## In/not In

**一、1.mysql in语句**

select \* from tb\_name where id in (10,12,15,16);

**2.mysql not in 语句**

select \* from tb\_name where **not id in** (21,22,23);

**3.mysql in not in 一起使用**

select \* from tb\_name where id in (10,12,15,16) and not id in (21,22,23);

二、select \* from abc\_number\_prop where number\_id in (select number\_id from abc\_number\_phone where phone = '82306839');

## 2. Exists/not exists



我们要查出 article 表中的数据，但要求 uid 必须在 user 表中存在。SQL 语句如下：

**SELECT \* FROM article WHERE EXISTS (SELECT \* FROM user WHERE article.uid = user.uid)**

## 3. UNION 和 UNION ALL 操作符

**解释**：UNION 操作符用于合并两个或多个 SELECT 语句的结果集。

**请注意**，UNION 内部的 SELECT 语句必须拥**有相同数量的列。**列也必须拥有**相似的数据类型**。同时，每条 SELECT 语句中的**列的顺序必须相同**。

SELECT column\_name(s) FROM table\_name1

UNION

SELECT column\_name(s) FROM table\_name2

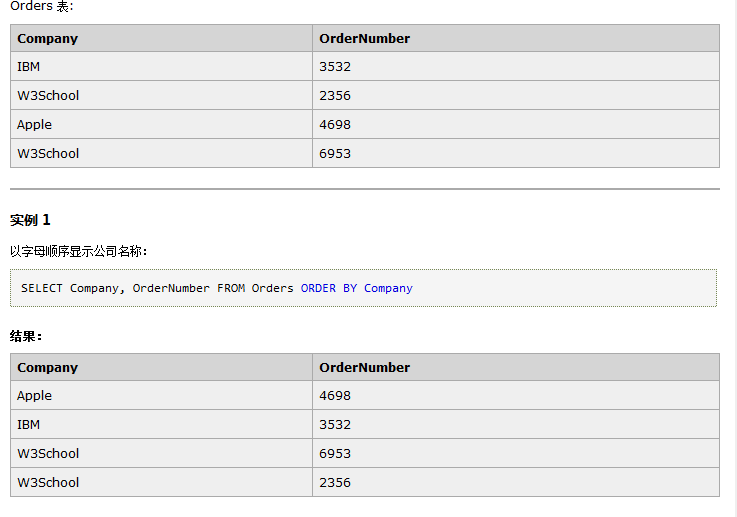
注释：默认地，UNION 操作符选取不同的值。如果允许重复的值，请使用 UNION ALL。

