|  |  |  |
| --- | --- | --- |
| **SQL 语句** | **语法** | **介绍** |
| AND / OR | SELECT column\_name(s) FROM table\_name WHERE condition AND|OR condition | AND 和 OR 可在 WHERE 子语句中把两个或多个条件结合起来。  如果第一个条件和第二个条件都成立，则 AND 运算符显示一条记录。  如果第一个条件和第二个条件中只要有一个成立，则 OR 运算符显示一条记录。 |
| ALTER TABLE (add column) | ALTER TABLE table\_name  ADD column\_name datatype | ALTER TABLE 语句用于在已有的表中添加列。 |
| ALTER TABLE (drop column) | ALTER TABLE table\_name  DROP COLUMN column\_name | 删除表中的列 |
| AS (alias for column) | SELECT column\_name AS column\_alias FROM table\_name | 为列指定别名 |
| AS (alias for table) | SELECT column\_name FROM table\_name  AS table\_alias | 为表指定别名 |
| BETWEEN | SELECT column\_name(s) FROM table\_name WHERE column\_name BETWEEN value1 AND value2 | 选取介于两个值之间的数据范围。这些值可以是数值、文本或者日期。 |
| CREATE DATABASE | CREATE DATABASE database\_name | 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, ... ) | 用于创建数据库中的表。 |
| CREATE INDEX | CREATE INDEX index\_name ON table\_name (column\_name) | **用于在表中创建索引。**  **在不读取整个表的情况下，索引使数据库应用程序可以更快地查找数据。**  在表上创建一个简单的索引。允许使用重复的值： |
| CREATE UNIQUE INDEX | CREATE UNIQUE INDEX index\_name ON table\_name (column\_name) | 在表上创建一个唯一的索引。唯一的索引意味着两个行不能拥有相同的索引值。 |
| CREATE VIEW | CREATE VIEW view\_name AS SELECT column\_name(s) FROM table\_name WHERE condition | **创建、更新和删除视图。** |
| DELETE | DELETE FROM table\_name WHERE some\_column=some\_value  **删除某行**  or  DELETE FROM table\_name  (**Note:**Deletes the entire table!!)  DELETE \* FROM table\_name  (**Note:**Deletes the entire table!!) 删除所有行 | 用于删除表中的行。 |
| DROP DATABASE | DROP DATABASE database\_name | 删除数据库 |
| 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 | 用于结合合计函数，根据一个或多个列对结果集进行分组 |
| 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 | WHERE 关键字无法与聚合函数一起使用。  HAVING 子句可以让我们筛选分组后的各组数据 |
| IN | SELECT column\_name(s) FROM table\_name WHERE column\_name IN (value1,value2,..) | 允许在 WHERE 子句中规定多个值。 |
| INSERT INTO | INSERT INTO table\_name VALUES (value1, value2, value3,....)  第一种形式无需指定要插入数据的列名，只需提供被插入的值即可：  *or*  INSERT INTO table\_name (column1, column2, column3,...) VALUES (value1, value2, value3,....)  第二种形式需要指定列名及被插入的值： | 向表中插入新记录。 |
| INNER JOIN | SELECT column\_name(s) FROM table\_name1 INNER JOIN table\_name2  ON table\_name1.column\_name=table\_name2.column\_name | INNER JOIN 关键字在表中存在至少一个匹配时返回行。 |
| LEFT JOIN | SELECT column\_name(s) FROM table\_name1 LEFT JOIN table\_name2  ON table\_name1.column\_name=table\_name2.column\_name | 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。 |
| RIGHT JOIN | SELECT column\_name(s) FROM table\_name1 RIGHT JOIN table\_name2  ON table\_name1.column\_name=table\_name2.column\_name | 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。 |
| FULL JOIN | SELECT column\_name(s) FROM table\_name1 FULL JOIN table\_name2  ON table\_name1.column\_name=table\_name2.column\_name | 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.  FULL OUTER JOIN 关键字结合了 LEFT JOIN 和 RIGHT JOIN 的结果。 |
| LIKE | SELECT column\_name(s) FROM table\_name WHERE column\_name LIKE pattern | LIKE 操作符用于在 WHERE 子句中搜索列中的指定模式。 |
| ORDER BY | SELECT column\_name(s) FROM table\_name ORDER BY column\_name [ASC|DESC] | ORDER BY 关键字用于对结果集按照一个列或者多个列进行排序。  ORDER BY 关键字默认按照升序对记录进行排序。如果需要按照降序对记录进行排序，您可以使用 DESC 关键字。 |
| SELECT | SELECT column\_name(s) FROM table\_name | SELECT 语句用于从数据库中选取数据。  结果被存储在一个结果表中，称为结果集。 |
| SELECT \* | SELECT \* FROM table\_name | 从 " table\_name" 表中选取所有列 |
| SELECT DISTINCT | SELECT DISTINCT column\_name(s) FROM table\_name | 用于返回唯一不同的值。 |
| SELECT INTO | SELECT \* INTO new\_table\_name [IN externaldatabase] FROM old\_table\_name  把所有的列插入新表：  *or*  SELECT column\_name(s) INTO new\_table\_name [IN externaldatabase] FROM old\_table\_name  把希望的列插入新表： | SELECT INTO 语句用于创建表的备份复件。  SELECT INTO 语句从一个表中选取数据，然后把数据插入另一个表中。  SELECT INTO 语句常用于创建表的备份复件或者用于对记录进行存档。 |
| SELECT TOP | SELECT TOP number|percent column\_name(s) FROM table\_name | 规定要返回的记录的数目。  ***注意:****并非所有的数据库系统都支持 SELECT TOP 语句。 MySQL 支持 LIMIT 语句来选取指定的条数数据， Oracle 可以使用 ROWNUM 来选取。* |
| TRUNCATE TABLE | TRUNCATE TABLE table\_name | 表中的资料会完全消失，可是表格本身会继续存在 |
| UNION | SELECT column\_name(s) FROM table\_name1 UNION SELECT column\_name(s) FROM table\_name2 | 合并两个或多个 SELECT 语句的结果. |
| UNION ALL | SELECT column\_name(s) FROM table\_name1 UNION ALL SELECT column\_name(s) FROM table\_name2 | 默认地，UNION 操作符选取不同的值。如果允许重复的值，请使用 UNION ALL。  UNION 结果集中的列名总是等于 UNION 中第一个 SELECT 语句中的列名。 |
| UPDATE | UPDATE table\_name SET column1=value, column2=value,... WHERE some\_column=some\_value | 用于更新表中已存在的记录。 |
| WHERE | SELECT column\_name(s) FROM table\_name WHERE column\_name operator value | 用于过滤记录。提取那些满足指定标准的记录。 |