**SQL 快速参考**

**Table: Websites**

**+----+--------------+---------------------------+-------+---------+**

**| id | name | url | alexa | country |**

**+----+--------------+---------------------------+-------+---------+**

**| 1 | Google | https://www.google.cm/ | 1 | USA |**

**| 2 | 淘宝 | https://www.taobao.com/ | 13 | CN |**

**| 3 | 菜鸟教程 | http://www.runoob.com/ | 4689 | CN |**

**| 4 | 微博 | http://weibo.com/ | 20 | CN |**

**| 5 | Facebook | https://www.facebook.com/ | 3 | USA |**

**+----+--------------+---------------------------+-------+---------+**

**Table: access\_log**

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

| aid | site\_id | count | date |

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

| 1 | 1 | 45 | 2016-05-10 |

| 2 | 3 | 100 | 2016-05-13 |

| 3 | 1 | 230 | 2016-05-14 |

| 4 | 2 | 10 | 2016-05-14 |

| 5 | 5 | 205 | 2016-05-14 |

| 6 | 4 | 13 | 2016-05-15 |

| 7 | 3 | 220 | 2016-05-15 |

| 8 | 5 | 545 | 2016-05-16 |

| 9 | 3 | 201 | 2016-05-17 |

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

|  |  |  |
| --- | --- | --- |
| **SQL 语句** | **语法** | **示例** |
| AND / OR | SELECT column\_name(s) FROM table\_name WHERE condition AND|OR condition | SELECT \* FROM Websites WHERE alexa > 15 AND (country='CN' OR country='USA'); |
| ALTER TABLE | ALTER TABLE table\_name ADD column\_name datatype  or  ALTER TABLE table\_name DROP COLUMN column\_name | ALTER TABLE 语句  ALTER TABLE 语句用于在已有的表中添加、删除或修改列。 |
| AS (alias) | SELECT column\_name AS column\_alias FROM table\_name  or  SELECT column\_name FROM table\_name AS table\_alias |  |
| BETWEEN | SELECT column\_name(s) FROM table\_name WHERE column\_name BETWEEN value1 AND value2 | SELECT \* FROM Websites WHERE alexa BETWEEN 1 AND 20;  SELECT \* FROM Websites WHERE alexa NOT BETWEEN 1 AND 20; |
| CREATE DATABASE | CREATE DATABASE database\_name | CREATE DATABASE 语句用于创建数据库。 |
| CREATE TABLE | CREATE TABLE table\_name ( column\_name1 data\_type, column\_name2 data\_type, column\_name2 data\_type, ... ) | CREATE TABLE Persons ( PersonID int, LastName varchar(255), FirstName varchar(255), Address varchar(255), City varchar(255) ); |
| CREATE INDEX | CREATE INDEX index\_name ON table\_name (column\_name)  or  CREATE UNIQUE INDEX index\_name ON table\_name (column\_name) | SQL CREATE INDEX 语法  在表上创建一个简单的索引。允许使用重复的值  SQL CREATE UNIQUE INDEX 语法  在表上创建一个唯一的索引。不允许使用重复的值：唯一的索引意味着两个行不能拥有相同的索引值  下面的 SQL 语句在 "Persons" 表的 "LastName" 列上创建一个名为 "PIndex" 的索引：  CREATE INDEX PIndex  ON Persons (LastName)  如果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 PIndex  ON Persons (LastName, FirstName) |
| CREATE VIEW | CREATE VIEW view\_name AS SELECT column\_name(s) FROM table\_name WHERE condition | 在 SQL 中，视图是基于 SQL 语句的结果集的可视化的表。  视图包含行和列，就像一个真实的表。视图中的字段就是来自一个或多个数据库中的真实的表中的字段。  您可以向视图添加 SQL 函数、WHERE 以及 JOIN 语句，也可以呈现数据，就像这些数据来自于某个单一的表一样。 |
| DELETE | DELETE FROM table\_name WHERE some\_column=some\_value  or  DELETE FROM table\_name (**Note:**Deletes the entire table!!)  DELETE \* FROM table\_name (**Note:**Deletes the entire table!!) |  |
| DROP DATABASE | DROP DATABASE database\_name | 通过使用 DROP 语句，可以轻松地删除索引、表和数据库。 |
| DROP INDEX | DROP INDEX table\_name.index\_name (SQL Server) DROP INDEX index\_name ON table\_name (MS Access) DROP INDEX index\_name (DB2/Oracle) ALTER TABLE table\_name DROP INDEX index\_name (MySQL) | 删除表中的索引 |
| DROP TABLE | DROP TABLE table\_name | 删除表 |
| GROUP BY | SELECT column\_name, aggregate\_function(column\_name) FROM table\_name WHERE column\_name operator value GROUP BY column\_name | SELECT site\_id, SUM(access\_log.count) AS nums  FROM access\_log GROUP BY site\_id; |
