NULL值

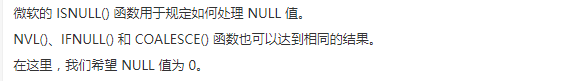


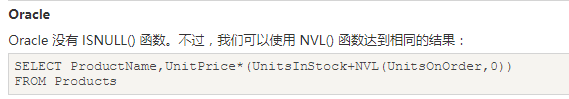


由于不能使用比较操作符来对null进行比较，因此要对null来进行查找就需要用咋混用的IS NULL 和IS NOT NULL来查找。

二十三、NULL函数



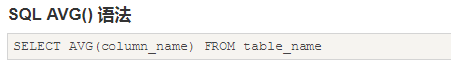




注意，在plsql developer中是不能自己去新建一个空间的，并且我们所看到的那些都不是一个文件夹而是一种分类，table，constraints等目录下面就是专门存放相关类别的空间。

第三部分 函数

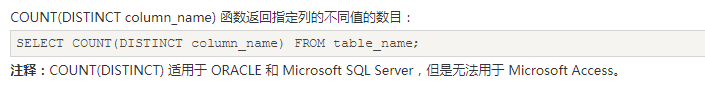
1. AVGA()——返回数值列的平均值



还可以在计算这个平均值的时候给这个计算结果一个别名，然后再结果集中国显示出来。

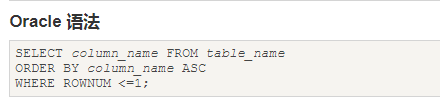
1. COUNT（）函数——返回匹配指定条件的函数

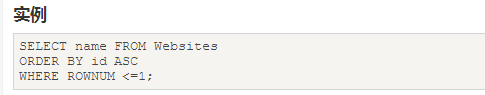
返回的是符合count函数参数这个条件的行数，要是是一个\*符号，呢么就会返回这个表中所有的记录的总数。其他条件以此类推。



