import sqlite3

conn = sqlite3.connect('example.db')

conn = sqlite3.connect('database.db')

cursor = conn.cursor()

cursor.execute("SELECT name FROM sqlite\_master WHERE type='table';")

print(cursor.fetchall())

tạo bảng

c = conn.cursor()

*# Create table*

c.execute('''CREATE TABLE stocks

(date text, trans text, symbol text, qty real, price real)''')

*# Insert a row of data*

c.execute("INSERT INTO stocks VALUES ('2006-01-05','BUY','RHAT',100,35.14)")

*# Save (commit) the changes*

conn.commit()

*# We can also close the connection if we are done with it.*

*# Just be sure any changes have been committed or they will be lost.*

conn.close()

CREATE TABLE `TEST\_TRIGGER` (

`ID` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

`NAME` TEXT NOT NULL,

`ENDNODE` TEXT NOT NULL

);

 CASE x WHEN w1 THEN r1 WHEN w2 THEN r2 ELSE r3 END

 CASE WHEN x=w1 THEN r1 WHEN x=w2 THEN r2 ELSE r3 END

CREATE TABLE `network\_network` (

`ID` integer NOT NULL PRIMARY KEY AUTOINCREMENT,

`MODE` varchar ( 20 ) NOT NULL,

`SSID` varchar ( 50 ) NOT NULL,

`PSK` varchar ( 50 ) NOT NULL,

`IP` varchar ( 20 ) NOT NULL,

`GateWay` varchar ( 20 ) NOT NULL,

`SubnetMask` varchar ( 20 ) NOT NULL,

`IP\_Mode` varchar ( 20 ) NOT NULL,

`SignalStrength` varchar ( 10 ) NOT NULL,

`Update` datetime NOT NULL

);

c.execute(''' CREATE TABLE `network\_network\_2` (

`ID` integer NOT NULL PRIMARY KEY AUTOINCREMENT,

`MODE` varchar ( 20 ) NOT NULL,

`SSID` varchar ( 50 ) NOT NULL,

`PSK` varchar ( 50 ) NOT NULL,

`IP` varchar ( 20 ) NOT NULL,

`GateWay` varchar ( 20 ) NOT NULL,

`SubnetMask` varchar ( 20 ) NOT NULL,

`IP\_Mode` varchar ( 20 ) NOT NULL,

`SignalStrength` varchar ( 10 ) NOT NULL,

`Update` datetime NOT NULL)''')

Xóa bảng

c.execute(''' DROP TABLE 'network\_network\_2' ''')

conn.commit()

SELECT name FROM sqlite\_master WHERE type = 'trigger'

DROP TRIGGER AFTER;

DROP TRIGGER BBB;

CREATE TRIGGER audit\_log AFTER INSERT

ON MANAGE\_ENDNODE

BEGIN

INSERT INTO TEST\_TRIGGER(ID, NAME, ENDNODE) VALUES (NEW.ID, INSERTED.NODENAME, INSERTED.NODENAME);

END;

CREATE TABLE emp (

name VARCHAR2(10),

deptno NUMBER,

sal NUMBER,

comm NUMBER

)

/

CREATE TABLE exception (

name VARCHAR2(10),

old\_sal NUMBER,

new\_sal NUMBER

)

/

CREATE OR REPLACE TRIGGER emp\_comm\_trig

BEFORE INSERT OR UPDATE ON emp

FOR EACH ROW

BEGIN

IF (:NEW.deptno = 30 and INSERTING) THEN

:NEW.comm := :NEW.sal \* .4;

END IF;

IF (UPDATING and (:NEW.sal - :OLD.sal) > :OLD.sal \* .5) THEN

INSERT INTO exception VALUES (:NEW.name, :OLD.sal, :NEW.sal);

END IF;

END

/

SELECT

trackid,

name,

composer,

unitprice

FROM

tracks;