| HAVING | SELECT column\_name, aggregate\_function(column\_name) FROM table\_name WHERE column\_name operator value GROUP BY column\_name HAVING aggregate\_function(column\_name) operator value | SELECT Websites.name, Websites.url, SUM(access\_log.count) AS nums FROM (access\_log INNER JOIN Websites ON access\_log.site\_id=Websites.id) GROUP BY Websites.name HAVING SUM(access\_log.count) > 200; |
| IN | SELECT column\_name(s) FROM table\_name WHERE column\_name IN (value1,value2,..) |  |
| INSERT INTO | INSERT INTO table\_name VALUES (value1, value2, value3,....)  *or*  INSERT INTO table\_name (column1, column2, column3,...) VALUES (value1, value2, value3,....) | INSERT INTO Websites (name, url, alexa, country) VALUES ('百度','https://www.baidu.com/','4','CN'); |
| INNER JOIN | SELECT column\_name(s) FROM table\_name1 INNER JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name | SQL INNER JOIN  INNER JOIN 关键字在表中存在至少一个匹配时返回行。  SELECT Websites.name, access\_log.count, access\_log.date FROM Websites INNER JOIN access\_log ON Websites.id=access\_log.site\_id ORDER BY access\_log.count; |
| LEFT JOIN | SELECT column\_name(s) FROM table\_name1 LEFT JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name | SQL LEFT JOIN  LEFT JOIN 关键字从左表（table1）返回所有的行，即使右表（table2）中没有匹配。如果右表中没有匹配，则结果为 NULL。  SELECT Websites.name, access\_log.count, access\_log.date FROM Websites LEFT JOIN access\_log ON Websites.id=access\_log.site\_id ORDER BY access\_log.count DESC; |
| RIGHT JOIN | SELECT column\_name(s) FROM table\_name1 RIGHT JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name | SQL RIGHT JOIN  RIGHT JOIN 关键字从右表（table2）返回所有的行，即使左表（table1）中没有匹配。如果左表中没有匹配，则结果为 NULL。  SELECT websites.name, access\_log.count, access\_log.date FROM websites RIGHT JOIN access\_log ON access\_log.site\_id=websites.id ORDER BY access\_log.count DESC; |
| FULL JOIN | SELECT column\_name(s) FROM table\_name1 FULL JOIN table\_name2 ON table\_name1.column\_name=table\_name2.column\_name | SQL FULL OUTER JOIN  FULL OUTER JOIN 关键字只要左表（table1）和右表（table2）其中一个表中存在匹配，则返回行.  SELECT Websites.name, access\_log.count, access\_log.date FROM Websites FULL OUTER JOIN access\_log ON Websites.id=access\_log.site\_id ORDER BY access\_log.count DESC; |
| LIKE | SELECT column\_name(s) FROM table\_name WHERE column\_name LIKE pattern |  |
| ORDER BY | SELECT column\_name(s) FROM table\_name ORDER BY column\_name [ASC|DESC] | SELECT column1, column2, ...  FROM table\_name  ORDER BY column1, column2, ... ASC|DESC;  column1, column2, ...：要排序的字段名称，可以为多个字段。  ASC：表示按升序排序。  DESC：表示按降序排序。 |
| SELECT | SELECT column\_name(s) FROM table\_name |  |
| SELECT \* | SELECT \* FROM table\_name |  |
| SELECT DISTINCT | SELECT DISTINCT column\_name(s) FROM table\_name | 在表中，一个列可能会包含多个重复值，有时您也许希望仅仅列出不同（distinct）的值。  DISTINCT 关键词用于返回唯一不同的值。 |
| SELECT INTO | SELECT \* INTO new\_table\_name [IN externaldatabase] FROM old\_table\_name  *or*  SELECT column\_name(s) INTO new\_table\_name [IN externaldatabase] FROM old\_table\_name |  |
| SELECT TOP | SELECT TOP number|percent column\_name(s) FROM table\_name | SELECT TOP 子句用于规定要返回的记录的数目。 |
| TRUNCATE TABLE | TRUNCATE TABLE table\_name |  |
| UNION | SELECT column\_name(s) FROM table\_name1 UNION SELECT column\_name(s) FROM table\_name2 | SQL UNION 操作符合并两个或多个 SELECT 语句的结果。  UNION 操作符选取不同的值 |
| UNION ALL | SELECT column\_name(s) FROM table\_name1 UNION ALL SELECT column\_name(s) FROM table\_name2 | UNION ALL允许重复的值 |
| UPDATE | UPDATE table\_name SET column1=value, column2=value,... WHERE some\_column=some\_value | UPDATE 语句用于更新表中的记录。  请注意 SQL UPDATE 语句中的 WHERE 子句！WHERE 子句规定哪条记录或者哪些记录需要更新。如果您省略了 WHERE 子句，所有的记录都将被更新！ |
| WHERE | SELECT column\_name(s) FROM table\_name WHERE column\_name operator value |  |