In [1]:

```
import sqlite3
```

In [ ]:

```
User -> DBMS -> Database
    Connection: DBMS 접속
    Cursor: 작업
1. Connection -> 2. Cursor
```

In [11]:

```
con = sqlite3.connect(':memory:')
```

In [12]:

```
cur = con.cursor()
```

In [8]:

```
type(con), type(cur), type(con.cursor)
```

Out[8]:

```
(sqlite3.Connection, sqlite3.Cursor, builtin_function_or_method)
```

In [10]:

```
con.close()
```

In [ ]:

```
cur.execute(), cur.executemany(), cur.executescript()
```

In [13]:

```
cur.execute('''
CREATE TABLE CITY(
    CNO   INTEGER PRIMARY KEY,
    CNAME TEXT
);
''')
```

Out[13]:

```
<sqlite3.Cursor at 0x11296b030>
```

In [ ]:

```
Client        DBMS          DB
   excute ->  해석    ----->
                     <----
  cur --> None, [결과,결과,결과]
```

In [14]:

```
cur.execute('SELECT * FROM CITY')
```

Out[14]:

```
<sqlite3.Cursor at 0x11296b030>
```

In [16]:

```
cur.execute('''
INSERT INTO CITY(CNO, CNAME) VALUES(1, '성북구')
''')
```

Out[16]:

```
<sqlite3.Cursor at 0x11296b030>
```

In [17]:

```
cur.execute('SELECT * FROM CITY')
```

Out[17]:

```
<sqlite3.Cursor at 0x11296b030>
```

In [18]:

```
cur.fetchall()
```

Out[18]:

```
[(1, '성북구')]
```

In [20]:

```
cur.execute('''
INSERT INTO CITY(CNO, CNAME) VALUES(NULL, '강북구')
''')
```

Out[20]:
```
<sqlite3.Cursor at 0x11296b030>
```

In [22]:
```
cur.execute('SELECT * FROM CITY')
cur.fetchall()
```

Out[22]:
```
[(1, '성북구'), (2, '강북구')]
```

In [24]:
```
cur.execute('''
INSERT INTO CITY(CNAME) VALUES('성동구')
''')
cur.execute('''
INSERT INTO CITY VALUES(NULL, '동대문구')
''')
# PK -> Unique, Not null, auto-increment
# Column -> Not null, -> Error
```

Out[24]:
```
<sqlite3.Cursor at 0x11296b030>
```

In [25]:
```
cur.execute('SELECT * FROM CITY')
cur.fetchall()
```

Out[25]:
```
[(1, '성북구'), (2, '강북구'), (3, '성동구'), (4, '성동구'), (5, '동대문구')]
```

In [26]:
```
cur.lastrowid
```

Out[26]:
```
5
```

In [32]:
```
cityList = ['성북구', '강북구', '동대문구', '중구', '도봉구']

for cname in cityList:
    cur.execute('''
    SELECT CNO FROM CITY WHERE CNAME=?
    ''', [cname,])

    if len(cur.fetchall()) == 0:
        cur.execute('''
        INSERT INTO CITY VALUES(NULL, :cname)
        ''', {'cname':cname})
        print('새로입력', cname, cur.lastrowid)
    else:
        print('이미있음', cname, cur.fetchone())
```

```
이미있음 성북구 None
이미있음 강북구 None
이미있음 동대문구 None
새로입력 중구 6
새로입력 도봉구 7
```

In [31]:
```
len(cur.fetchall()), cur.fetchall()
```

Out[31]:
```
(0, [])
```

In [33]:
```
con.close()
```

In [34]:
```
con = sqlite3.connect(':memory:')
cur = con.cursor()
```

In [35]:
```
cur.execute('''
CREATE TABLE CITY(
    CNO   INTEGER PRIMARY KEY,
    CNAME TEXT
)
''')
```

Out[35]: `<sqlite3.Cursor at 0x1129891f0>`

In [36]:

```
# 3가지 방법 -> SQL만 이용
cur.execute('''
INSERT INTO CITY(CNO, CNAME)
VALUES(1, '성북구')
''')

cur.execute('''
INSERT INTO CITY(CNAME)
VALUES('강북구')
''')

cur.execute('''
INSERT INTO CITY
VALUES(NULL, '강동구')
''')
```

Out[36]: `<sqlite3.Cursor at 0x1129891f0>`

In [39]:

```
# qmark
# SQL(?), Python(iterable객체)
# named style
# SQL(:key), Python(dict객체)
cur.execute('''
INSERT INTO CITY(CNAME) VALUES(?)
''', ('동대문구',))
cur.lastrowid
```