CREATE TRIGGER audit\_log AFTER INSERT

ON MANAGE\_ENDNODE

BEGIN

CASE

WHEN NEW.NAME NOT LIKE '%\_@\_\_%.\_\_%' THEN

RAISE (

ABORT,

'Invalid email address'

)

END;

END;

DROP TRIGGER audit\_log;

CREATE TRIGGER audit\_log AFTER INSERT

ON MANAGE\_ENDNODE

BEGIN

SELECT

CASE

WHEN NEW.NAME NOT LIKE '%\_@\_\_%.\_\_%' THEN

RAISE ( ABORT, 'Invalid email address' )

END;

END;

CREATE TRIGGER audit\_log AFTER INSERT

ON MANAGE\_ENDNODE

FOR EACH ROW

WHEN NEW.NAME = 'RLACS'

BEGIN

INSERT INTO RLACS (IDNODE, D1, D2, A1, A2) VALUES('RLACS01','BUY','RHAT','3','4');

END;

CREATE TRIGGER audit\_log AFTER INSERT ON MANAGE\_ENDNODE

BEGIN

SELECT

CASE

WHEN NEW.NAME = 'RLACS' THEN

INSERT INTO RLACS (IDNODE, D1, D2, A1, A2) VALUES('RLACS01','BUY','RHAT','3','4');

END;

END;

CREATE TRIGGER audit\_log\_1 AFTER INSERT

ON MANAGE\_ENDNODE

FOR EACH ROW

WHEN NEW.NAME = 'RLACS'

BEGIN

INSERT INTO RLACS (IDNODE, D1, D2, A1, A2) VALUES(NEW.IDNODE, 'OFF', 'OFF', '0', '0');

END;

CREATE TRIGGER audit\_log\_2 AFTER INSERT

ON MANAGE\_ENDNODE

FOR EACH ROW

WHEN NEW.NAME = 'RLTDS'

BEGIN

INSERT INTO RLTDS (IDNODE, D, TDS) VALUES(NEW.IDNODE, 'OFF', '0');

END;

CREATE TRIGGER audit\_log\_3 AFTER INSERT

ON MANAGE\_ENDNODE

FOR EACH ROW

WHEN NEW.NAME = 'THL\_SENSOR'

BEGIN

INSERT INTO THL\_SENSOR (IDNODE, T, H, L) VALUES(NEW.IDNODE, '0', '0', '0');

END;

CREATE TRIGGER audit\_log\_11 AFTER DELETE

ON MANAGE\_ENDNODE

FOR EACH ROW

WHEN OLD.NAME = 'RLACS'

BEGIN

DELETE FROM RLACS WHERE IDNODE = OLD.IDNODE;

END;

CREATE TRIGGER audit\_log\_22 AFTER DELETE

ON MANAGE\_ENDNODE

FOR EACH ROW

WHEN OLD.NAME = 'RLTDS'

BEGIN

DELETE FROM RLTDS WHERE IDNODE = OLD.IDNODE;

END;

CREATE TRIGGER audit\_log\_33 AFTER DELETE

ON MANAGE\_ENDNODE

FOR EACH ROW

WHEN OLD.NAME = 'THL\_SENSOR'

BEGIN

DELETE FROM THL\_SENSOR WHERE IDNODE = OLD.IDNODE;

END;

INSERT INTO MANAGE\_ENDNODE(NAME, IDNODE) VALUES ('RLACS', 'RLACS01');

INSERT INTO MANAGE\_ENDNODE(NAME, IDNODE) VALUES ('RLTDS', 'RLTDS01');

INSERT INTO MANAGE\_ENDNODE(NAME, IDNODE) VALUES ('THL\_SENSOR', 'THL01');

INSERT INTO MANAGE\_ENDNODE(NAME, IDNODE) VALUES ('RLACS', 'RLACS02');

INSERT INTO MANAGE\_ENDNODE(NAME, IDNODE) VALUES ('RLTDS', 'RLTDS02');

INSERT INTO MANAGE\_ENDNODE(NAME, IDNODE) VALUES ('THL\_SENSOR', 'THL02');