## 4. Order by

**解释：**ORDER BY 语句**默认按照升序**对记录进行排序。如果您希望按照降序对记录进行排序，可以使用 DESC 关键字。

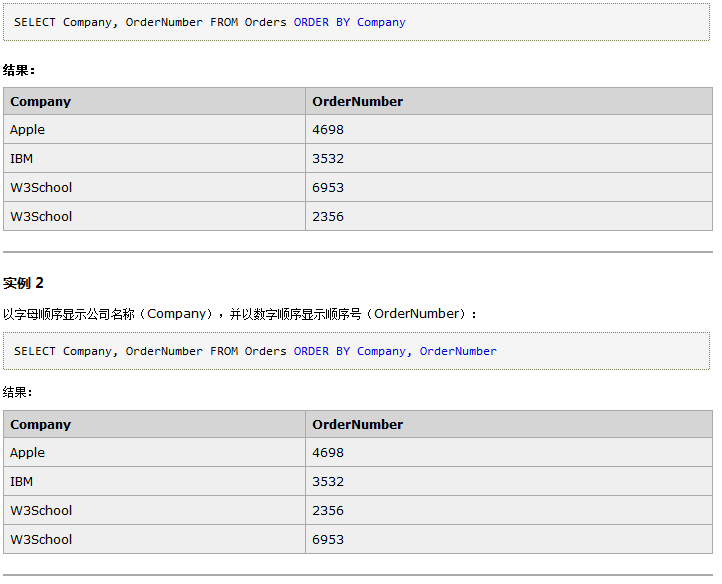
### 一、以字母顺序

SELECT Company, OrderNumber FROM Orders ORDER BY Company



### 二、以字母顺序，并且按照数字顺序进行排列

**SELECT Company, OrderNumber FROM Orders ORDER BY Company, OrderNumber**



### 三 DESC 逆序 排列 及多条件顺序

以逆字母顺序显示公司名称，并以数字顺序显示顺序号：

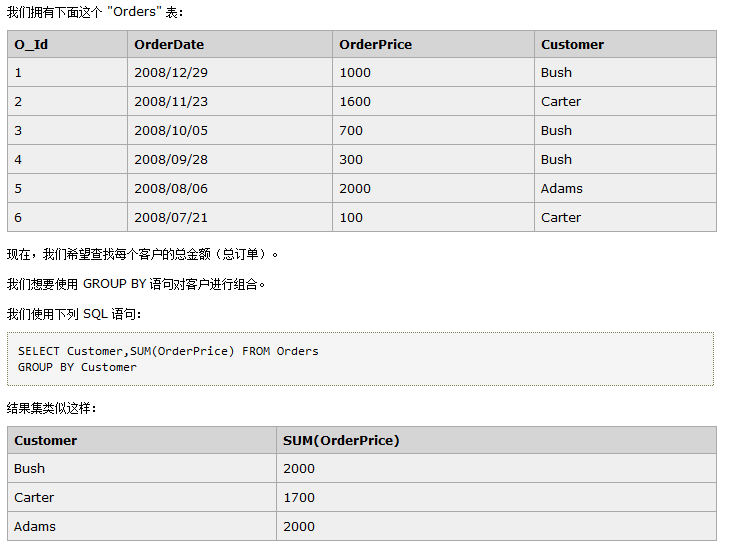
SELECT Company, OrderNumber FROM Orders ORDER BY Company DESC, OrderNumber ASC

## **5.GROUP BY**

解释：语句用于结合合计函数，根据一个或多个列对结果集进行分组。

**通俗点讲就是 根据Group by 后面的段位 进行分组，将相同的放到一起**

### 一、SELECT Customer,SUM(OrderPrice) FROM Orders GROUP BY Customer



### 二、如果没有group by 乱套了吧



## 6. exp、imp

Oracle数据导入导出imp/exp就相当于逻辑(数据的)备份与恢复，支持客户端与服务器操作。  
exp命令可以把数据从远程数据库服务器导出到本地的dmp文件，  
imp命令可以把dmp文件从本地导入到远处的数据库服务器中。  
执行环境：在客户端与服务端的shell状态下DOS（命令行）中执行

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*数据导出 exp \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
1、全库方式,导出整个数据库中所有的对象，但并不包括sys用户中的对象，即数据字典无法导出。  
exp user/pwd@db\_name full=y file=D:\database.dmp log=D:\database.log

2、用户方式：导出某一用户下所有的对象，授权了权限的用户可以导出其他用户所拥有的对象。作为全库导出的补充应用  
exp user/pwd@db\_name owner=(system,sys) file=D:\owner.dmp log=D:\owner.log

3、表方式：只导出某一用户下指定的表，而不是所有的表。  
exp user/pwd@db\_name tables=(tb1,tb2) file= D:\table.dmp log=e:\table.log

4、补充：将数据库中的表table1中的字段filed1以"00"打头的数据导出  
exp user/pwd@db\_name tables=(table1) query=" where filed1 like '00%'" file=D:\query.dmp log=d:\query.log  
    
\*\*\*上面是常用的导出 \*\*更多参数exp help=y 查看

===================================================================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*数据的导入 imp \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
1、全部导入  
imp user/pwd@db\_name full=y file=D:\database.dmp log=d:\impdatabase.log ignore=y

2、选择表导入 将D:\table.dmp中的表table1 导入  
imp user/pwd@db\_name file=D:\table.dmp log=d:\imptable.log tables=(table1)   
如果源表已经存在，导入时报错。在后面加上 ignore=y 就可以了。  
不少情况要先是将表彻底删除，然后导入。或创建和原表一样结构的临时表然后导入到临时表中。

备注：语句执行中可能碰上的问题：  
\*\*\***EXP-00091 正在导出有问题的统计信息**原因：字符集问题  
解决：exp命令加statistics=none选项

