



Khoa Công Nghệ Thông Tin Trường Đại Học Cần Thơ

Lệnh SQL cơ bản PostgreSQL



Đỗ Thanh Nghi
dtngchi@cit.ctu.edu.vn

Cần Thơ
24-04-2005

Nội dung

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

-
- **Tạo bảng**
 - Xen dữ liệu
 - Xóa dữ liệu
 - Cập nhật dữ liệu
 - Truy vấn dữ liệu

- **Tạo bảng**
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Tạo bảng

- **Cú pháp:**

```
CREATE [ TEMPORARY | TEMP ] TABLE table_name (  
    { column_name type [ column_constraint [...] ] |  
      table_constraint }  
    [, ...]  
)  
[ INHERITS ( inherited_table [,...] ) ]
```

Kiểu dữ liệu

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

■ Các kiểu thông dụng

- Số: integer, smallint, bigint, real, float8, numeric, serial
- Chuỗi ký tự: char, varchar, text
- Luận lý: boolean
- Ngày giờ: date, time, timestamp
- Nhị phân: bytea
- Tiền tệ: money
- Mảng: array
- Chuỗi bit: bit, varbit
- Địa chỉ mạng: inet, cidr, macaddr
- Hình học: path, point, line, polygon, box, circle, lseg
- Object Identifiers: oid, regclass, regtype, regproc, regoper
- Composite: dùng create type để tạo dữ liệu cấu trúc
- Pseudo: any, opaque, trigger, void, record, language_handler, etc

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ràng buộc trên column

■ column_constraint ::=

```

[ CONSTRAINT column_constraint_name ]
{ NOT NULL | NULL | UNIQUE | PRIMARY KEY |
  DEFAULT default_value |
  CHECK (condition |
  REFERENCES foreign_table [ ( foreign_column ) ]
    [ MATCH FULL | MATCH PARTIAL ]
    [ ON DELETE action ]
    [ ON UPDATE action ]
    [ DEFERRABLE | NOT DEFERRABLE ]
    [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]
  }

```

action ::= { NO ACTION | RESTRICT | CASCADE | SET
 NULL | SET DEFAULT }

Ràng buộc trên bảng

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

■ table constraint ::=

```
[ CONSTRAINT table_constraint_name ]  
{ UNIQUE ( column_name [, ... ] ) |  
  PRIMARY KEY ( column_name [, ... ] ) |  
  CHECK ( condition ) |  
  FOREIGN KEY ( column_name [, ... ] )  
    REFERENCES foreign_table  
    [ ( foreign_column [, ... ] ) ]  
    [ MATCH FULL | MATCH PARTIAL ]  
    [ ON DELETE action ]  
    [ ON UPDATE action ]  
    [ DEFERRABLE | NOT DEFERRABLE ]  
    [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]  
}
```

- **Tạo bảng**
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 1

- CREATE TABLE s (
 sid varchar(3) PRIMARY KEY,
 sname text NOT NULL,
 status smallint,
 city text);
- CREATE TABLE p (
 pid varchar(3) UNIQUE NOT NULL,
 pname text NOT NULL,
 color text,
 weight smallint,
 city text);

- **Tạo bảng**
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 2

- ```
CREATE TABLE sp (
 sid varchar(3),
 pid varchar(3),
 qty integer NOT NULL CHECK (qty > 0),
 CONSTRAINT sp_pkey PRIMARY KEY (sid, pid));
```
- ```
CREATE TABLE films (  
    code char(5) PRIMARY KEY,  
    title text NOT NULL,  
    date_prod date DEFAULT current_date);
```

