

SQL 常用语句词典

参考: <https://zhuanlan.zhihu.com/p/113941048>

一、查 SELECT

[检索单个列]	SELECT [列名] FROM [表名];																				
[检索多个列]	SELECT [列名 1], [列名 2] FROM [表名];																				
[检索所有列]	SELECT * FROM [表名];																				
[剔除重复行]	SELECT DISTINCT [列名 1][列名 2] FROM [表名]; # DISTINCT 作用于其后面的所有列																				
[限制显示的行]	SELECT [列名] FROM [表名] LIMIT 5 OFFSET 5; #LIMIT 限制 5 行、OFFSET 从第 6 行起, 即 6-10 行 #第一个被检索的行是第 0 行																				
[按列排序数据]	SELECT [列名 1],[列名 2] FROM [表名] ORDER BY [列名 1],[列名 2]; #可用非检索列排序, 如[列名 3]																				
[指定排序方向]	SELECT [列名 1],[列名 2] FROM [表名] ORDER BY [列名 1],[列名 2] DESC; #按列 1 升序、列 2 降序排序 #默认升序 ASC, 降序 DESC, ASC 和 DESC 只作用于前一列																				
[单条件查找数据]	SELECT [列名] FROM [表名] WHERE 条件 #一般条件表达: [列名][操作符][条件值], 如 price= 10 #操作符及说明 (NOT 位置不同于其他操作符)																				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">=</td> <td style="text-align: center;">等于</td> <td style="text-align: center;">></td> <td style="text-align: center;">大于</td> </tr> <tr> <td style="text-align: center;"><>或!=</td> <td style="text-align: center;">不等于</td> <td style="text-align: center;">>=</td> <td style="text-align: center;">大于等于</td> </tr> <tr> <td style="text-align: center;"><</td> <td style="text-align: center;">小于</td> <td style="text-align: center;">BETWEEN AND</td> <td style="text-align: center;">两值之间</td> </tr> <tr> <td style="text-align: center;"><=</td> <td style="text-align: center;">小于等于</td> <td style="text-align: center;">IS NULL</td> <td style="text-align: center;">空值检查</td> </tr> <tr> <td style="text-align: center;">IN</td> <td style="text-align: center;">指定条件范围</td> <td style="text-align: center;">NOT</td> <td style="text-align: center;">否定全部条件, WHERE NOT 条件</td> </tr> </table>	=	等于	>	大于	<>或!=	不等于	>=	大于等于	<	小于	BETWEEN AND	两值之间	<=	小于等于	IS NULL	空值检查	IN	指定条件范围	NOT	否定全部条件, WHERE NOT 条件
=	等于	>	大于																		
<>或!=	不等于	>=	大于等于																		
<	小于	BETWEEN AND	两值之间																		
<=	小于等于	IS NULL	空值检查																		
IN	指定条件范围	NOT	否定全部条件, WHERE NOT 条件																		
	#条件值为字符是需要用单引号括起来, 如'helloworld'																				
[多条件查找数据]	SELECT [列名] FROM [表名] WHERE 条件 1 AND/OR 条件 2 #AND 表示同时满足, OR 表示满足其一即可 #优先级: 括号 > AND > OR																				
[通配符查找文本数据]	SELECT [列名] FROM [表名]																				

