1. Select语法

SELECT

[ALL | DISTINCT | DISTINCTROW ] //在ALL和DISTINCT 修饰符指定重复行是否应该返回。 ALL（缺省）指定应该返回所有匹配的行，包括重复项。 DISTINCT指定从结果集中删除重复的行。

[HIGH\_PRIORITY]

[STRAIGHT\_JOIN]

[SQL\_SMALL\_RESULT] [SQL\_BIG\_RESULT] [SQL\_BUFFER\_RESULT]

[SQL\_CACHE | SQL\_NO\_CACHE] [SQL\_CALC\_FOUND\_ROWS]

*select\_expr* [, *select\_expr* ...] //每个select\_expr表示您想要检索的列。必须至少有一个 select\_expr。

[FROM *table\_references //table\_references指示从中检索行的表格。*

[PARTITION *partition\_list*] //SELECT支持显式分区选择，使用PARTITION列表中的分区或子分区（或两者）列表table\_reference中的表名称。在这种情况下，只从列出的分区中选择行，并忽略表中的任何其他分区。

[WHERE *where\_condition*] ///该WHERE条款如果给出，则表示行必须满足选择的条件。

[GROUP BY {*col\_name* | *expr* | *position*}

[ASC | DESC], ... [WITH ROLLUP]]

[HAVING *where\_condition*]

[ORDER BY {*col\_name* | *expr* | *position*}

[ASC | DESC], ...]

[LIMIT {[*offset*,] *row\_count* | *row\_count* OFFSET *offset*}]

[PROCEDURE *procedure\_name*(*argument\_list*)]

[INTO OUTFILE '*file\_name*'

[CHARACTER SET *charset\_name*]

*export\_options*

| INTO DUMPFILE '*file\_name*'

| INTO *var\_name* [, *var\_name*]]

[FOR UPDATE | LOCK IN SHARE MODE]]

1. [SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html)用于检索从一个或多个表中选择的行，并且可以包含 [UNION](https://dev.mysql.com/doc/refman/5.7/en/union.html)语句和子查询

2.通常，使用的子句必须按照语法描述中显示的顺序给出。例如，一个 HAVING条款必须在任何 GROUP BY条款之后，在任何ORDER BY条款之前。例外情况是，该 INTO子句可以显示在语法描述中，也可以紧跟在 ***select\_expr***列表之后。

3.表引用可以使用或 别名 ： ***tbl\_name*** AS ***alias\_nametbl\_name alias\_name***

SELECT t1.name, t2.salary FROM employee AS t1, info AS t2

WHERE t1.name = t2.name;

4．择用于输出的列可以 使用列名称，列别名或列位置在ORDER BY和GROUP BY子句中引用 。列位置是整数，以1开头： order by \*\*\* desc（降序排列）

SELECT college, region, seed FROM tournament

ORDER BY region, seed;

SELECT college, region AS r, seed AS s FROM tournament

ORDER BY r, s;

SELECT college, region, seed FROM tournament

ORDER BY 2, 3;

5. 如果ORDER BY在子查询中发生，并且在外部查询中应用，则最外层 ORDER BY优先。例如，以下语句的结果按降序排列，而不是升序：

(SELECT ... ORDER BY a) ORDER BY a DESC;

6. SQL标准要求HAVING必须仅引用GROUP BY 聚合函数中使用的子句或列中的列。但是，MySQL支持此行为的扩展，并允许 HAVING引用[SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html)列表中的 列和外部子查询中的列。如果该HAVING子句引用不明确的列，则会发生警告。在下面的语句中，col2由于它被用作别名和列名，所以是不明确的：

SELECT COUNT(col1) AS col2 FROM t GROUP BY col2 HAVING col2 = 2;

7. 不要HAVING用于应该在WHERE子句中的项目。例如，不要写下面的内容：

SELECT col\_name FROM tbl\_name HAVING col\_name > 0;

8. 该HAVING子句可以引用聚合函数，WHERE子句不能：

SELECT user, MAX(salary) FROM users

GROUP BY user HAVING MAX(salary) > 10;

9. LIMIT子句可以用来限制被返回的行数 [SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html)的语句。 LIMIT需要一个或两个数字参数，它们都必须是非负整数常量，但有以下例外：

* 在准备好的语句中，LIMIT 可以使用? 占位符标记来指定参数。
* 在存储的程序中，LIMIT 可以使用整数常数参数或局部变量指定参数。

…有两个参数，第一个参数指定要返回的第一行的偏移量，第二个参数指定要返回的最大行数。初始行的偏移量是0（不是1）:

SELECT \* FROM tbl LIMIT 5,10; #6-15hang

…要从某个偏移量直到结果集的末尾检索所有行，可以使用一些较大的数字作为第二个参数。这个语句检索从第96行到最后一行的所有行：

SELECT \* FROM tbl LIMIT 95,18446744073709551615;

…使用一个参数，该值指定从结果集的开头返回的行数：

SELECT \* FROM tbl LIMIT 5; # Retrieve first 5 rows

…对于准备好的语句，您可以使用占位符。以下语句将从tbl表中返回一行 ：

SET @a=1;

PREPARE STMT FROM 'SELECT \* FROM tbl LIMIT ?';

EXECUTE STMT USING @a;

…如果LIMIT在子查询中发生，并且在外部查询中应用，则最外层 LIMIT优先。例如，下面的语句产生两行，而不是一行：

(SELECT ... LIMIT 1) LIMIT 2;

1. Join语法

*join\_table*:

*table\_reference* [INNER | CROSS] JOIN *table\_factor* [*join\_condition*]

| *table\_reference* STRAIGHT\_JOIN *table\_factor*

| *table\_reference* STRAIGHT\_JOIN *table\_factor* ON *conditional\_expr*

| *table\_reference* {LEFT|RIGHT} [OUTER] JOIN *table\_reference* *join\_condition*

| *table\_reference* NATURAL [{LEFT|RIGHT} [OUTER]] JOIN *table\_factor*

*join\_condition*:

ON *conditional\_expr* | USING (*column\_list*)

*index\_hint\_list*:

*index\_hint* [, *index\_hint*] ...