- **Tạo bảng**
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 3

- ```
CREATE TABLE favorite_books (
 id varchar(5),
 books text[]);
```
- ```
CREATE TABLE sal_emp (  
    name text,  
    pay_by_quarter integer[],  
    schedule text[][]);
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 4

- CREATE TABLE fruit (
 name char(30),
 image OID);
- CREATE TABLE motos (
 id serial,
 owner text);
- CREATE TABLE authors (
 id char(4) PRIMARY KEY,
 name varchar(30),
 sex boolean DEFAULT false);

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Chỉnh sửa bảng

- Cú pháp:

ALTER TABLE table [*]

ADD [COLUMN] column type

ALTER TABLE table [*]

ALTER [COLUMN] column { SET DEFAULT
defaultvalue | DROP DEFAULT }

ALTER TABLE table [*]

RENAME [COLUMN] column TO newcolumn

ALTER TABLE table

RENAME TO newtable

ALTER TABLE table

ADD CONSTRAINT newconstraint definition

ALTER TABLE table

OWNER TO newowner

- **Tạo bảng**
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 5

- ALTER TABLE authors ADD COLUMN address text;
- ALTER TABLE motos RENAME COLUMN id TO no;
- ALTER TABLE authors DROP COLUMN address;
- ALTER TABLE motos ALTER COLUMN owner TYPE varchar(30);

-
- Tạo bảng
 - **Xen dữ liệu**
 - Xóa dữ liệu
 - Cập nhật dữ liệu
 - Truy vấn dữ liệu

- Tạo bảng
- **Xen dữ liệu**
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Xen dữ liệu vào bảng

- Cú pháp:

```
INSERT INTO table [ ( column [, ...] ) ]  
{ VALUES ( expression [, ...] ) | SELECT query }
```

Ví dụ 6

- Xen dữ liệu vào bảng s:

INSERT INTO s VALUES ('S01', 'Smith', 20, 'London');

INSERT INTO s VALUES ('S02', 'Jones', 10, 'Paris');

INSERT INTO s VALUES ('S03', 'Blacke', 30, 'Paris');

- Xen dữ liệu vào bảng p:

INSERT INTO p VALUES ('P01', 'Nut', 'red', 12, 'London');

INSERT INTO p VALUES ('P02', 'Bolt', 'green', 17, 'Paris');

INSERT INTO p VALUES ('P03', 'Screw', 'blue', 17, 'Roma');

INSERT INTO p VALUES ('P04', 'Screw', 'red', 14, 'London');

- Tạo bảng
- **Xen dữ liệu**
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 7

- Xen dữ liệu vào bảng sp:

```
INSERT INTO sp VALUES ('S01', 'P01', 300);  
INSERT INTO sp VALUES ('S01', 'P02', 200);  
INSERT INTO sp VALUES ('S01', 'P03', 400);  
INSERT INTO sp VALUES ('S02', 'P01', 300);  
INSERT INTO sp VALUES ('S02', 'P02', 400);  
INSERT INTO sp VALUES ('S03', 'P02', 200);
```

-
- Tạo bảng
 - Xen dữ liệu
 - **Xóa dữ liệu**
 - Cập nhật dữ liệu
 - Truy vấn dữ liệu

- Tạo bảng
- Xen dữ liệu
- **Xóa dữ liệu**
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Xóa dữ liệu từ bảng

- Cú pháp:

DELETE FROM table [WHERE condition]

- Ví dụ:

DELETE FROM films;

DELETE FROM films WHERE code = '00013';

-
- Tạo bảng
 - Xen dữ liệu
 - Xóa dữ liệu
 - **Cập nhật dữ liệu**
 - Truy vấn dữ liệu

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Cập nhật dữ liệu của bảng

■ Cú pháp:

UPDATE table

SET col = expression [, ...]

[FROM fromlist]

[WHERE condition]

■ Ví dụ:

UPDATE favorite_books

SET books = ‘{“databases”, “data mining”, “machine learning”}’

WHERE id = ‘00013’;

-
- Tạo bảng
 - Xen dữ liệu
 - Xóa dữ liệu
 - Cập nhật dữ liệu
 - **Truy vấn dữ liệu**

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Truy vấn dữ liệu

■ Cú pháp:

```
SELECT [ ALL | DISTINCT [ ON ( distinct_expression [, ...] ) ] ]  
    target_expression [ AS output_name ] [, ...]  
[ FROM from_item [ { , | CROSS JOIN } ... ] ]  
[ WHERE condition ]  
[ GROUP BY aggregate_expression [, ...] ]  
[ HAVING aggregate_condition [, ...] ]  
[ { UNION | INTERSECT | EXCEPT [ALL] } select ]  
[ ORDER BY order_expression [ ASC | DESC | USING  
operator ] [, ...] ]  
[ FOR UPDATE [ OF update_table [, ...] ] ]  
[ LIMIT { ALL | count } [ { OFFSET | , } start ] ]
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Mệnh đề FROM

