MySQL LIKE 语句：

有时候我们需要获取 runoob_author 字段含有 "COM" 字符的所有记录，这时我们就需要在 WHERE 子句中使用 SQL LIKE 子句。SQL LIKE 子句中使用百分号%字符来表示任意字符，类似于 UNIX 或正则表达式中的星号*。如果没有使用百分号%， LIKE 子句与等号=的效果是一样的。

SQL SELECT 语句使用 LIKE 子句从数据表中读取数据的通用语法：

```
SELECT field1, field2,...fieldN
FROM table_name
WHERE field1 LIKE condition1 [AND [OR]] filed2 = 'somevalue'
```

注意：可以在 WHERE 子句中指定任何条件；可以在 WHERE 子句中使用 LIKE 子句；可以使用 LIKE 子句代替等号 =；LIKE 通常与 % 一同使用，类似于一个元字符的搜索；可以使用 AND 或者 OR 指定一个或多个条件；可以在 DELETE 或 UPDATE 命令中使用 WHERE...LIKE 子句来指定条件。

实例将 runoob_tbl 表中获取 runoob_author 字段中以 COM 为结尾的的所有记录：

```
mysql> SELECT * from runoob_tbl   WHERE runoob_author LIKE '%COM';
```

MySQL UNION 操作符：

MySQL UNION 操作符用于连接**两个以上的 SELECT 语句**的结果组合到一个结果集合中。多个 SELECT 语句会删除重复的数据。

MySQL UNION 操作符语法格式：

```
SELECT expression1, expression2, ... expression_n
FROM tables
[WHERE conditions]
UNION [ALL | DISTINCT]
SELECT expression1, expression2, ... expression_n
FROM tables
[WHERE conditions];
```

注意：

expression1, expression2, ... expression_n: 要检索的列；

tables: 要检索的数据表；
WHERE conditions: 可选，检索条件；
DISTINCT: 可选，删除结果集中重复的数据。默认情况下 UNION 操作符已经删除了重复数据，所以 DISTINCT 修饰符对结果没啥影响；
ALL: 可选，返回所有结果集，包含重复数据。

SQL UNION 实例：

```
SELECT country FROM Websites
UNION
SELECT country FROM apps
ORDER BY country;
```

SQL UNION ALL 实例：

```
SELECT country FROM Websites
UNION ALL
SELECT country FROM apps
ORDER BY country;
```

带有 WHERE 的 SQL UNION ALL 实例：

```
SELECT country, name FROM Websites
WHERE country='CN'
UNION ALL
SELECT country, app_name FROM apps
WHERE country='CN'
ORDER BY country;
```

MySQL 排序：

如果需要**对读取的数据进行排序**，就可以使用 MySQL 的 ORDER BY 子句来设定你想按**哪个字段哪种方式**来进行排序，再返回搜索结果。

以下是 SQL SELECT 语句使用 ORDER BY 子句将查询数据排序后再返回数据：

```
SELECT field1, field2,...fieldN FROM table_name1, table_name2...
ORDER BY field1 [ASC [DESC][默认 ASC]], [field2...] [ASC [DESC][默认 ASC]]
```

注意：可以使用任何字段来作为排序的条件，从而返回排序后的查询结果；可以设定多个字段来排序；可以使用 ASC 或 DESC 关键字来设置查询结果是按升序或降序排列，默认情况下，它是按升序排列；可以添加 WHERE...LIKE 子句

来设置条件。

实例：

mysql> SELECT * from runoob_tbl ORDER BY submission_date ASC;（按照时间顺序排列）

mysql> SELECT * from runoob_tbl ORDER BY submission_date DESC;（按照时间倒序排列）

MySQL GROUP BY 语句：

GROUP BY 语句根据**一个或多个列对结果集进行分组**；在分组的列上可以使用 COUNT、SUM、AVG 等函数。

GROUP BY 语法：

```
SELECT column_name, function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name;
```

实例：先创建新表

```
DROP TABLE IF EXISTS `employee_tbl`;
CREATE TABLE `employee_tbl` (
  `id` int(11) NOT NULL,
  `name` char(10) NOT NULL DEFAULT '',
  `date` datetime NOT NULL,
  `singin` tinyint(4) NOT NULL DEFAULT '0' COMMENT '登录次数',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
BEGIN;
INSERT INTO `employee_tbl` VALUES ('1', '小明', '2016-04-22 15:25:33', '1'), ('2', '小王', '2016-04-20 15:25:47', '3'), ('3', '小丽', '2016-04-19 15:26:02', '2'), ('4', '小王', '2016-04-07 15:26:14', '4'), ('5', '小明', '2016-04-11 15:26:40', '4'), ('6', '小明', '2016-04-04 15:26:54', '2');
COMMIT;
```

使用 GROUP BY 语句 将数据表按名字进行分组，并统计每个人有多少条记录：

```
mysql> SELECT name, COUNT(*) FROM employee_tbl GROUP BY name;
```

使用 WITH ROLLUP 可以实现在分组统计数据基础上再进行相同的统计（SUM,AVG,COUNT…）。例如我们将以上的数据表按名字进行分组，再统计每个人登录的次数：

mysql> SELECT name, SUM(singin) as singin_count FROM employee_tbl GROUP BY name WITH ROLLUP;

其中记录 NULL 表示所有人的登录次数。可以使用 coalesce 来设置一个可以取代 NUll 的名称，coalesce 语法：

select coalesce(a,b,c);

参数说明：如果 a==null,则选择 b；如果 b==null,则选择 c；如果 a!=null,则选择 a；如果 a b c 都为 null ，则返回为 null（没意义）。

mysql> SELECT coalesce(name, '总数'), SUM(singin) as singin_count FROM employee_tbl GROUP BY name WITH ROLLUP;