Out[39]: 4

In [40]:

```
cur.execute('''
INSERT INTO CITY(CNAME) VALUES(:key)
''', {'key':'중랑구'})
cur.lastrowid
```

Out[40]: 5

In [42]:

```
type(('문자')), type(('문자',)), ('문자'), ('문자',), len(('문자',))
```

Out[42]: `(str, tuple, '문자', ('문자',), 1)`

In [43]:

```
con.close()
```

In [44]:

```
con = sqlite3.connect(':memory:')
cur = con.cursor()
```

In [45]:

```
cur.execute('''
CREATE TABLE CITY(
    CNO   INTEGER PRIMARY KEY,
    CNAME TEXT
)
''')
```

```
Out[45]: <sqlite3.Cursor at 0x112a69a40>
```

```
In [46]: cur.executemany('''
         INSERT INTO CITY(CNAME) VALUES(?)
         ''', [('성북구1',), ('성북구2',), ('성북구3',), ('성북구4',)])
```

```
Out[46]: <sqlite3.Cursor at 0x112a69a40>
```

```
In [ ]:  execute        --> iterable 객체 (scheme, attribute N개)

         executemany    --> iterable(iterable)
```

```
In [47]: cur.execute('SELECt * FROM CITY')
         cur.fetchall()
```

```
Out[47]: [(1, '성북구1'), (2, '성북구2'), (3, '성북구3'), (4, '성북구4')]
```

```
In [48]: cur.executemany('''
         INSERT INTO CITY(CNAME) VALUES(?)
         ''', ('성북구5',))
```

```
---------------------------------------------------------------------------
ProgrammingError                          Traceback (most recent call last)
Cell In [48], line 1
----> 1 cur.executemany('''
      2 INSERT INTO CITY(CNAME) VALUES(?)
      3 ''', ('성북구5',))

ProgrammingError: Incorrect number of bindings supplied. The current statement uses 1, a
nd there are 4 supplied.
```

```
In [50]: cur.executescript('''
         INSERT INTO CITY(CNAME) VALUES('성북구5');
         INSERT INTO CITY(CNAME) VALUES('성북구6');
         INSERT INTO CITY(CNAME) VALUES('성북구7');
         INSERT INTO CITY(CNAME) VALUES('성북구8');
         ''')
```

```
Out[50]: <sqlite3.Cursor at 0x112a69a40>
```

```
In [51]: cur.execute('SELECT * FROM CITY')
         cur.fetchall()
```

```
Out[51]: [(1, '성북구1'),
          (2, '성북구2'),
          (3, '성북구3'),
          (4, '성북구4'),
          (5, '성북구5'),
          (6, '성북구6'),
          (7, '성북구7'),
          (8, '성북구8')]
```

SQL문 DBMS 일시키는 법 with Python 객체 Python -(execute*) => DBMS -> DB

1. execute; 단일 SQL 실행(DBMS->DB); Qmark, Named 파라미터(Python) -> 단일 데이터에 대한

Attributes

2. executemany; 단일 SQL 여러번 실행; -> 여러 데이터에 대한
3. executescript; 비표준, 여러 SQL 실행; 문(statement)의 끝에는 ;

In [ ]:

```
SQL문 해석한 결과 DBMS에서 가져오는 법
Python - (execute*) => DBMS -> DB
                      (데이터)
(저장)     (fetch*) <= 가져오는
fetchone   => 데이터 한개
fetchmany  => 데이터 N개
fetchall   => 전체 데이터
          (더이상 데이터가 없을때까지)
```

