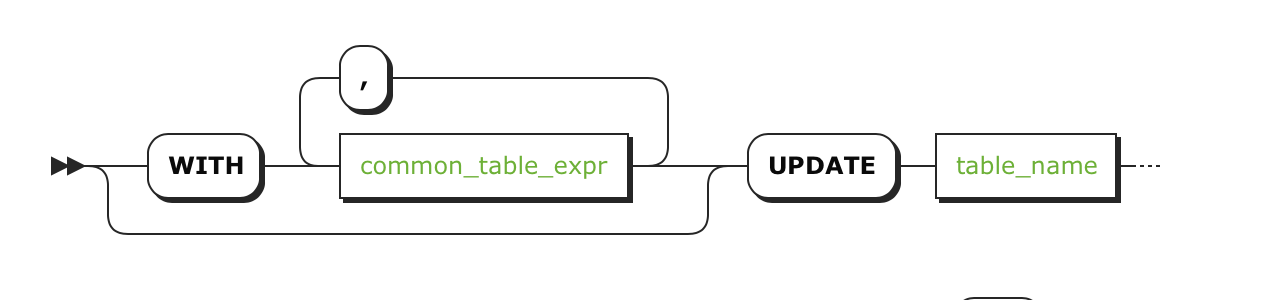
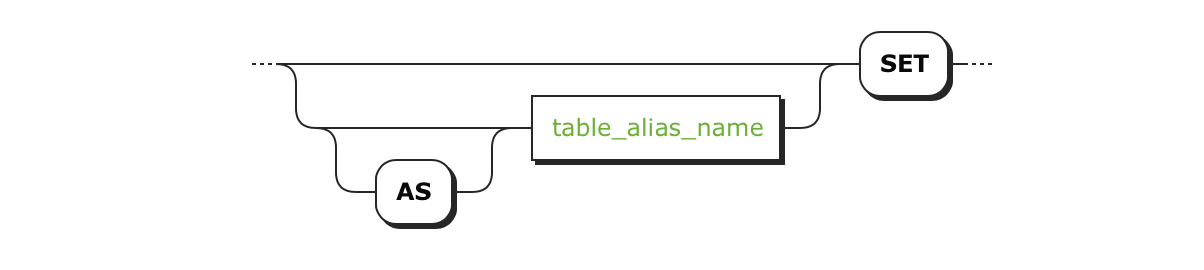
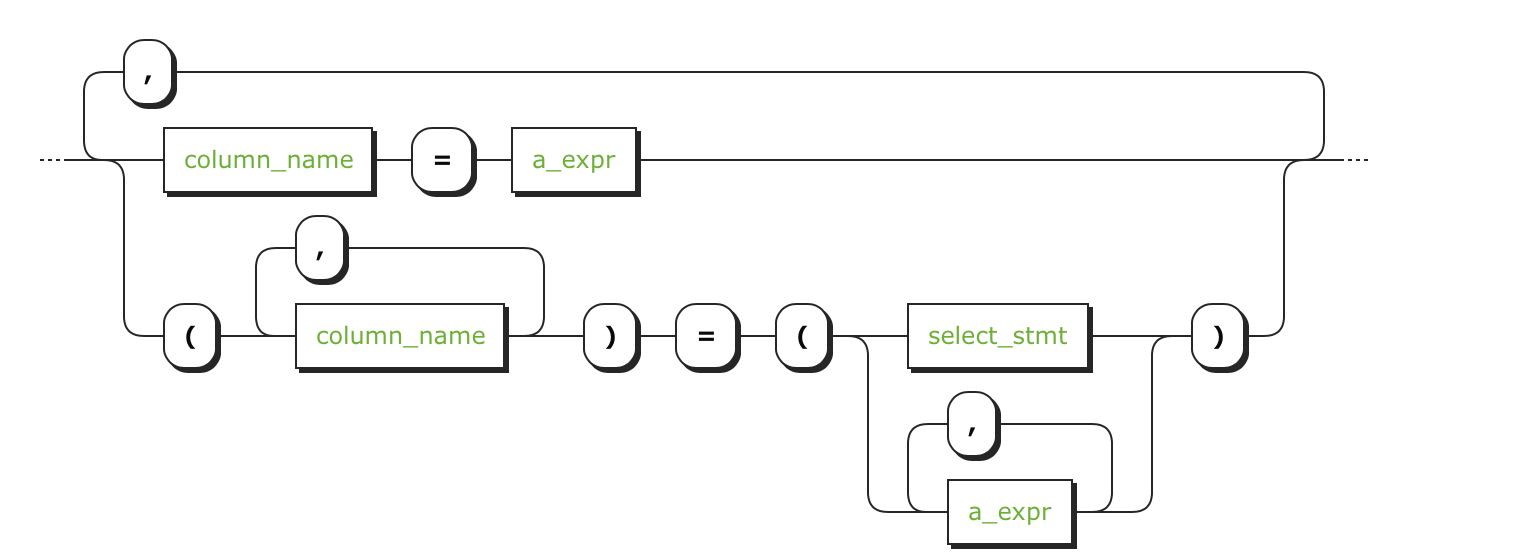
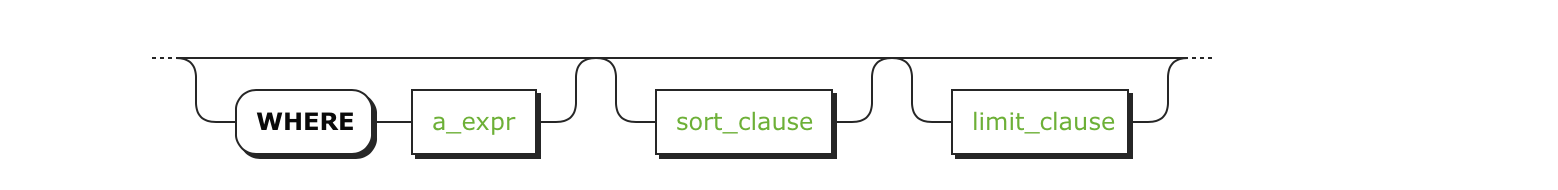
