## 十三、Refining Selections & Aggregate Functions（数据筛选与聚合函数）

### 1. DISTINCT

DISTINCT is used to return only unique (different) values in a query result.  
DISTINCT 关键字用于筛选唯一值（去重）。  
  
SELECT author\_lname FROM books;  
SELECT DISTINCT author\_lname FROM books;  
SELECT DISTINCT author\_fname, author\_lname FROM books;  
SELECT DISTINCT CONCAT(author\_fname, ' ', author\_lname) FROM books;  
  
可以单列去重，也可以多列组合去重（例如 CONCAT(fname, lname) 可以作为唯一组合）。

### 2. ORDER BY（排序结果）

ORDER BY sorts query results in ascending or descending order.  
ORDER BY 用于对结果排序，默认升序。  
  
SELECT \* FROM books ORDER BY author\_lname;  
SELECT \* FROM books ORDER BY author\_lname DESC;  
SELECT \* FROM books ORDER BY released\_year;  
  
ORDER BY column position 示例：  
SELECT book\_id, author\_fname, author\_lname, pages FROM books ORDER BY 2 DESC;  
SELECT book\_id, author\_fname, author\_lname, pages FROM books ORDER BY author\_lname, author\_fname;

### 3. LIMIT（限制结果数量）

LIMIT restricts how many rows are returned.  
LIMIT 用于限制返回行数。  
语法：LIMIT starting\_row, number\_of\_rows  
  
SELECT title, released\_year FROM books ORDER BY released\_year DESC LIMIT 0, 5;  
SELECT title, released\_year FROM books ORDER BY released\_year DESC LIMIT 1, 3;  
  
第一个数字表示起始行（从0开始），第二个表示要显示的数量。

### 4. LIKE（模糊查询）

LIKE allows pattern matching with wildcards % and \_.  
LIKE 用于模糊匹配，% 代表任意字符序列，\_ 代表单个字符。  
  
SELECT title, author\_fname, author\_lname, pages FROM books WHERE author\_fname LIKE '%da%';  
SELECT \* FROM books WHERE author\_fname LIKE '\_a\_';  
SELECT \* FROM books WHERE author\_fname LIKE '%n';  
  
If you need to search literal % or \_, use escape character \.  
如果要查找包含 % 或 \_ 本身的内容，用反斜杠 \ 转义。  
SELECT \* FROM books WHERE title LIKE '%\%%';  
SELECT \* FROM books WHERE title LIKE '%\\_%';

### 5. Aggregate Functions（聚合函数）

Aggregate functions perform calculations on multiple rows (e.g., sum, average, count).  
聚合函数用于对多行数据进行统计汇总。  
  
COUNT() examples:  
SELECT COUNT(\*) FROM books;  
SELECT COUNT(author\_lname) FROM books;  
SELECT COUNT(DISTINCT author\_lname) FROM books;  
SELECT COUNT(\*) FROM books WHERE title LIKE '%the%';  
  
NULL 不会被计算在内。

### 6. GROUP BY（分组）

GROUP BY groups rows sharing the same values.  
GROUP BY 将具有相同值的行分为一组，常与聚合函数一起使用。  
  
SELECT author\_lname, COUNT(\*) FROM books GROUP BY author\_lname;  
  
Using alias and sorting:  
SELECT author\_lname, COUNT(\*) AS books\_written FROM books GROUP BY author\_lname ORDER BY books\_written DESC;

### 7. MIN() and MAX()

Return the minimum or maximum value.  
返回最小或最大值。  
  
SELECT MAX(pages) FROM books;  
SELECT MIN(author\_lname) FROM books;

### 8. Subqueries（子查询）

A subquery is a query inside another query.  
子查询是嵌套在另一条查询中的查询。  
  
SELECT title, pages FROM books WHERE pages = (SELECT MAX(pages) FROM books);  
SELECT title, released\_year FROM books WHERE released\_year = (SELECT MIN(released\_year) FROM books);  
  
先用内层查询找出最大或最小值，再用外层查询定位具体行。

### 9. GROUP BY Multiple Columns

Group results by multiple fields.  
按多个列进行分组。  
  
SELECT author\_fname, author\_lname, COUNT(\*) FROM books GROUP BY author\_lname, author\_fname;  
SELECT CONCAT(author\_fname, ' ', author\_lname) AS author, COUNT(\*) FROM books GROUP BY author;

### 10. MIN() / MAX() with GROUP BY

Find earliest and latest release for each author.  
计算每位作者最早与最新出版年份。  
  
SELECT author\_lname, COUNT(\*) AS books\_written, MAX(released\_year) AS latest\_release, MIN(released\_year) AS earliest\_release, MAX(pages) AS longest\_page\_count FROM books GROUP BY author\_lname;

### 11. SUM()

Add up numeric values.  
对数值型数据求和。  
  
SELECT SUM(pages) FROM books;  
SELECT author\_lname, COUNT(\*), SUM(pages) FROM books GROUP BY author\_lname;

### 12. AVG()

Compute the average value.  
计算平均值。  
  
SELECT AVG(pages) FROM books;  
SELECT AVG(released\_year) FROM books;  
SELECT released\_year, AVG(stock\_quantity), COUNT(\*) FROM books GROUP BY released\_year;

### References

Aggregate functions: https://dev.mysql.com/doc/refman/8.4/en/aggregate-functions.html  
String functions: https://dev.mysql.com/doc/refman/8.4/en/string-functions.html