In [71]:

```
cur.execute('SELECT * FROM CITY')
```

Out[71]:

```
<sqlite3.Cursor at 0x112a69a40>
```

In [ ]:

```
cur.fetchone()
```

In [69]:

```
cur.fetchmany(3)
```

Out[69]:

```
[(4, '성북구4'), (5, '성북구5'), (6, '성북구6')]
```

In [70]:

```
cur.fetchall()
```

Out[70]:

```
[(7, '성북구7'), (8, '성북구8')]
```

In [72]:

```
con.close()
```

In [74]:

```
con = sqlite3.connect('fnb.db')
cur = con.cursor()
```

In [75]:

```
con.close()
```

In [77]:

```
con = sqlite3.connect('fnb.db')
cur = con.cursor()
```

In [ ]:

```
ALTER TABLE CITY DROP COLUMN CNO
```

In [79]:

```
#REFEREECES CITY(CNO)
cur.executescript('''
DROP TABLE IF EXISTS CITY;
CREATE TABLE CITY(
    CNO   INTEGER PRIMARY KEY,
    CNAME TEXT
);

CREATE TABLE SUPPLIER(
    SNO   INTEGER PRIMARY KEY,
    SNAME TEXT,
    CNO   INTEGER NOT NULL
);

CREATE TABLE PART(
    PNO   INTEGER PRIMARY KEY,
    PNAME TEXT
);

CREATE TABLE SELLS(
    SNO   INTEGER,
    PNO   INTEGER,
    PRICE INTEGER
);
''')
```

```
---------------------------------------------------------------------------
OperationalError                          Traceback (most recent call last)
Cell In [79], line 2
      1 #REFEREces CITY(CNO)
----> 2 cur.executescript('''
      3 CREATE TABLE CITY(
      4     CNO   INTEGER PRIMARY KEY,
      5     CNAME TEXT
      6 );
      7 
      8 CREATE TABLE SUPPLIER(
      9     SNO   INTEGER PRIMARY KEY,
     10     SNAME TEXT,
     11     CNO   INTEGER NOT NULL
     12 );
     13 
     14 CREATE TABLE PART(
     15     PNO   INTEGER PRIMARY KEY,
     16     PNAME TEXT
     17 );
     18 
     19 CREATE TABLE SELLS(
     20     SNO   INTEGER,
     21     PNO   INTEGER,
     22     PRICE INTEGER
     23 );
     24 ''')

OperationalError: table CITY already exists
```

In [86]:
```
cur.execute('INSERT INTO CITY(CNAME) VALUES("성북구")')
```

Out[86]:
```
<sqlite3.Cursor at 0x112b1c6c0>
```

In [87]:
```
cur.lastrowid
```

Out[87]:
```
1
```

In [82]:
```
cur.execute('SELECT * FROM CITY')
cur.fetchall()
```

Out[82]:
```
[(1, '성북구')]
```

In [83]:
```
con.close()
```

In [84]:
```
con = sqlite3.connect('fnb.db')
cur = con.cursor()
```

In [85]:

```
cur.execute('SELECT * FROM CITY')
cur.fetchall()
```

Out[85]:

```
[]
```

In [ ]:

```
cur.execute('INSERT INTO CITY(CNAME) VALUES("성북구")')
cur.lastrowid
```

In [88]:

```
# DBMS -> DB 반영
con.commit()
```

In [89]:

```
con.close()
```

In [90]:

```
con = sqlite3.connect('fnb.db')
cur = con.cursor()
```

In [91]:

```
cur.execute('SELECT * FROM CITY')
cur.fetchall()
```

Out[91]:

```
[(1, '성북구')]
```

In [92]:

```
cityList = [('성북구',), ('강북구',), ('강동구',), ('동대문구',)]

for city in cityList: # List -> Tuple(city)
    cur.execute('SELECT * FROM CITY WHERE CNAME=?', city)
    if len(cur.fetchall()) == 0:
        cur.execute('INSERT INTO CITY(CNAME) VALUES(?)', city)
```

In [93]:

```
cur.execute('SELECT * FROM CITY')
cur.fetchall()
```

Out[93]:

```
[(1, '성북구'), (2, '강북구'), (3, '강동구'), (4, '동대문구')]
```

In [94]:

```
con.commit()
```

In [95]:

```
supplierList = [('1호점',1), ('2호점',2), ('3호점',3), ('4호점',4),]
cur.executemany('INSERT INTO SUPPLIER(SNAME, CNO) VALUES(?,?)',
                supplierList)
```

Out[95]:

```
<sqlite3.Cursor at 0x112b1c7a0>
```

In [96]:

```
con.commit()
```

In [97]:

```
cur.execute('SELECT * FROM SUPPLIER')
cur.fetchall()
```

Out[97]:

```
[(1, '1호점', 1), (2, '2호점', 2), (3, '3호점', 3), (4, '4호점', 4)]
```

In [110]…

```
cur.execute('SELECT CNO FROM CITY WHERE CNAME=?', ('강북구',))
row = cur.fetchone() # 한 데이터 row(tuple, record)
CNO = 0
if row:
    CNO = row[0]
CNO
```

Out[110]:

```
2
```

In [111]…

```
cur.execute('INSERT INTO SUPPLIER(SNAME, CNO) VALUES(?,?)',
            ('6호점', CNO))
```