1. first（）函数

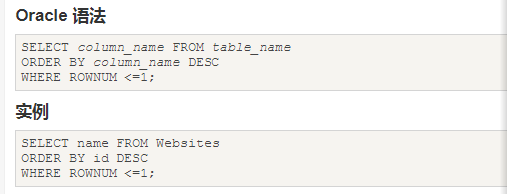
返回指定的列的第一个记录值。





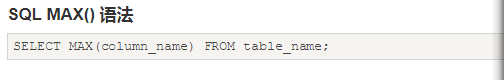
1. last（）函数

返回指定的列的最后一个记录。



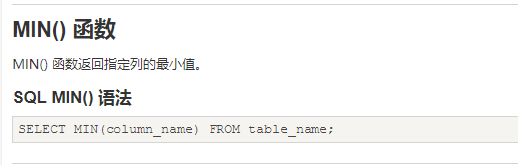
上面的fist和last两个函数只有myaccess支持！！！

1. MAX（）函数返回指定列的最大值



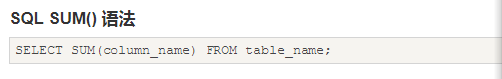
1. min()函数

返回指定列的最小值。

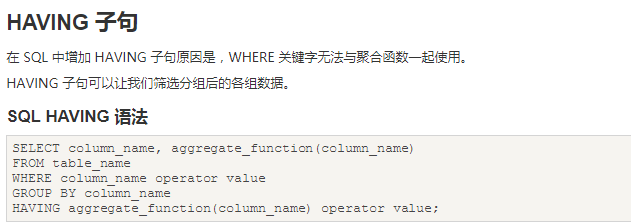


1. SUM（）函数

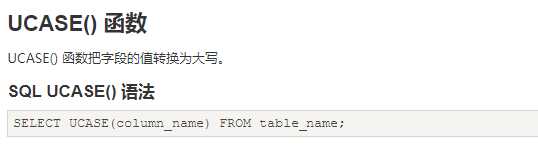
返回数值列的总数。



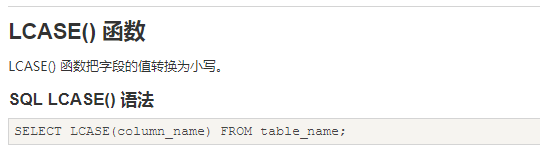
1. HAVING子句



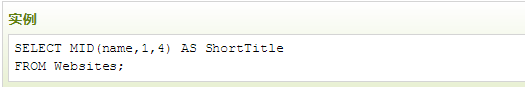
1. UCASE()函数



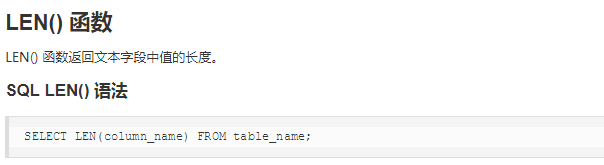
1. LCASE()函数



1. MID函数

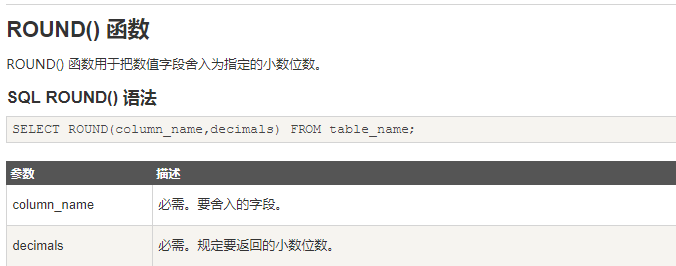


1. LEN()函数

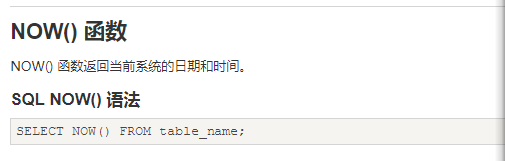


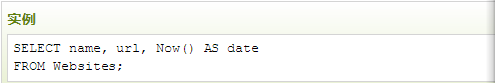
就是相当于计算某一列的那个文本的长度。

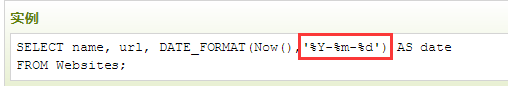
1. ROUND（）函数



1. ROUND（X）——返回四舍五入后的整数。
2. ROUND（X，D）——返回四舍五入后，带有D位小数的数字。
3. NOW()函数







上面红色的方框中就是指定了now（）函数获取的时间以什么样的格式来进行显示。

第四部分 触发器

1. 相关概念

触发器在数据库里以独立的对象存储，它与存储过程和函数不同的是，存储过程与函数需要用户显示调用才执行，而触发器是由一个事件来启动运行。即触发器是当某个事件发生时自动地隐式运行。并且，触发器不能接收参数。

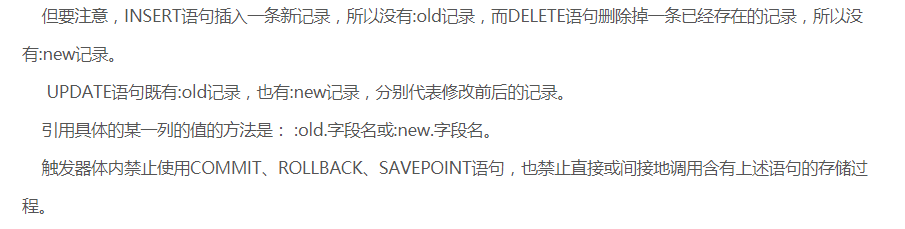
所以运行触发器就叫触发或点火（firing）。ORACLE事件指的是对数据库的表进行的INSERT、UPDATE及DELETE操作或对视图进行类似的操作。

ORACLE将触发器的功能扩展到了触发ORACLE，如数据库的启动与关闭等。所以触发器常用来完成由数据库的完整性约束难以完成的复杂业务规则的约束，或用来监视对数据库的各种操作，实现审计的功能。