- from_item ::=
 - { [ONLY] table_name [*]
 - [[AS] from_alias [(column_alias_list)]] |
 - (select) [[AS] alias [(column_alias_list)]] |
 - from_item [NATURAL] join_type from_item
 - [ON (join_condition) | USING (join_column_list)]
 - }

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Kiểu kết nối (JOIN)

- `join_type ::=`
[INNER |
LEFT [OUTER] |
RIGHT [OUTER] |
FULL [OUTER]
] JOIN

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**

Ví dụ 8

```
mydb=# SELECT 3*3 AS nb;
```

```
nb
```

```
----
```

```
9
```

```
(1 row)
```

```
mydb=# SELECT random(), 10-2 AS nb;
```

```
random          | nb
```

```
-----+-----
```

```
0.106375307359907 | 8
```

```
(1 row)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**

Ví dụ 9

```
mydb=# SELECT *
mydb=# FROM s;
  sid |  sname   | status |  city
-----+-----+-----+-----
  S01 | Smith    |      20 | London
  S02 | Jones    |      10 | Paris
  S03 | Blacke   |      30 | Paris
(3 rows)
```

```
mydb=# SELECT *
mydb=# FROM p;
  pid |  pname   | color | weight |  city
-----+-----+-----+-----+-----
  P01 | Nut      | red   |      12 | London
  P02 | Bolt     | green |      17 | Paris
  P03 | Screw    | blue  |      17 | Roma
  P04 | Screw    | red   |      14 | London
(4 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 10

```
mydb=# SELECT *
mydb=# FROM sp;
  sid | pid | qty
-----+-----+-----
  S01 | P01 | 300
  S01 | P02 | 200
  S01 | P03 | 400
  S02 | P01 | 300
  S02 | P02 | 400
  S03 | P02 | 200
(6 rows)
```

```
mydb=# SELECT *
mydb=# FROM sp
mydb=# LIMIT 3;
  sid | pid | qty
-----+-----+-----
  S01 | P01 | 300
  S01 | P02 | 200
  S01 | P03 | 400
(3 rows)
```

```
mydb=# SELECT *
mydb=# FROM sp
mydb=# OFFSET 2
mydb=# LIMIT 3;
  sid | pid | qty
-----+-----+-----
  S01 | P03 | 400
  S02 | P01 | 300
  S02 | P02 | 400
```

```
mydb=# SELECT sid
mydb=# FROM s
mydb=# WHERE city='Paris';
  sid
-----
  S02
  S03
(2 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**

Ví dụ 11

```
mydb=# SELECT *
mydb=# FROM s
mydb=# WHERE city LIKE 'P%';
  sid | sname   | status | city
-----+-----+-----+-----
  S02 | Jones   |      10 | Paris
  S03 | Blacke  |      30 | Paris
(2 rows)
```

```
mydb=# SELECT DISTINCT pid
mydb=# FROM sp;
  pid
-----
  P01
  P02
  P03
(3 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**

Ví dụ 12

```
mydb=# SELECT sid
mydb=# FROM s
mydb=# WHERE city='Paris' and status > 20;
   sid
-----
   S03
(1 row)
```

```
mydb=# SELECT sid, status
mydb=# FROM s
mydb=# WHERE city='Paris' ORDER BY status DESC;
   sid | status
-----+-----
   S03 |     30
   S02 |     10
(2 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**

Ví dụ 13

```
mydb=# SELECT sname
mydb-# FROM s
mydb-# WHERE sid IN (SELECT sid
mydb (#          FROM sp
mydb (#          WHERE pid='P02');
      sname
```

```
-----
Blacke
Jones
Smith
(3 rows)
```

```
mydb=# SELECT sname
mydb-# FROM s
mydb-# WHERE 'P01' NOT IN (SELECT pid
mydb (#          FROM sp
mydb (#          WHERE sid=s.sid);
      sname
```

```
-----
Blacke
(1 row)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 14

```
mydb=# SELECT pid
mydb=# FROM sp
mydb=# WHERE sid='S01'
mydb=# INTERSECT
mydb=# SELECT pid
mydb=# FROM p
mydb=# WHERE weight>13;
pid
```

```
P02
P03
(2 rows)

mydb=# SELECT pid
mydb=# FROM sp
mydb=# WHERE sid='S01'
mydb=# EXCEPT
mydb=# SELECT pid
mydb=# FROM p
mydb=# WHERE weight>13;
pid
-----
P01
(1 row)
```

```
mydb=# SELECT pid
mydb=# FROM sp
mydb=# WHERE sid='S01'
mydb=# UNION
mydb=# SELECT pid
mydb=# FROM p
mydb=# WHERE weight>13;
pid
```

```
P01
P02
P03
P04
(4 rows)
```


- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**

Ví dụ 15

```
mydb=# SELECT count(sid)
mydb-# FROM sp
mydb-# WHERE pid='P02';
      count
-----
         3
(1 row)
```

```
mydb=# SELECT SUM(qty)
mydb-# FROM sp
mydb-# WHERE pid='P02';
      sum
-----
      800
(1 row)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 16

```
mydb=# SELECT pid, count(sid)
mydb=# FROM sp
mydb=# GROUP BY pid;
 pid | count
-----+-----
 P01 |      2
 P02 |      3
 P03 |      1
(3 rows)
```

```
mydb=# SELECT pid, SUM(qty)
mydb=# FROM sp
mydb=# GROUP BY pid HAVING pid <> 'P02';
 pid | sum
-----+-----
 P01 | 600
 P03 | 400
(2 rows)
```

```
mydb=# SELECT max(qty), avg(qty)
mydb=# FROM sp;
 max |          avg
-----+-----
 400 | 300.000000000000000000
(1 row)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 17

```
mydb=# SELECT *
mydb=# FROM t1;
 id | name
----+-----
  1 | Peter
  2 | Joe
  3 | Bob
(3 rows)
```

```
mydb=# SELECT *
mydb=# FROM t2;
 id | address
----+-----
  2 | 53000-LAVAL
  5 | 75000-PARIS
  3 | 44000-NANTES
(3 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**

Ví dụ 18

```
mydb=# SELECT *
mydb=# FROM t1 CROSS JOIN t2;
 id | name  | id | address
-----+-----+-----+-----
  1 | Peter |  2 | 53000-LAVAL
  1 | Peter |  5 | 75000-PARIS
  1 | Peter |  3 | 44000-NANTES
  2 | Joe   |  2 | 53000-LAVAL
  2 | Joe   |  5 | 75000-PARIS
  2 | Joe   |  3 | 44000-NANTES
  3 | Bob   |  2 | 53000-LAVAL
  3 | Bob   |  5 | 75000-PARIS
  3 | Bob   |  3 | 44000-NANTES
(9 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 19

```
mydb=# SELECT *
mydb=# FROM t1 INNER JOIN t2 ON t1.id = t2.id;
 id | name | id | address
----+-----+----+-----
  2 | Joe  |  2 | 53000-LAVAL
  3 | Bob  |  3 | 44000-NANTES
(2 rows)
```

```
mydb=# SELECT *
mydb=# FROM t1 NATURAL INNER JOIN t2;
 id | name | address
----+-----+-----
  2 | Joe  | 53000-LAVAL
  3 | Bob  | 44000-NANTES
(2 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**

Ví dụ 20

```
mydb=# SELECT *
mydb=# FROM t1 LEFT JOIN t2 ON t1.id = t2.id;
 id | name  | id | address
-----+-----+-----+-----
  1 | Peter |   | 
  2 | Joe   |  2 | 53000-LAVAL
  3 | Bob   |  3 | 44000-NANTES
(3 rows)
```

```
mydb=#
mydb=# SELECT *
mydb=# FROM t1 RIGHT JOIN t2 ON t1.id = t2.id;
 id | name  | id | address
-----+-----+-----+-----
  2 | Joe   |  2 | 53000-LAVAL
    |       |  5 | 75000-PARIS
  3 | Bob   |  3 | 44000-NANTES
(3 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 21

```
mydb=# SELECT *
mydb=# FROM t1 FULL JOIN t2 ON t1.id = t2.id;
 id | name  | id | address
-----+-----+-----+-----
  1 | Peter |    |
  2 | Joe   |  2 | 53000-LAVAL
  3 | Bob   |  3 | 44000-NANTES
    |       |  5 | 75000-PARIS
(4 rows)
```

```
mydb=# SELECT *
mydb=# FROM t1 LEFT JOIN t2 ON t1.id = t2.id
mydb=# AND t2.address = '53000-LAVAL';
 id | name  | id | address
-----+-----+-----+-----
  1 | Peter |    |
  2 | Joe   |  2 | 53000-LAVAL
  3 | Bob   |    |
(3 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Tạo chỉ mục

- Cú pháp:

```
CREATE [ UNIQUE ] INDEX index_name ON table  
    [ USING method ] ( column [ op_class ] [, ...] )
```

```
CREATE [ UNIQUE ] INDEX index_name ON table  
    [ USING method ] ( func_name ( column [, ... ] ) [ op_class ] )
```

- B-Tree, Hash, R-Tree, GiST

- Tăng tốc cho phép truy vấn

Ví dụ 22

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- **Truy vấn dữ liệu**

```
mydb=# EXPLAIN ANALYZE
mydb=# SELECT *
mydb=# FROM sp
mydb=# WHERE pid='P03';
```

QUERY PLAN

```
-----
Seq Scan on sp  (cost=0.00..29.62 rows=8 width=18)
(actual time=0.069..0.076 rows=1 loops=1)
Filter: ((pid)::text = 'P03'::text)
Total runtime: 0.171 ms
(3 rows)
```

```
mydb=# CREATE INDEX pid_index
mydb=# ON sp(pid);
CREATE INDEX
```

```
mydb=#
mydb=# EXPLAIN ANALYZE
mydb=# SELECT *
mydb=# FROM sp
mydb=# WHERE pid='P03';
```

QUERY PLAN

```
-----
Seq Scan on sp  (cost=0.00..1.07 rows=1 width=18)
(actual time=0.033..0.040 rows=1 loops=1)
Filter: ((pid)::text = 'P03'::text)
Total runtime: 0.114 ms
(3 rows)
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 23

■ Dữ liệu binary:

```
CREATE TABLE fruit (  
    name char(30),  
    image OID);
```

```
INSERT INTO fruit VALUES ('peach', lo_import('/usr/images/peach.jpg'));
```

```
SELECT lo_export(fruit.image, '/tmp/outimage.jpg') FROM fruit  
WHERE name = 'peach';
```

```
SELECT lo_unlink(fruit.image) FROM fruit;
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 24

■ Dữ liệu mảng:

```
CREATE TABLE array_test (  
    col1 INTEGER[5],  
    col2 INTEGER[][],  
    col3 INTEGER[2][2][]);  
INSERT INTO array_test VALUES (  
    '{1,2,3,4,5}',  
    '{{1,2},{3,4}}',  
    '{{{{1,2},{3,4}}},{{5,6},{7,8}}}' );  
SELECT * FROM array_test;  
SELECT col1[4] FROM array_test;  
SELECT col2[2][1] FROM array_test;  
SELECT col3[1][2][2] FROM array_test;
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 25

■ Câu truy vấn đệ quy: (suy diễn)

```
CREATE TABLE parent (  
    f varchar(10),  
    c varchar(10));
```

f là cha của c hay

c là con của f

```
INSERT INTO parent VALUES('Paul', 'Henri');  
INSERT INTO parent VALUES('Jean', 'Marie');  
INSERT INTO parent VALUES('Louis', 'Paul');  
INSERT INTO parent VALUES('Katie', 'Louis');  
INSERT INTO parent VALUES('Nicole', 'Jean');
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 25

- Câu truy vấn đệ quy: (suy diễn) tìm tổ tiên của **Paul**

```
WITH RECURSIVE anc(An, De) AS
(
  SELECT f, c FROM parent
  UNION ALL
  SELECT a.An, p.c
  FROM anc a, parent p
  WHERE a.De = p.f
)
SELECT * FROM anc WHERE De = 'Paul';
```

- Tạo bảng
- Xen dữ liệu
- Xóa dữ liệu
- Cập nhật dữ liệu
- Truy vấn dữ liệu

Ví dụ 26

- Câu truy vấn đệ quy: tính tổng dãy số nguyên 1, ..., 100

```
WITH RECURSIVE t(n) AS
(
  SELECT 1
  UNION ALL
  SELECT n+1
  FROM t
  WHERE n < 100
)
SELECT sum(n) FROM t;
```

Ví dụ 27

- Câu truy vấn đệ quy: tính giai thừa của 5

```
WITH RECURSIVE gt(n, r) AS
(
  SELECT 1, 1
  UNION ALL
  SELECT n+1, (n+1)*r
  FROM gt
)
SELECT r FROM gt OFFSET 4 LIMIT 1;
```



Cám ơn !