Out[111]:

```
<sqlite3.Cursor at 0x112b1c7a0>
```

In [112]…

```
cur.execute('SELECT * FROM SUPPLIER')
cur.fetchall()
```

Out[112]:

```
[(1, '1호점', 1),
 (2, '2호점', 2),
 (3, '3호점', 3),
 (4, '4호점', 4),
 (5, '5호점', 1),
 (6, '6호점', 2)]
```

In [113]…

```
cur.execute('''
    INSERT INTO SUPPLIER(SNAME, CNO) VALUES(?, (
        SELECT CNO FROM CITY WHERE CNAME=? LIMIT 0,1
    ))
''', ('7호점', '강동구'))
```

Out[113]:

```
<sqlite3.Cursor at 0x112b1c7a0>
```

In [114]…

```
cur.execute('SELECT * FROM SUPPLIER')
cur.fetchall()
```

Out[114]:

```
[(1, '1호점', 1),
 (2, '2호점', 2),
 (3, '3호점', 3),
 (4, '4호점', 4),
 (5, '5호점', 1),
 (6, '6호점', 2),
 (7, '7호점', 3)]
```

In [115]…

```
cur.execute('SELECT * FROM CITY WHERE CNAME LIKE ?', ('강%',))
cur.fetchall()
```

Out[115]:

```
[(2, '강북구'), (3, '강동구')]
```

In [117]…

```
# Cross Join X
cur.execute('''
    SELECT * FROM CITY, SUPPLIER WHERE CITY.CNO = SUPPLIER.CNO
''')
cur.fetchall()
```

Out[117]:

```
[(1, '성북구', 1, '1호점', 1),
 (2, '강북구', 2, '2호점', 2),
 (3, '강동구', 3, '3호점', 3),
 (4, '동대문구', 4, '4호점', 4),
 (1, '성북구', 5, '5호점', 1),
 (2, '강북구', 6, '6호점', 2),
 (3, '강동구', 7, '7호점', 3)]
```

In [118]…

```
cur.execute('''
    SELECT * FROM CITY
    INNER JOIN SUPPLIER
    ON SUPPLIER.CNO = CITY.CNO
''')
```

```
cur.fetchall()
```

Out[118]:

```
[(1, '성북구', 1, '1호점', 1),
 (2, '강북구', 2, '2호점', 2),
 (3, '강동구', 3, '3호점', 3),
 (4, '동대문구', 4, '4호점', 4),
 (1, '성북구', 5, '5호점', 1),
 (2, '강북구', 6, '6호점', 2),
 (3, '강동구', 7, '7호점', 3)]
```

In [119]…

```
cur.execute('INSERT INTO CITY(CNAME) VALUES("중구")')
```

Out[119]:

```
<sqlite3.Cursor at 0x112b1c7a0>
```

In [120]…

```
cur.execute('''
    SELECT * FROM CITY
    LEFT JOIN SUPPLIER
    ON SUPPLIER.CNO = CITY.CNO
''')
cur.fetchall()
```

Out[120]:

```
[(1, '성북구', 1, '1호점', 1),
 (1, '성북구', 5, '5호점', 1),
 (2, '강북구', 2, '2호점', 2),
 (2, '강북구', 6, '6호점', 2),
 (3, '강동구', 3, '3호점', 3),
 (3, '강동구', 7, '7호점', 3),
 (4, '동대문구', 4, '4호점', 4),
 (5, '중구', None, None, None)]
```

In [122]…

```
cur.execute('''
    SELECT * FROM SUPPLIER
    LEFT JOIN CITY
    ON SUPPLIER.CNO = CITY.CNO
''')
cur.fetchall()
```

Out[122]:

```
[(1, '1호점', 1, 1, '성북구'),
 (2, '2호점', 2, 2, '강북구'),
 (3, '3호점', 3, 3, '강동구'),
 (4, '4호점', 4, 4, '동대문구'),
 (5, '5호점', 1, 1, '성북구'),
 (6, '6호점', 2, 2, '강북구'),
 (7, '7호점', 3, 3, '강동구')]
```

In [124]…

```
cur.execute('''
    SELECT CITY.CNAME, COUNT(*) FROM CITY
    LEFT JOIN SUPPLIER
    ON SUPPLIER.CNO = CITY.CNO
    GROUP BY CITY.CNAME
''')
cur.fetchall()
```