	<p>WHERE [列名] LIKE [文本+通配符]</p> <p>#通配符唯一操作符 LIKE</p> <p>#通配符仅用于文本搜索</p> <table border="1"> <tr> <td>百分号%</td> <td>表示任意字符出现任意次数</td> <td>'Fish%':以Fish开头的文本</td> </tr> <tr> <td>下划线_</td> <td>表示任意单个字符</td> <td></td> </tr> <tr> <td>方括号[]</td> <td>指定字符集</td> <td>'[JM]':以J或M开头的文本</td> </tr> <tr> <td>前缀^</td> <td>表否定</td> <td>'[^JM]':不以J或M开头的文本</td> </tr> </table>	百分号%	表示任意字符出现任意次数	'Fish%':以Fish开头的文本	下划线_	表示任意单个字符		方括号[]	指定字符集	'[JM]':以J或M开头的文本	前缀^	表否定	'[^JM]':不以J或M开头的文本
百分号%	表示任意字符出现任意次数	'Fish%':以Fish开头的文本											
下划线_	表示任意单个字符												
方括号[]	指定字符集	'[JM]':以J或M开头的文本											
前缀^	表否定	'[^JM]':不以J或M开头的文本											
[拼接文本字段]	<p>SELECT [列名 1]+[列名 2]</p> <p>AS 列名 1_列名 2</p> <p>FROM [表名];</p> <p>#拼接列 1 和列 2 的值, 并取别名: 列名 1_列名 2</p>												
[算术运算]	<p>SELECT [列名 1] *或+或-或/ [列名 2]</p> <p>AS 列名 1_列名 2</p> <p>FROM [表名];</p>												
[汇总数据]	<p>SELECT COUNT (列名 1) AS count 列名 1, #计算行数</p> <p>AVG(列名 1) AS avg_列名 1, #计算均值</p> <p>MAX(列名 1) AS MAX_列名 1, #计算最大值</p> <p>MIN(列名 1) AS MIN_列名 1, #计算最小值</p> <p>SUM(列名 1) AS SUM_列名 1, #求和</p> <p>FROM [表名];</p> <p>#使用聚集函数对数据进行汇总, 可单独或组合使用聚集函数</p>												
[数据分组汇总]	<p>SELECT [列名 1],[列名 2],COUNT(*) #可用其他函数</p> <p>FROM [表名]</p> <p>GROUP BY [列名 1],[列名 2];</p> <p>#GROUP BY 可包含任意数目的列。如果为多列, 数据将在最后指定的分组上进行汇总</p>												
[过滤分组]	<p>SELECT [列名 1],COUNT(*) #可用其他函数</p> <p>FROM [表名]</p> <p>GROUP BY [列名 1]</p> <p>HAVING 条件; #如 COUNT(*)>2</p>												
下面是稍微复杂的一些语句介绍													
[子查询过滤]	<p>SELECT [列名 1]</p> <p>FROM [表名 1]</p> <p>WHERE [列名 1] IN (SELECT [列名 1] FROM [表名 2] WHERE 条件);</p> <p>#子查询的 SELECT 只能查询单列</p> <p>#子查询可作为条件值放在 WHERE 子句后, 也可以作为计算字段放在 SELECT 后</p> <p>#相较于之前的查询, 子查询可用于跨表操作</p>												
[子查询过滤]	<p>SELECT [列名 1],</p> <p>(SELECT COUNT(*)</p> <p>FROM [表名 2])</p>												