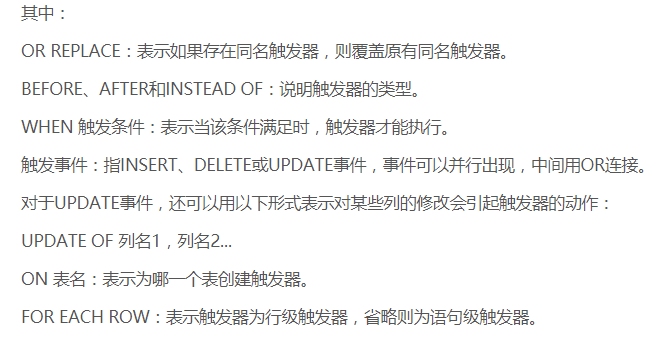
1. 触发器分类
2. DML触发器

ORACLE可以在DML语句进行触发，可以在DML操作前或操作后进行触发，并且可以对每个行或语句操作上进行触发。

（DDL——数据定义语句；DML——数据操作语句）



1. 创建触发器



1. 替代触发器

由于在ORACLE里，不能直接对由两个以上的表建立的视图进行操作。所以给出了替代触发器。它就是ORACLE 8专门为进行视图操作的一种处理方法。

1. 系统触发器

ORACLE 8i 提供了第三种类型的触发器叫系统触发器。它可以在ORACLE数据库系统的事件中进行触发，如ORACLE系统的启动与关闭等。

1. 触发器的组成

（1）触发事件：引起触发器被触发的事件。 例如：DML语句(INSERT, UPDATE, DELETE语句对表或视图执行数据处理操作)、DDL语句（如CREATE、ALTER、DROP语句在数据库中创建、修改、删除模式对象）、数据库系统事件（如系统启动或退出、异常错误）、用户事件（如登录或退出数据库）。

（2）触发时间：即该TRIGGER 是在触发事件发生之前（BEFORE）还是之后(AFTER)触发，也就是触发事件和该TRIGGER 的操作顺序。

（3）触发操作：即该TRIGGER 被触发之后的目的和意图，正是触发器本身要做的事情。 例如：PL/SQL 块。

（4）触发对象：包括表、视图、模式、数据库。只有在这些对象上发生了符合触发条件的触发事件，才会执行触发操作。

（5）触发条件：由WHEN子句指定一个逻辑表达式。只有当该表达式的值为TRUE时，遇到触发事件才会自动执行触发器，使其执行触发操作。

（6） 触发频率：说明触发器内定义的动作被执行的次数。即语句级(STATEMENT)触发器和行级(ROW)触发器。

语句级(STATEMENT)触发器：是指当某触发事件发生时，该触发器只执行一次；

行级(ROW)触发器：是指当某触发事件发生时，对受到该操作影响的每一行数据，触发器都单独执行一次。

**存储过程**

1. 定义：存储过程（Stored Procedure ）是一组为了完成特定功能的SQL 语句 集，经编译后存储在数据库中。用户通过指定存储过程的名字并给出参数 （如果该存储过程带有参数）来执行它。

存储过程是数据库中的一个重要对象，任何一个设计良好的数据库应用程序都应该用到存储过程。 存储过程是由流控制和SQL 语句书写的过程，这个过程经编译和优化后 存储在数据库服务器中，应用程序使用时只要调用即可。在ORACLE 中， 若干个有联系的过程可以组合在一起构成程序包。

优点：

1. 减少编译次数（只在创建的时候编译一次，调用不编译），使用存储过程可提高数据库执行速度。
2. 当对数据库进行复杂操作时(如对多个表进行Update、Insert、Query、Delete时），可将此复杂操作用存储过程封装起来与数据库提供的事务处理结合一起使用。
3. 存储过程可以重复使用,可减少数据库开发人员的工作量；

问题及解决方法记录

1. 在创建表的时候，如果规定了主键，那么在为表格插入内容的时候，如果主键出现相同的值，那么一定会提示“违反唯一性原则”，这时候如果确实需要，可以在query data去将主键去掉。
2. 在sql语句中，即使是英文输入法也不能使用双引号，否则会报错；在写代码的时候，数字不用写引号，其余的需要引号，注意是单引号！