Out[124]:

```
[('강동구', 2), ('강북구', 2), ('동대문구', 1), ('성북구', 2), ('중구', 1)]
```

In [125]…

```
cur.execute('''
    SELECT CITY.CNAME, COUNT(*) FROM SUPPLIER
    LEFT JOIN CITY
    ON SUPPLIER.CNO = CITY.CNO
    GROUP BY CITY.CNAME
''')
cur.fetchall()
```

Out[125]:

```
[('강동구', 2), ('강북구', 2), ('동대문구', 1), ('성북구', 2)]
```

In [126]…

```
partList = [('메뉴1',), ('메뉴2',), ('메뉴3',), ('메뉴4',)]
cur.executemany('INSERT INTO PART(PNAME) VALUES(?)', partList)
```

Out[126]:

```
<sqlite3.Cursor at 0x112b1c7a0>
```

In [127]…

```
sellsList = [
    {'sno':1,'pno':1,'price':10},
    {'sno':2,'pno':2,'price':8},
    {'sno':3,'pno':3,'price':11},
    {'sno':4,'pno':4,'price':20},
    {'sno':5,'pno':1,'price':1},
    {'sno':6,'pno':2,'price':40},
    {'sno':7,'pno':3,'price':50},
    {'sno':1,'pno':4,'price':10},
    {'sno':2,'pno':1,'price':14},
    {'sno':3,'pno':2,'price':8},
]
cur.executemany('INSERT INTO SELLS VALUES(:sno, :pno, :price)',
                sellsList)
```

Out[127]:

```
<sqlite3.Cursor at 0x112b1c7a0>
```

In [129]…

```
sellsList = [
    {'sno':'1%','pno':'%1','price':10},
    {'sno':'2%','pno':'%2','price':8},
    {'sno':'3%','pno':'%3','price':11},
    {'sno':'4%','pno':'%4','price':20},
    {'sno':'5%','pno':'%1','price':1},
    {'sno':'6%','pno':'%2','price':40},
    {'sno':'7%','pno':'%3','price':50},
    {'sno':'1%','pno':'%4','price':10},
    {'sno':'2%','pno':'%1','price':14},
    {'sno':'3%','pno':'%2','price':8},
]
cur.executemany('''
    INSERT INTO SELLS VALUES(
        (SELECT SNO FROM SUPPLIER WHERE SNAME LIKE :sno LIMIT 0,1),
        (SELECT PNO FROM PART WHERE PNAME LIKE :pno LIMIT 0,1),
        :price
    )
''', sellsList)
```

Out[129]:

```
<sqlite3.Cursor at 0x112b1c7a0>
```

In [130]…

```
con.commit()
```

In [131]…

```
cur.execute('SELECT * FROM SELLS')
cur.fetchall()
```

Out[131]:

```
[(1, 1, 10),
 (2, 2, 8),
 (3, 3, 11),
 (4, 4, 20),
 (5, 1, 1),
 (6, 2, 40),
 (7, 3, 50),
 (1, 4, 10),
 (2, 1, 14),
 (3, 2, 8),
 (1, 1, 10),
 (2, 2, 8),
 (3, 3, 11),
 (4, 4, 20),
 (5, 1, 1),
 (6, 2, 40),
 (7, 3, 50),
 (1, 4, 10),
 (2, 1, 14),
 (3, 2, 8)]
```

In [138]…

```
cur.execute('''
    SELECT CITY.CNAME, SUPPLIER.SNAME, PART.PNAME, SELLS.PRICE FROM SELLS
    INNER JOIN SUPPLIER
    ON SUPPLIER.SNO = SELLS.SNO
    INNER JOIN PART
    ON PART.PNO = SELLS.PNO
    INNER JOIN CITY
    ON CITY.CNO = SUPPLIER.CNO
    ORDER BY SELLS.PRICE DESC
    LIMIT 0,3
''')
cur.fetchall()
```

Out[138]:

```
[('강동구', '7호점', '메뉴3', 50),
 ('강동구', '7호점', '메뉴3', 50),
 ('강북구', '6호점', '메뉴2', 40)]
```

In [140]…

```
cur.execute('''
    SELECT CITY.CNAME, SUM(SELLS.PRICE) FROM SELLS
    INNER JOIN SUPPLIER
    ON SUPPLIER.SNO = SELLS.SNO
    INNER JOIN PART
    ON PART.PNO = SELLS.PNO
    INNER JOIN CITY
    ON CITY.CNO = SUPPLIER.CNO
    GROUP BY CITY.CNAME
''')
cur.fetchall()
```

