

sql必知必会：

大小写不区分，但关键字常使用大写，而列名和表名使用小写

1.检索不同的值：SELECT DISTINCT prob_name(列名) from Products (表名)；(但当存在多列时，DISTINCT不能部分使用，关键字作用于所有的列，不仅仅是跟在其后的那一列，仅仅去除这些列组合值重复的行)

2.限制结果：

当使用SqlServer、Access时使用TOP关键字，SELECT TOP 5 prob_name(列名) from Products (表名)；

当使用MySQL时使用LIMIT关键字，SELECT prob_name(列名) from Products (表名) LIMIT 5; (表示返回不超过5行数据)，但当需要指定从哪儿开始时，SELECT prob_name(列名) from Products (表名) LIMIT 5 OFFSET 5; (表示从第5行起 (不包含) 的5行数据)，但当有 limit 2,1时表示表示跳过2行数据，读取第3行数据

当使用Oracle时需要使用ROWNUM，SELECT prob_name(列名) from Products (表名) WHERE ROWNUM<=5;

3.排序数据：

ORDER BY：通常放在最后，默认是升序

可以通过非选择列进行排序，例如：SELECT prod_name FROM Products ORDER BY prob_price;

可以按多个列排序，例如：SELECT prod_id, prod_name, prob_price FROM Products ORDER BY prob_price, prob_name;

可以按列位置排序：例如：SELECT prod_id, prod_name, prob_price FROM Products ORDER BY 2, 3 (这里2,3指的是选择列的相对位置);

可以指定排序方向：利用关键字DESC，SELECT prod_id, prod_name, prob_price FROM Products ORDER BY prob_price DESC;(若想在多个列上降序排序，则需要每一列都指定DESC关键字)

4.过滤数据

WHERE子句操作符：一些特殊的有：<>不等于，!<不小于，!>不大于，BETWEEN 在指定的两个值之间，IS NULL 为NULL值 (并非所有DBMS都支持这些操作符)

空值检查：SELECT prod_name FROM Products WHERE prob_price IS NULL; 通过过滤选择不包含某些指定值的所有行时，是不会返回含NULL的值的，比如WHERE prod_name != 'aaa', prod_name为null的行也是不会被返回的。因为NULL代表未知，数据库也无法知道它是不是匹配。过滤数据时，一定要验证被过滤列中含NULL的行确实出现在返回的数据中。

求值顺序（组合使用AND和OR操作符时）例如：SELECT prod_id, prob_price FROM Products WHERE venid='1' OR venid='2' AND prob_price>=10;，由于AND 优先级较高导致结果中出现prob_price<10的数据，在组合使用AND和OR时都应使用圆括号，改：SELECT prod_id, prob_price FROM Products WHERE (venid='1' OR venid='2') AND prob_price>=10;

IN操作符：与OR功能相同，好处：更直观、与AND或OR操作符组合使用时求值顺序更容易管理，比一组OR操作符执行得更快，能够包含其他的SELECT语句，更动态地建立WHERE 子句

NOT操作符：可以否定跟在其后的条件，SELECT prod_id, prob_price FROM Products WHERE NOT venid='1';

LIKE操作符：%通配符：任何字符出现任意次数，SELECT prod_id, prob_price FROM Products WHERE prob_name LIKE '%Fish%';(注意当后面跟空格时（有些DBMS用空格填补字段内容），使用'F%y'无法匹配，改正办法可以在后面加上%，例如'F%y%'，还有就是'%'并不匹配NULL

_通配符：匹配单个字符

[]通配符（只有Access和SQL Server支持）：指定一个字符集，必须匹配指定位置的一个字符

5.创建计算字段

拼接字段：可以用+来拼接两个列，例如：SELECT vend_name+'('+vend_country+)' FROM Vendors ORDER BY vend_name; MySql中用Concat来拼接，并且可以使用RTRIM去除不需要的空格，并且可以在后面使用AS关键字为新创建的列起名字，

日期和时间处理函数：Date_PART('yyy', order_date):取年份（SQL Server版本），Oracle中to_date()用来将2个字符串转换为日期，MySQL中用YEAR()从

日期中提取年份

6.汇总数据

聚集函数：AVG(): 返回某列的平均值（参数只能为单列）：SELECT

AVG(prob_price) AS avg_price FROM Products ;

COUNT():COUNT(*)对表中行的数目进行计数，不管是否包含NULL值，COUNT(prob_price)对特定列中具有值的行进行计数

MAX(): 返回指定列中最大值，用于文本数据时返回该列排序后的最后一行

MIN(): 返回指定列中最小值，同上

SUM():返回指定列值的和

7.分组数据

GROUP BY:

例如SELECT ven_id, COUNT(*) AS num_probs FROM Products GROUP BY vend_id;

