

1 **sqlite3.connect(database)**

en API der skaber en forbindelse til en SQLite database. Findes der ingen database fil forsøger den at lave en. Er den "Successfull" vil den returnere et Connection objekt.

Når en forbindelse laver ændringer i databasen bliver DB'en låst indtil den bliver "Committed" Der er som standart en timeout på ca 5sec for denne lås. Man bør derfor Committe inden for denne tid.

```
con = sql.connect('test.db')
```

2 **connection.cursor()**

This routine creates a cursor which will be used throughout of your database programming with Python. This method accepts a single optional parameter cursorClass. If supplied, this must be a custom cursor class that extends sqlite3.Cursor.

Eksempel: `cur = con.cursor()`

3 **cursor.execute(sql sætning)**

denne rutine eksekvere en SQL sætning

Eksempel: `cur.execute("CREATE TABLE IF NOT EXISTS Cars(Id INT, Name TEXT, Price INT)")`

```
cur.execute("insert into people values ('Per', 45)", (who, age))
```

```
cur.execute("INSERT INTO Cars VALUES(1,'%s',52642)" % (var1))
```

4 **connection.commit()**

Denne rutine gemmer ændringerne i Databasen (kan sammenlignes med "save" i word)

Eksempel: `con.commit()`

5 **connection.close()**

Lukker forbindelsen til databasen

Eksempel: `con.close()`

6 **cursor.fetchone()**

Denne rutine henter en enkelt række af resultater fra en SQL query eller None hvis intet blev fundet.

Eksempel: `row = cur.fetchone()`

7 **cursor.fetchall()**

Samme som fetchone men henter alle rækker på en gang

SQL Sætninger

```
ALTER TABLE table_navn ADD column_navn datatype
```

```
ALTER TABLE table_navn DROP COLUMN column_navn
```

```
CREATE TABLE IF NOT EXISTS table_navn(name datatype, name datatype, name datatype)
```

```
INSERT into table_navn (navn, navn) VALUES('tekst_data', int_data)
```

```
SELECT * FROM table_navn
```

```
DELETE FROM table_navn WHERE et_navn=en_værdi
```

Eksempel: `rows = cur.fetchall()`