Out[140]:

```
[('강동구', 138, 6), ('강북구', 124, 6), ('동대문구', 40, 2), ('성북구', 42, 6)]
```

In [142]…

```
cur.execute('''
    SELECT CITY.CNAME, SUPPLIER.SNAME, SUM(SELLS.PRICE) FROM SELLS
    INNER JOIN SUPPLIER
    ON SUPPLIER.SNO = SELLS.SNO
    INNER JOIN PART
    ON PART.PNO = SELLS.PNO
    INNER JOIN CITY
    ON CITY.CNO = SUPPLIER.CNO
    GROUP BY CITY.CNAME, SUPPLIER.SNAME
    ORDER BY SUPPLIER.SNAME ASC
''')
cur.fetchall()
```

Out[142]:

```
[('성북구', '1호점', 40),
 ('강북구', '2호점', 44),
 ('강동구', '3호점', 38),
 ('동대문구', '4호점', 40),
 ('성북구', '5호점', 2),
 ('강북구', '6호점', 80),
 ('강동구', '7호점', 100)]
```

In [144]:…

```
[_ for _ in con.iterdump()]
```

Out[144]:

```
['BEGIN TRANSACTION;',
 'CREATE TABLE CITY(\n    CNO   INTEGER PRIMARY KEY,\n    CNAME TEXT\n);',
 'INSERT INTO "CITY" VALUES(1,\'성북구\');',
 'INSERT INTO "CITY" VALUES(2,\'강북구\');',
 'INSERT INTO "CITY" VALUES(3,\'강동구\');',
 'INSERT INTO "CITY" VALUES(4,\'동대문구\');',
 'INSERT INTO "CITY" VALUES(5,\'중구\');',
 'CREATE TABLE PART(\n    PNO   INTEGER PRIMARY KEY,\n    PNAME TEXT\n);',
 'INSERT INTO "PART" VALUES(1,\'메뉴1\');',
 'INSERT INTO "PART" VALUES(2,\'메뉴2\');',
 'INSERT INTO "PART" VALUES(3,\'메뉴3\');',
 'INSERT INTO "PART" VALUES(4,\'메뉴4\');',
 'CREATE TABLE SELLS(\n    SNO   INTEGER,\n    PNO   INTEGER,\n    PRICE INTEGER\n);',
 'INSERT INTO "SELLS" VALUES(1,1,10);',
 'INSERT INTO "SELLS" VALUES(2,2,8);',
 'INSERT INTO "SELLS" VALUES(3,3,11);',
 'INSERT INTO "SELLS" VALUES(4,4,20);',
 'INSERT INTO "SELLS" VALUES(5,1,1);',
 'INSERT INTO "SELLS" VALUES(6,2,40);',
 'INSERT INTO "SELLS" VALUES(7,3,50);',
 'INSERT INTO "SELLS" VALUES(1,4,10);',
 'INSERT INTO "SELLS" VALUES(2,1,14);',
 'INSERT INTO "SELLS" VALUES(3,2,8);',
 'INSERT INTO "SELLS" VALUES(1,1,10);',
 'INSERT INTO "SELLS" VALUES(2,2,8);',
 'INSERT INTO "SELLS" VALUES(3,3,11);',
 'INSERT INTO "SELLS" VALUES(4,4,20);',
 'INSERT INTO "SELLS" VALUES(5,1,1);',
 'INSERT INTO "SELLS" VALUES(6,2,40);',
 'INSERT INTO "SELLS" VALUES(7,3,50);',
 'INSERT INTO "SELLS" VALUES(1,4,10);',
 'INSERT INTO "SELLS" VALUES(2,1,14);',
 'INSERT INTO "SELLS" VALUES(3,2,8);',
 'CREATE TABLE SUPPLIER(\n    SNO   INTEGER PRIMARY KEY,\n    SNAME TEXT,\n    CNO   IN
TEGER NOT NULL\n);',
 'INSERT INTO "SUPPLIER" VALUES(1,\'1호점\',1);',
 'INSERT INTO "SUPPLIER" VALUES(2,\'2호점\',2);',
 'INSERT INTO "SUPPLIER" VALUES(3,\'3호점\',3);',
 'INSERT INTO "SUPPLIER" VALUES(4,\'4호점\',4);',
 'INSERT INTO "SUPPLIER" VALUES(5,\'5호점\',1);',
 'INSERT INTO "SUPPLIER" VALUES(6,\'6호점\',2);',
 'INSERT INTO "SUPPLIER" VALUES(7,\'7호점\',3);',
 'COMMIT;']
```