	<p>WHERE [表名 2].[列名 1]=[表名 1].[列名 1]) FROM [表名 1];</p> <p>#列名 1 是两个表共有的列</p>
[拓展]	<p>这部分内容有助于子查询的理解 数据库通常使用关系表，表只存储同类信息，表通过外键与主键相关联。 举例来说： Order 表 存储<u>订单编号</u>、<u>顾客 ID</u>、订单日期； OrderItems 表 存储<u>物品 ID</u>、物品名、<u>订单编号</u>； Customers 表 存储<u>顾客 ID</u>、顾客姓名、顾客年龄； #下划直线表示主键、下滑波浪线表示外键 #便于理解，用中文命名</p> <p>Order 表与 Customers 表通过顾客 ID 关联，OrderItems 表与 Order 表通过订单编号关联。在这样的关系表基础上，使用子查询就可以通过一个语句实现跨表查询。</p> <p>查询订购物品 泡芙 的所有顾客信息，可以用子查询实现： SELECT 顾客姓名，顾客年龄 FROM Customers WHERE 顾客 ID IN (SELECT 顾客 ID FROM Order WHERE 订单编号 IN (SELECT 订单编号 FROM OrderItems WHERE 物品名 = 泡芙)；</p>
[联结两个表]	<p>结合上面的关系表介绍 一次查询 顾客姓名、顾客年龄、订单日期： SELECT 顾客姓名,顾客年龄,订单日期 FROM Customers INNER JOIN Order ON Customers. 顾客 ID = Order. 顾客 ID; 或 SELECT 顾客姓名,顾客年龄,订单日期 FROM Customers,Order WHERE Customers. 顾客 ID = Order. 顾客 ID;</p>
[联结多个表]	<p>结合上面的关系表介绍 一次查询 顾客姓名、顾客年龄、订单编号、订单日期、物品名： SELECT 顾客姓名,顾客年龄,订单编号,订单日期,物品名 FROM Customers,Order,OrderItems WHERE Customers. 顾客 ID = Order. 顾客 ID AND Order. 订单编号 = OrderItems. 订单编号;</p>
[自联结]	<p>结合上面的关系表介绍 一次查询 与小王年龄相同的所有顾客： SELECT c1. 顾客姓名 FROM Customers AS c1,Customers AS c2</p>

	<p>WHERE c1. 顾客年龄 = c2. 顾客年龄 AND c2. 顾客姓名 = 小王;</p>
[外联结]	<p>结合上面的关系表介绍 检索没有下过订单的顾客信息: SELECT 顾客姓名, 顾客年龄, 订单编号 FROM Customers LEFT OUTER JOIN Order ON Customers. 顾客 ID = Order. 顾客 ID; #不用于内联结, 外联结还包括没有关联的行 #LEFT 表示左边的表显示所有行, RIGHT 表示右边的表显示</p>
[组合查询]	<p>SELECT [列名 1], [列名 2], [列名 3] FROM [表名 1] WHERE 条件 1 UNION SELECT [列名 1], [列名 2], [列名 3] FROM [表名 1] WHERE 条件 2 ORDER BY [列名 1]; #UNION 会自动删除重复行, 不想删除用 UNION ALL #两个 SELECT 子句要包括相同的列、表达式或聚集函数</p>

二、增 INSERT、删 DELETE、改 UPDATE

[插入数据]	<p>INSERT INTO [表名] ([列名 1], [列名 2], [列名 3], [列名 4]) VALUES('列 1 值', '列 2 值', '列 3 值', '列 4 值');</p> <p>#可插入全部列或部分列, 省略的列要满足: 允许 NULL 或给出默认值</p>
[插入检索数据]	<p>INSERT INTO [表名 1] ([列名 1], [列名 2], [列名 3], [列名 4]) SELECT [列名 1], [列名 2], [列名 3], [列名 4] FROM [表名 2];</p> <p>#SELECT 列名不要求与 INSERT 列名匹配, 看返回列的位置, SELECT</p>

	返回第 1 列填充指定表第 1 列。
[表复制]	SELECT * INTO [新表名] FROM [表名]; #创建新表, 并把原表内容全部复制到新表
[更改数据]	UPDATE [表名] SET [列名 1]='文本' [列名 2]=数值 WHERE 条件;
[删除数据]	DELETE FROM [表名] WHERE 条件;

三、 创建、操作表

[创建表]	CREATE TABLE [表名] ([列名] 数据类型 是否允许空值(选填) 默认值(选填), order_num INTEGER NOT NULL, order_date DATETIME NOT NULL DEFAULT GETDATE(), cust_id CHAR(10) NOT NULL, order_price DECIMAL(8,2) NOT NULL,);
[更新表]	ALTER TABLE [表名] ADD [列名 1] 数据类型 是否允许空值 默认值 DROP COLUMN [列名 2];
[删除表]	DROP TABLE [表名];

四、 辅助工具

[创建视图]	CREATE VIEW [视图名] AS SELECT * FROM [列名], WHERE 条件;
[使用视图]	SELECT * FROM [视图名];