



9-3-1、upsert语法

(1) 参考

- [SQL 中的Upsert：什么是Upsert，何时应该使用Upsert？ - 墨天轮 \(modb.pro\)](#)

(2) upsert介绍

- + • 是INSERT与UPDATE的结合体，在关系型数据库的上下文中，upsert 指的是一种数据库操作，表示行存在时执行UPDATE语法，不存在时执行INSERT。
- 执行UPSERT的时候必须指定完全的primary key的相关列的信息。UPSERT语法支持批量写入。
- :: • 事实上，在许多RDBMS中，UPSERT甚至不作为命令存在！这就是为什么如果在文档中搜索所选数据库，可能找不到“upsert”的条目。

(3) mysql数据库

- 不支持直接编写upsert语法，而是通过插入数据，**主键冲突时则更新，不冲突则插入**，来达到upsert的功能

复制代码

```
1
2 INSERT INTO employees (id, name, email)
3 VALUES (1, 'Dennis', 'dennisp@weyland.corp')
4 ON DUPLICATE KEY UPDATE
```

sql >

(4) PostgreSQL数据库

- 不支持直接编写upsert语法，而又比mysql更复杂一点，因为在**重复键上**，它允许我们有更多的控制，插入数据时，可以指定冲突列，在**冲突列冲突时则更新，不冲突则插入，不单单只支持主键列冲突更新**，这使我们在如何应用我们的优势方面更有针对性。
- conflict_target，冲突目标，即应在哪个字段上检测冲突。
- conflict_action，冲突动作，即在冲突列上检测到冲突时应执行什么命令。

[复制代码](#)

```
1 INSERT INTO table (col1, col2, col3)
2 VALUES (val1, val2, val3)
3 ON CONFLICT conflict_target conflict_action;
```

[sql >](#)

- 我们可以**手动指定冲突目标**（在本例中为name，即指定了name列是冲突目标）以及在检测到冲突时要执行的操作（在本例中，我们是更新现有行）来实现upsert功能。
 - 我们试图插入一个name值为Dennis的行，如果员工中已经存在一个name为Dennis的行，对该行运行更新，不存在name为Dennis的行，对该行插入

[复制代码](#)

```
1
2 INSERT INTO employees (id, name, email)
3 VALUES (1, 'Dennis', 'dennisp@weyland.corp')
4 ON CONFLICT (name) DO UPDATE;
```

[sql >](#)

（5）CockroachDB数据库

- 支持直接编写upsert语法，和postgresql数据库一样，也是冲突时更新，但是CockroachDB的upsert不能指定冲突目标，**默认只支持冲突目标是主键列（单一、联合主键）**，冲突行为是更新，只要主键冲突就更新，不冲突就插入，来实现upsert功能

[复制代码](#)

```
1 UPSERT INTO employees (id, name, email)
2 VALUES (6, 'Lambert', 'lambert@weyland.corp');
```

[sql >](#)