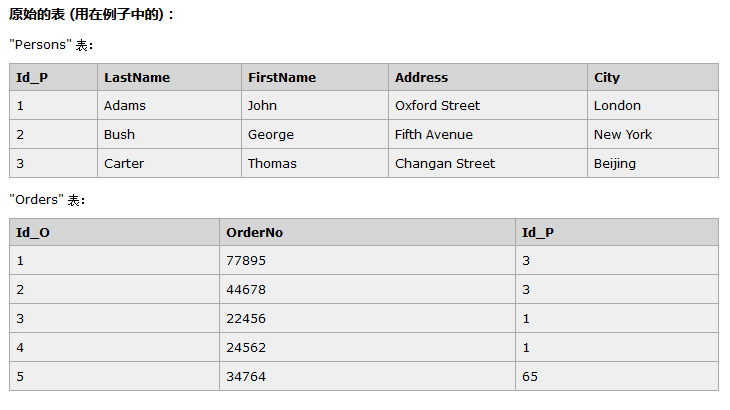
**expdp的impdp (数据泵)**是Oracle10G新引入的工具.它不但包括了imp/exp的功能，还进行了扩充与加强。其速度也快。但只能在数据库服务端运行。使用示例如下：<http://www.oracle-base.com/articles/10g/OracleDataPump10g.php>

## 8. INNER JOIN 关键字

解释：在表中存在至少一个匹配时，INNER JOIN 关键字返回行。

通俗的解释为：如果是一个表加入另一个表，那么就是先看第一个表在第二个表中有没有，没有就不列出来了。

注释：INNER JOIN 与 JOIN 是相同的。



SELECT Persons.LastName, Persons.FirstName, Orders.OrderNo FROM Persons INNER JOIN Orders ON Persons.Id\_P=Orders.Id\_P ORDER BY Persons.LastName



8.2 内连接 左链接，右连接，全连接

### JOIN: 如果表中有至少一个匹配，则返回行 （同8中图片例子）

### LEFT JOIN: 即使右表中没有匹配，也从左表返回所有的行

* 

### RIGHT JOIN: 即使左表中没有匹配，也从右表返回所有的行

* 

### FULL JOIN: 只要其中一个表中存在匹配，就返回行



## 9.下翻页Oracle Mysql

Oracle "select top ? \* form user where useId not in(selest top ? form user)")