*index\_hint*:

USE {INDEX|KEY}

[FOR {JOIN|ORDER BY|GROUP BY}] ([*index\_list*])

| IGNORE {INDEX|KEY}

[FOR {JOIN|ORDER BY|GROUP BY}] (*index\_list*)

| FORCE {INDEX|KEY}

[FOR {JOIN|ORDER BY|GROUP BY}] (*index\_list*)

*index\_list*:

*index\_name* [, *index\_name*] ...

1在MySQL， ，JOIN，CROSS JOIN和INNER JOIN是句法当量（它们可以彼此替换）。在标准的SQL中，它们不是等价的。INNER JOIN与ON子句一起 使用，否则使用CROSS JOIN。

2这样的子查询必须包含一个别名来给子查询结果一个表名。 ***table\_subquery***也被称为FROM 子句中的派生表或子查询：

SELECT \* FROM (SELECT 1, 2, 3) AS t1;

3INNER JOIN和, （逗号）在没有连接条件的情况下在语义上是等价的：都在指定的表之间产生笛卡尔乘积（也就是说，第一个表中的每一行都连接到第二个表中的每一行）。

4如果在右表中没有匹配的行 ON或USING在一个部分 LEFT JOIN，与设置所有列的列 NULL用于右表。您可以使用这个事实来查找在另一个表中没有对应表的行中的行：

SELECT left\_tbl.\*

FROM left\_tbl LEFT JOIN right\_tbl ON left\_tbl.id = right\_tbl.id

WHERE right\_tbl.id IS NULL;

5此示例查找所有行中不存在left\_tbl 的id值 right\_tbl（即left\_tbl没有对应行的 所有行 right\_tbl）。

6.Use用法：

SELECT \* FROM table1 LEFT JOIN table2 USING (id);

7. NATURAL不会显示连接的冗余列。

Eg

t1 (a,b) t2(a,c)

---- ----

1 x 2 z

2 y 3 w

然后，对于这个连接，列a包含以下值t1.a：

mysql> SELECT \* FROM t1 NATURAL LEFT JOIN t2;

+------+------+------+

| a | b | c |

+------+------+------+

| 1 | x | NULL |

| 2 | y | z |

+------+------+------+

相反，对于这个连接，列a 包含的值t2.a。

mysql> SELECT \* FROM t1 NATURAL RIGHT JOIN t2;

+------+------+------+

| a | c | b |

+------+------+------+

| 2 | z | y |

| 3 | w | NULL |

+------+------+------+

将这些结果与其他等效的查询进行比较JOIN ... ON：

mysql> SELECT \* FROM t1 LEFT JOIN t2 ON (t1.a = t2.a);

+------+------+------+------+

| a | b | a | c |

+------+------+------+------+

| 1 | x | NULL | NULL |

| 2 | y | 2 | z |

+------+------+------+------+

mysql> SELECT \* FROM t1 RIGHT JOIN t2 ON (t1.a = t2.a);

+------+------+------+------+

| a | b | a | c |

+------+------+------+------+

| 2 | y | 2 | z |

| NULL | NULL | 3 | w |

+------+------+------+------+

3.Union语法

SELECT ...

UNION [ALL | DISTINCT] SELECT ...

[UNION [ALL | DISTINCT] SELECT ..

[UNION](https://dev.mysql.com/doc/refman/5.7/en/union.html)用于将多个[SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html) 语句的结果合并到一个结果集中。

mysql> SELECT REPEAT('a',1) UNION SELECT REPEAT('b',10);

这些[SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html)语句是正常的选择语句，但有以下限制：

* 只有最后一个[SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html) 陈述可以使用INTO OUTFILE。（但是，整个[UNION](https://dev.mysql.com/doc/refman/5.7/en/union.html)结果写入文件。）

默认的行为[UNION](https://dev.mysql.com/doc/refman/5.7/en/union.html)是从结果中删除重复的行。可选的 DISTINCT关键字除了缺省值之外没有任何作用，因为它也指定了重复行删除。使用可选ALL关键字，重复行删除不会发生，结果包括所有[SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html)语句的所有匹配行。

要使用ORDER BY或 LIMIT子句对整个[UNION](https://dev.mysql.com/doc/refman/5.7/en/union.html)结果进行排序或限制 ，请将单个[SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html)语句括起来，并放在最后一个ORDER BY或 LIMIT之后。以下示例使用两个子句：

(SELECT a FROM t1 WHERE a=10 AND B=1)

UNION

(SELECT a FROM t2 WHERE a=11 AND B=2)

ORDER BY a LIMIT 10;