Atitit 数据库的重要功能一览

目录

[1. Sql扩展 1](#_Toc29265)

[2. 子查询 2](#_Toc16238)

[2.1. 子查询为json字段 2](#_Toc26865)

[2.2. 子查询作为json返回 2](#_Toc14024)

[3. Json字段支持 2](#_Toc14161)

[3.1. JSON表函数 2](#_Toc23547)

[4. 导入导出数据 3](#_Toc3551)

[5. 外部 脚ben java 存储过程 3](#_Toc8319)

[6. Other 3](#_Toc6122)

[6.1. 窗口函数(Window Functions)： 3](#_Toc27446)

[7. Join 3](#_Toc11864)

[8. 性能提升 雾化视图 4](#_Toc12500)

[8.1. 函数索引 4](#_Toc31972)

[8.2. DO SLEEP(5) 4](#_Toc14489)

# Sql扩展

Inset set

Replace into

Insert ingore

Merge融合语句 Insert On unique update

排除某一列

对于[INSERT ... SELECT](https://dev.mysql.com/doc/refman/8.0/en/insert-on-duplicate.html" \o "13.2.6.2 INSERT ... ON DUPLICATE KEY UPDATE语句)语句，请参见 [第13.2.6.2节“在重复键更新语句上插入...”](https://dev.mysql.com/doc/refman/8.0/en/insert-on-duplicate.html" \o "13.2.6.2 INSERT ... ON DUPLICATE KEY UPDATE语句)，[SELECT](https://dev.mysql.com/doc/refman/8.0/en/select.html" \o "13.2.10 SELECT语句)以了解在ON DUPLICATE KEY UPDATE 子句中[可以引用](https://dev.mysql.com/doc/refman/8.0/en/insert-on-duplicate.html" \o "13.2.6.2 INSERT ... ON DUPLICATE KEY UPDATE Statement)列的条件。这也适用于INSERT ... TABLE。

不带子句 的[SELECT](https://dev.mysql.com/doc/refman/8.0/en/select.html" \o "13.2.10 SELECT语句)or [TABLE](https://dev.mysql.com/doc/refman/8.0/en/table.html" \o "13.2.12 TABLE语句)语句 ORDER BY返回行的顺序是不确定的。这意味着，在使用复制时，不能保证这样的 [SELECT](https://dev.mysql.com/doc/refman/8.0/en/select.html" \o "13.2.10 SELECT语句)返回在主服务器和从服务器上以相同的顺序返回行，这可能导致它们之间的不一致。为了防止这种情况发生，总是写INSERT ... SELECT或 INSERT ... TABLE于使用复制陈述，ORDER BY产生对主单元和从属同一行顺序子句。另请参见[第17.5.1.18节“复制和限制”](https://dev.mysql.com/doc/refman/8.0/en/replication-features-limit.html" \o "17.5.1.18复制和限制)。

由于此问题， 对于基于语句的复制[INSERT ... SELECT ON DUPLICATE KEY UPDATE](https://dev.mysql.com/doc/refman/8.0/en/insert-on-duplicate.html" \o "13.2.6.2 INSERT ... ON DUPLICATE KEY UPDATE语句)， [INSERT IGNORE ... SELECT](https://dev.mysql.com/doc/refman/8.0/en/insert-select.html" \o "13.2.6.1 INSERT ... SELECT语句)语句被

INSERT  ... ON DUPLICATE KEY UPDATE语法

INSERT后面是插入语法，UPDATE后面是更新语法。

当没有键值冲突时，执行INSERT操作，当有键值冲突时，执行UPDATE操作。

应用场景，比如实现计数功能，type为计数类别，有unique index，那么一条SQL，就能解决所有问题，而不需要检查该类型是否存在：

# 子查询

## 子查询为json字段

## 子查询作为json返回

Mssql可以，mysql不可以，手动凭借

# Json字段支持

## JSON表函数

MySQL 8.0增加了JSON表函数，可以使用JSON数据的SQL机制。JSON\_TABLE()创建JSON数据的关系视图。它将JSON数据评估的结果映射到关系行和列。用户可以使用SQL查询函数返回的结果为常规关系表，例如join，project和aggregate。

作者：秦泽超  
链接：https://www.jianshu.com/p/c33433573c1c  
来源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

1. ****通用表表达式(Common Table Expressions CTE)****：

在复杂的查询中使用嵌入式表时，使用 CTE 使得查询语句更清晰

# 导入导出数据

[SELECT ... INTO Statement](https://dev.mysql.com/doc/refman/8.0/en/select-into.html)

*into\_option*: { INTO OUTFILE '*file\_name*' [CHARACTER SET *charset\_name*] *export\_options* | INTO DUMPFILE '*file\_name*' | INTO *var\_name* [, *var\_name*] ...

# 外部 脚ben java 存储过程

# Other

## ****窗口函数(Window Functions)****：

1. 从 MySQL 8.0 开始，新增了一个叫窗口函数的概念，它可以用来实现若干新的查询方式。窗口函数与 SUM()、COUNT() 这种集合函数类似，但它不会将多行查询结果合并为一行，而是将结果放回多行当中。即窗口函数不需要 GROUP BY。

# ****Join****

MySQL中cross join等同于inner join，多个表用逗号分隔，在无联合条件下与inner join是语义相同的。

可以使用STRAIGHT\_JOIN强制左表在右表之前被读取。STRAIGH\_JOIN可以被用于这样的情况，即联合优化符以错误的顺序排列表。

MYSQL扩展了SQL标准语法，可以使用下面的方式连接：

SELECT \* FROM t1 LEFT JOIN (t2, t3, t4) ON (t2.a=t1.a AND t3.b=t1.b AND t4.c=t1.c)

相当于：

SELECT \* FROM t1 LEFT JOIN (t2 INNER JOIN t3 INNER JOIN t4) ON (t2.a=t1.a AND t3.b=t1.b AND t4.c=t1.c);

可以使用USING(column\_list)子句为一系列的列进行命名。这些列必须同时在两个表中存在。如果表a和表b都包含列c1, c2和c3，则以下联合会对比来自两个表的对应的列：

a LEFT JOIN b USING (c1,c2,c3) 《=》 a left join b on a.c1= b.c1 and a.c2=b.c2 and a.c3=b.c3

# 性能提升 雾化视图

Other

## 函数索引

## DO SLEEP(5)

DO用于执行表达式，但是不返回任何结果。DO是SELECT expr的简化表达方式。DO有一个优势，就是如果您不太关心结果的话，DO的速度稍快。

DO主要用于执行有副作用的函数，比如RELEASE\_LOCK()。

[DO](https://dev.mysql.com/doc/refman/8.0/en/do.html" \o "13.2.3 DO声明)执行表达式，但不返回任何结果

Atitit.mysql 5.0 5.5 5.6 5.7 新特性 新功能

MySql中特有的语法 - 风生水起 - 博客园.html

Mysql 8新特性