Mysql SELECT \* FROM user LIMIT pageSize\*(pageNow-1),pageSize

limit 表示从即开始取几条

## 10. TOP

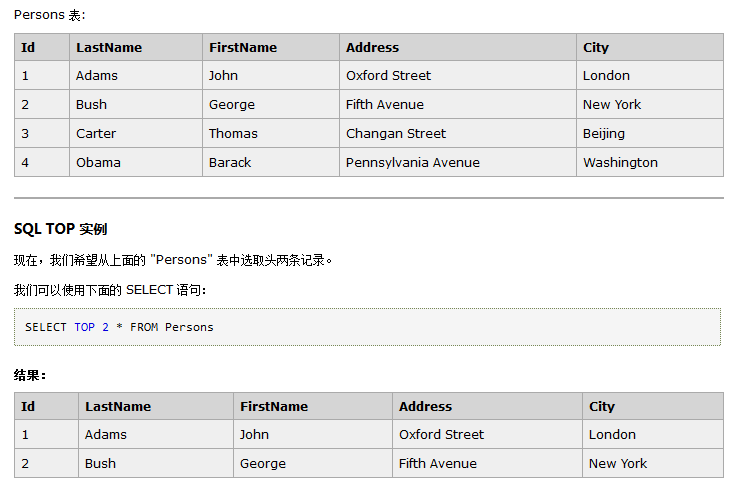
## 解释;TOP 子句用于规定要返回的记录的数目。

注释：并非所有的数据库系统都支持 TOP 子句。

### 10.1.取出最后一条插入的语句

SELECT TOP 1 \* FROM TB ORDER BY SDT DESC

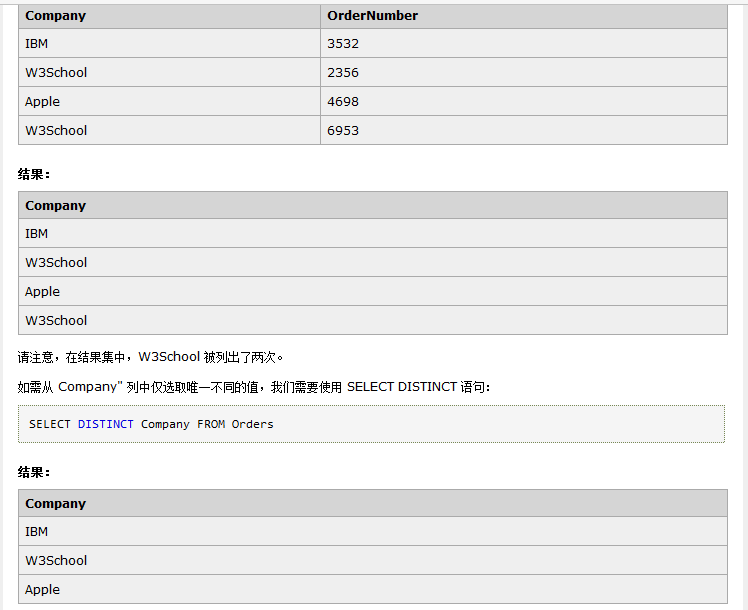
### 10.2 SELECT TOP 2 \* FROM Persons



## 11. DISTINCT

解释：关键词 DISTINCT 用于返回唯一不同的值。

SELECT DISTINCT Company FROM Orders



## 12创建序列

ORACLE序列的语法格式为：

CREATE SEQUENCE 序列名  
[INCREMENT BY n]  
[START WITH n]  
[{MAXVALUE/ MINVALUE n|NOMAXVALUE}]  
[{CYCLE|NOCYCLE}]  
[{CACHE n|NOCACHE}];

解释 **1)INCREMENT BY**用于定义序列的步长，如果省略，则默认为1，如果出现负值，则代表Oracle序列的值是按照此步长递减的。

**2)START WITH 定**义序列的初始值(即产生的第一个值)，默认为1。

**3)MAXVALUE** 定义序列生成器能产生的最大值。选项NOMAXVALUE是默认选项，代表没有最大值定义，这时对于递增Oracle序列，系统能够产生的最大值是10的27次方;对于递减序列，最大值是-1。

**4)MINVALUE**定义序列生成器能产生的最小值。选项NOMAXVALUE是默认选项，代表没有最小值定义，这时对于递减序列，系统能够产生的最小值是?10的26次方;对于递增序列，最小值是1。

**5)CYCLE和NOCYCLE** 表示当序列生成器的值达到限制值后是否循环。CYCLE代表循环，NOCYCLE代表不循环。如果循环，则当递增序列达到最大值时，循环到最小值;最小值 为1。对于递减序列达到最小值时，循环到最大值。如果不循环，达到限制值后，继续产生新值就会发生错误。

**6)CACHE(缓冲)**定义存放序列的内存块的大小，默认为20。NOCACHE表示不对序列进行内存缓冲。对序列进行内存缓冲，可以改善序列的性能。

### 12.1例子

1 --创建测试表--

2 CREATE TABLE tbl\_test(

3 test\_id NUMBER PRIMARY KEY,

4 test\_name VARCHAR2(20)

5 );

6

7 --为tbl\_test创建序列--

8 CREATE SEQUENCE seq\_test

9 　　　　INCREMENT BY 1 -- 每次加几个

10 　　　　START WITH 1 -- 从1开始计数

11 ;

12

13 --插入测试数据--

14 INSERT INTO tbl\_test VALUES(seq\_test.nextval,'测试');

15 COMMIT;

16

17 --查询表中的数据

18 SELECT \* FROM tbl\_test;

 以上程序没有结果：解决如下

1.更改数据库的“延迟段创建”特性为false（需要有相应的权限）

SQL语句如下：

ALTER SYSTEM SET deferred\_segment\_creation=FALSE;

或者

**2.在创建表时让seqment立即执行 （创建表时候让序列同时立即创建）**

SQL语句如下：

CREATE TABLE tbl\_test(

test\_id NUMBER PRIMARY KEY,

test\_name VARCHAR2(**20**)

)

SEGMENT CREATION IMMEDIATE;

## 13 View视图

## 14索引

解释: 索引是用于加速数据存取的数据对象，合理的使用索引可以大幅降低I/O次数，从而提高数据访问性能, **在不读取整个表的情况下，索引使数据库应用程序可以更快地查找数据。**您可以在表中创建索引，以便更加快速高效地查询数据。用户无法看到索引，它们只能被用来加速搜索/查询。注释：更新一个包含索引的表需要比更新一个没有索引的表更多的时间，这是由于索引本身也需要更新。因此，理想的做法是仅仅在常常被搜索的列（以及表）上面创建索引。

### 14.2 CREATE INDEX 实例

1.本例会创建一个简单的索引，名为 "PersonIndex"，在 Person 表的 LastName 列：

CREATE INDEX PersonIndex ON Person (LastName)

2.如果您希望以降序索引某个列中的值，您可以在列名称之后添加保留字 DESC：

CREATE INDEX PersonIndex ON Person (LastName DESC)

3.假如您希望索引不止一个列，您可以在括号中列出这些列的名称，用逗号隔开：

CREATE INDEX PersonIndex ON Person (LastName, FirstName)