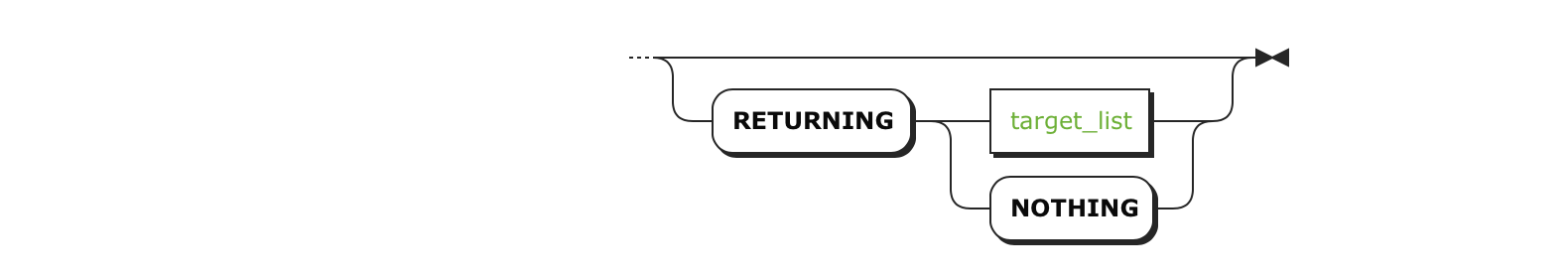
UPDATE [语句](http://doc.cockroachchina.baidu.com/#develop/sql-statements/overview/) 更新表中某行数据