GROUP BY 子句中可以包含任意数目的列，如果嵌套了分组，将在最后指定的分组上进行汇总

GROUP BY子句中的每一列必须是检索列或有效的表达式（不能是聚集函数），并且列不允许带有长度可变的数据类型

GROUP BY必须在WHERE之后，ORDER BY之前

HAVING (过滤分组)：

WHERE 是数据分组前进行过滤，HAVING 是数据分组后进行过滤

例如：SELECT ven_id, COUNT(*) AS orders FROM Products GROUP BY vend_id HAVING COUNT(*)>=2;

SELECT 子句顺序：WHERE 、GROUP BY 、 HAVING、 ORDER BY

8.联结表

创建联结：SELECT prod_id, prob_price , vend_name FROM Products ,Vendors WHERE Vendors.vend_id=Products.vend_id;

如果是这样返回的就是两个表的笛卡尔积：SELECT prod_id, prob_price , vend_name FROM Products ,Vendors;

内联结: SELECT prod_id, prob_price , vend_name FROM Vendors
INNER JOIN Products ON Vendors.vend_id=Products.vend_id;

自联结: 查询出与Jim Jones同一公司的所有顾客: SELECT c1.cust_id,
c1.cust_name,c1.cust_contact FROM Customers AS c1,Customers AS c2
WHERE c1.cust_name=c2.cust_name AND c2.cust_contact ='Jim Jones';

外联结:

- 左连接或左外连接: select 表A left (outer) join 表B on条件: 返回A(左表) 中所有数据及B中满足条件数据, 如果左表的某行在右表中没有匹配行, 则在相关联的结果集行中右表的所有选择列表列均为空值。
- 右连接或右外连接: select 表A right (outer) join 表B on条件: 返回B(右表) 中所有数据及A中满足条件数据, 如果右表的某行在左表中没有匹配行, 则将为左表返回空值。

SQL连接时, 过滤条件放在on中与放在where中的区别:

对于inner join 二者是没有区别的, 对于左连接来说:

- on 条件1 and 条件2: on条件是在生成临时表时使用的条件, 它不管on中的条件是否为真, 都会返回左边表中的记录。(on中条件只是针对右表来说)
- on 条件1 where 条件 2: where条件是在临时表生成好后, 再对临时表进行过滤的条件。这时已经没有left join的含义(必须返回左边表的记录)了, 条件不为真的就全部过滤掉, on后的条件用来生成左右表关联的临时表, where后的条件对临时表中的记录进行过滤。

9.组合查询

UNION:必须由两条或两条以上的SELECT语句组成, 语句之间用UNION关键字分割, 查询结果集中自动去除重复的行

SELECT cust_id, cust_name FROM Customers WHERE cust_state
IN('1','2','3')

UNION

```
SELECT cust_id, cust_name FROM Customers WHERE  
cust_name='Fun4ALL'  
ORDER BY cust_name;
```

ORDER BY是对组合过后的结果进行排序

10.插入数据

```
INSERT INTO Customers(cust_id, cust_name) values('1011','TONY');
```

INSERT SELECT:INSERT INTO Customers(cust_id, cust_name) SELECT
cust_id, cust_name FROM CustNew;(列名不需要匹配,使用的是列的位置)
表的复制:

```
SELECT * INTO CustCopy FROM Customers;
```

MySQL 中则使用: Create TABLE CustCopy AS SELECT * FROM
Customers;

11.更新数据:

```
UPDATE Customers SET cust_contact='SAM' WHERE cust_id='001';
```

12.删除数据:

```
DELETE FROM Customers WHERE cust_id='001'
```

13.更新表:

```
ALTER TABLE Customers ADD vend_phone CHAR(20) ;
```

14.创建视图

许多DBMS禁止在视图查询中使用ORDER BY子句

视图不能索引

```
CREATE VIEW ProductCustomers AS SELECT prod_id, prob_price ,  
vend_name FROM Products ,Orders  
WHERE Products.vend_id=Orders.vend_id;
```

15.创建索引

```
CREATE INDEX prob_name_id ON Products(prob_name);
```

16.MySql中Datetime与Timestamp的区别

- DATETIME的日期范围是1001——9999年, TIMESTAMP的时间范围是1970——2038年。

- DATETIME存储时间与时区无关，TIMESTAMP存储时间与时区有关，显示的值也依赖于时区。在mysql服务器，操作系统以及客户端连接都有时区的设置。
- DATETIME使用8字节的存储空间，TIMESTAMP的存储空间为4字节。因此，TIMESTAMP比DATETIME的空间利用率更高。
- DATETIME的默认值为null；TIMESTAMP的字段默认不为空（not null）默认值为当前时间（CURRENT_TIMESTAMP），如果不做特殊处理，并且update语句中没有指定该列的更新值，则默认更新为当前时间。