Warning 如果更新的某行数据有外键依赖[foreign key constraint](http://doc.cockroachchina.baidu.com/#develop/sql-statements/UPDATE/foreign-key.html) ，而且有ON UPDATE action，那么所有依赖行都会更新。

所需权限

用户需要相关表的 SELECT 和 UPDATE [权限](http://doc.cockroachchina.baidu.com/#deploy/access-management/privileges/) 。

概要

参数

| **Parameter** | **Description** |
| --- | --- |
| common\_table\_expr | 请看[Common Table Expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/common-table-expressions/). |
| table\_name | 包含要更新行的表名 |
| AS table\_alias\_name | 表名的一个别名。当提供别名后，会完全隐藏真实的表名 |
| column\_name | 想要更新列的名字 |
| a\_expr | 想要使用的一个新值，想要执行的聚合函数[aggregate function](http://doc.cockroachchina.baidu.com/#develop/functions-and-operators/#aggregate-functions)，或者想要使用的[scalar expression](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/scalar-expressions/)。 |
| DEFAULT | 为了使用 [default values](http://doc.cockroachchina.baidu.com/#develop/constraints/default-value/)填充列，使用DEFAULT VALUES代替a\_expr。为了使用默认值填充特定列，在该值保留在a\_expr之外，或者在合适的位置使用DEFAULT。 |
| column\_name | 要更新的列名字 |
| select\_stmt | 一个[selection query](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/selection-queries/)。该查询返回的值的 [data type](http://doc.cockroachchina.baidu.com/#develop/data-types/overview/)必须与 =左边的列匹配。 |
| WHERE a\_expr | a\_expr 必须是一个使用列条件的[scalar expression](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/scalar-expressions/)(例如 <column> = <value>)。更新行的语句返回TRUE。  **如果语句中没有 WHERE 子句， UPDATE updates 会更新所有行的数据** |
| sort\_clause | 一个 ORDER BY 子句。更多相关信息，请看 See [Ordering Query Results](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/ordering-query-results/)。 |
| limit\_clause | 一个LIMIT子句。更多相关信息，请看[Limiting Query Results](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/limit-query-results/)。 |
| RETURNING target\_list | 基于更新行的返回值，在target\_list中可以是表中具体的列名，其中，\* 表示所有列，也可以是一个使用[scalar expressions](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/scalar-expressions/)的条件。  如果不想返回任何东西，甚至不返回更新的行数，可以使用 RETURNING NOTHING。 |

例子

更新一行中的某一列

> **SELECT** \* **FROM** accounts;

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

| id | balance | customer |

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

| 1 | 10000.50 | Ilya |

| 2 | 4000.0 | Julian |

| 3 | 8700.0 | Dario |

| 4 | 3400.0 | Nitin |

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

(4 rows)

> **UPDATE** accounts **SET** balance = 5000.0 **WHERE** id = 2;

> **SELECT** \* **FROM** accounts;

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

| id | balance | customer |

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

| 1 | 10000.50 | Ilya |

| 2 | 5000.0 | Julian |

| 3 | 8700.0 | Dario |

| 4 | 3400.0 | Nitin |

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

(4 rows)

更新一行中的多个列的值

> **UPDATE** accounts **SET** (balance, customer) = (9000.0, 'Kelly') **WHERE** id = 2;

> **SELECT** \* **FROM** accounts;

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

| id | balance | customer |

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

| 1 | 10000.50 | Ilya |

| 2 | 9000.0 | Kelly |

| 3 | 8700.0 | Dario |

| 4 | 3400.0 | Nitin |

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

(4 rows)

> **UPDATE** accounts **SET** balance = 6300.0, customer = 'Stanley' **WHERE** id = 3;

> **SELECT** \* **FROM** accounts;

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

| id | balance | customer |

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

| 1 | 10000.50 | Ilya |

| 2 | 9000.0 | Kelly |

| 3 | 6300.0 | Stanley |

| 4 | 3400.0 | Nitin |

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

(4 rows)

使用 SELECT 语句进行更新

> **UPDATE** accounts **SET** (balance, customer) =

(**SELECT** balance, customer **FROM** accounts **WHERE** id = 2)

**WHERE** id = 4;

> **SELECT** \* **FROM** accounts;

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

| id | balance | customer |

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

| 1 | 10000.50 | Ilya |

| 2 | 9000.0 | Kelly |

| 3 | 6300.0 | Stanley |

| 4 | 9000.0 | Kelly |

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

(4 rows)

使用默认值进行更新

> **UPDATE** accounts **SET** balance = **DEFAULT** **where** customer = 'Stanley';

> **SELECT** \* **FROM** accounts;

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

| id | balance | customer |

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

| 1 | 10000.50 | Ilya |

| 2 | 9000.0 | Kelly |

| 3 | NULL | Stanley |

| 4 | 9000.0 | Kelly |

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

(4 rows)

更新所有列

Warning: 如果没有使用 WHERE 子句去指定了要更新的行，会更新所有行的数据。

> **UPDATE** accounts **SET** balance = 5000.0;

> **SELECT** \* **FROM** accounts;

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

| id | balance | customer |

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

| 1 | 5000.0 | Ilya |

| 2 | 5000.0 | Kelly |

| 3 | 5000.0 | Stanley |

| 4 | 5000.0 | Kelly |

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

(4 rows)

更新和返回一些值

在这个例子中，使用RETURNING返回更新行的id值，假设已经安装了相关语言的对应的 [client drivers](http://doc.cockroachchina.baidu.com/#develop/install-client-drivers/)。

Tips: RETURNING 的使用反映了MYSQL的code>last\_insert\_id()函数的行为。

**Shell语言**

> **UPDATE** accounts **SET** balance = **DEFAULT** **WHERE** id = 1 RETURNING id;

+----+

| id |

+----+

| 1 |

+----+

(1 row)

**Python语言**

*# Import the driver.*

**import** psycopg2

*# Connect to the "bank" database.*

conn = psycopg2.connect(

database='bank',

user='root',

host='localhost',

port=26257

)

*# Make each statement commit immediately.*

conn.set\_session(autocommit=**True**)

*# Open a cursor to perform database operations.*

cur = conn.cursor()

*# Update a row in the "accounts" table*

*# and return the "id" value.*

cur.execute(

'UPDATE accounts SET balance = DEFAULT WHERE id = 1 RETURNING id'

)

*# Print out the returned value.*

rows = cur.fetchall()

print('ID:')

**for** row **in** rows:

print([str(cell) **for** cell **in** row])

*# Close the database connection.*

cur.close()

conn.close()

输出值：

ID:

['1']

**Ruby语言**

*# Import the driver.*

**require** 'pg'

*# Connect to the "bank" database.*

conn = PG.connect(

user: 'root',

dbname: 'bank',

host: 'localhost',

port: 26257

)

*# Update a row in the "accounts" table*

*# and return the "id" value.*

conn.exec(

'UPDATE accounts SET balance = DEFAULT WHERE id = 1 RETURNING id'

) **do** |res|

*# Print out the returned value.*

puts "ID:"

res.each **do** |row|

puts row

**end**

**end**

*# Close communication with the database.*

conn.close()

输出值

**ID**:

{"id"=>"1"}

**Go语言**

**package** main

**import** (

"database/sql"

"fmt"

"log"

\_ "github.com/lib/pq"

)

**func** main() {

*//Connect to the "bank" database.*

db, err := sql.Open(

"postgres",

"postgresql://root@localhost:26257/bank?sslmode=disable"

)

**if** err != nil {

log.Fatal("error connecting to the database: ", err)

}

*// Update a row in the "accounts" table*

*// and return the "id" value.*

rows, err := db.Query(

"UPDATE accounts SET balance = DEFAULT WHERE id = 1 RETURNING id",

)

**if** err != nil {

log.Fatal(err)

}

*// Print out the returned value.*

**defer** rows.Close()

fmt.Println("ID:")

**for** rows.Next() {

**var** id int

**if** err := rows.Scan(&id); err != nil {

log.Fatal(err)

}

fmt.Printf("%d\n", id)

}

}

输出值：

ID:

1

**Node.js语言**

**var** async = require('async');

*// Require the driver.*

**var** pg = require('pg');

*// Connect to the "bank" database.*

**var** config = {

user: 'root',

host: 'localhost',

database: 'bank',

port: 26257

};

pg.connect(config, **function** (err, client, done) {

*// Closes communication with the database and exits.*

**var** finish = **function** () {

done();

process.exit();

};

**if** (err) {

console.error('could not connect to cockroachdb', err);

finish();

}

async.waterfall([

**function** (next) {

*// Update a row in the "accounts" table*

*// and return the "id" value.*

client.query(

`UPDATE accounts SET balance = DEFAULT WHERE id = 1 RETURNING id`,

next

);

}

],

**function** (err, results) {

**if** (err) {

console.error('error updating and selecting from accounts', err);

finish();

}

*// Print out the returned value.*

console.log('ID:');

results.rows.forEach(**function** (row) {

console.log(row);

});

finish();

});

});

输出值：

ID:

{ id: '1' }

See Also

* [DELETE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DELETE/)
* [INSERT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/INSERT/)
* [UPSERT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/UPSERT/)
* [TRUNCATE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/TRUNCATE/)
* [ALTER TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALTER-TABLE/)
* [DROP TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-TABLE/)
* [DROP DATABASE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-DATABASE/)
* [Other SQL Statements](http://doc.cockroachchina.baidu.com/#develop/sql-statements/overview/)
* [Limiting Query Results](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/limit-query-results/)