

Modifying a Database

This pdf is completed and recording is watched

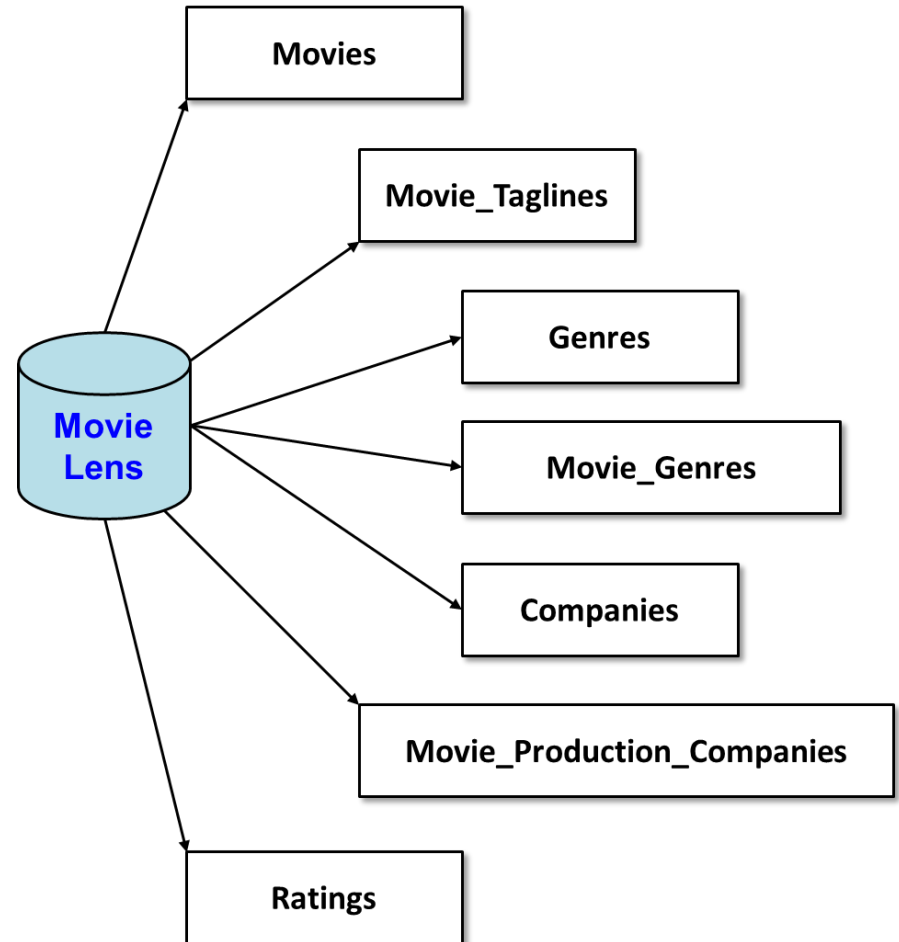
- **Modifying a database**
- **SQL action queries: insert, update, delete**
- **Committing changes**



MovieLens database

- **MovieLens**

- <https://movielens.org/>
- *45K movies*
- *26M reviews*




Action queries

- Action queries modify a database
- Most common forms:
 - *Insert*
 - *Update*
 - *Delete*

```
INSERT INTO tablename(column1, column2, ...)  
VALUES(value1, value2, ...);
```

```
UPDATE tablename  
SET column1 = newvalue1, column2 = newvalue2, ...  
[ WHERE condition ] ;
```

```
DELETE FROM tablename  
[ WHERE condition ] ;
```



Be careful! If you omit
the “where” clause,
updates/deletes all
rows in the table!

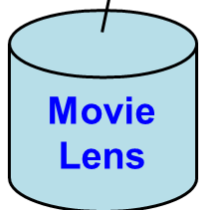
Example

- Insert a new review...

```
INSERT INTO Ratings(Movie_ID, Rating)
VALUES(603, 10);
```

Movie_ID	Title	Release_Date	Runtime	Original_L anguage	Budget	Revenue
603	The Matrix	1999-03-30 00:00:00.000	136	en	63000000	463517383
862	Toy Story	1995-10-30 00:00:00.000	81	en	30000000	373554033

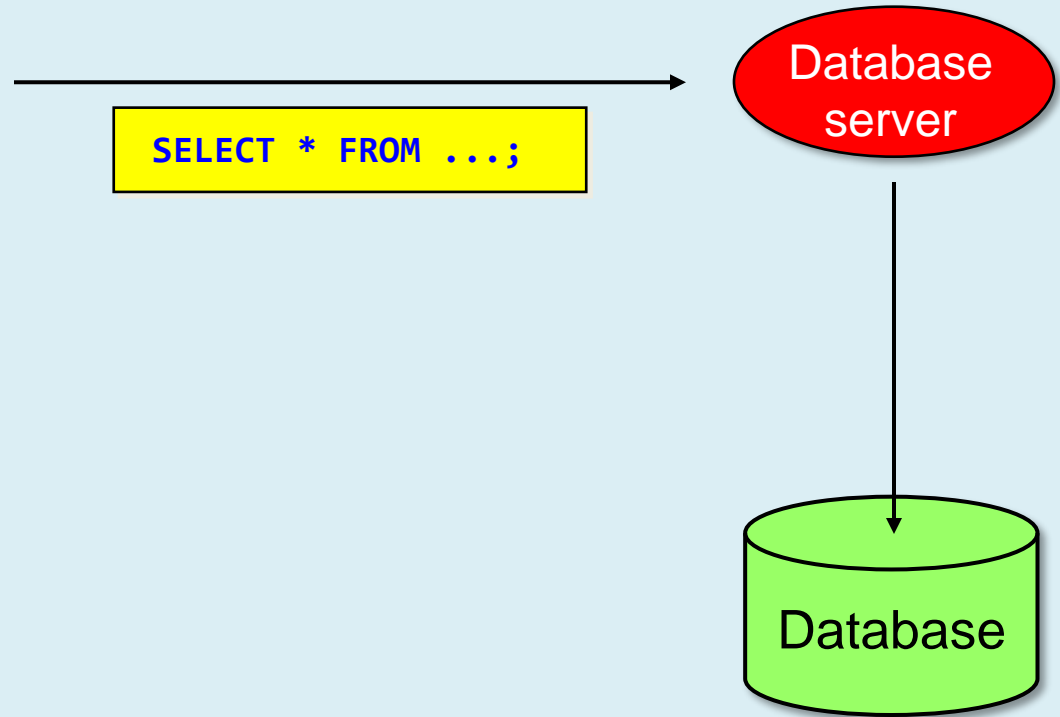
Movies



Ratings

Movie_ID	Rating
605	8
603	6
605	10
605	6

Executing SQL from Python



Example

- Insert a new review...

```
import sqlite3

dbConn = sqlite3.connect("movielens.db")
dbCursor = dbConn.cursor()

rating = input("What rating do you give 'The Matrix'? ")

# insert rating:
sql = """
    insert into ratings(movie_id, rating) values(603, ?);
"""

dbCursor.execute(sql, [rating])
inserted = dbCursor.rowcount

print("Rows inserted:", inserted)
```

This doesn't work ---
the changes aren't
saved?!?!



Commit your changes

- You need to *commit* (save) your changes, or *rollback* (undo) your changes

```
import sqlite3

dbConn = sqlite3.connect("movielens.db")
dbCursor = dbConn.cursor()

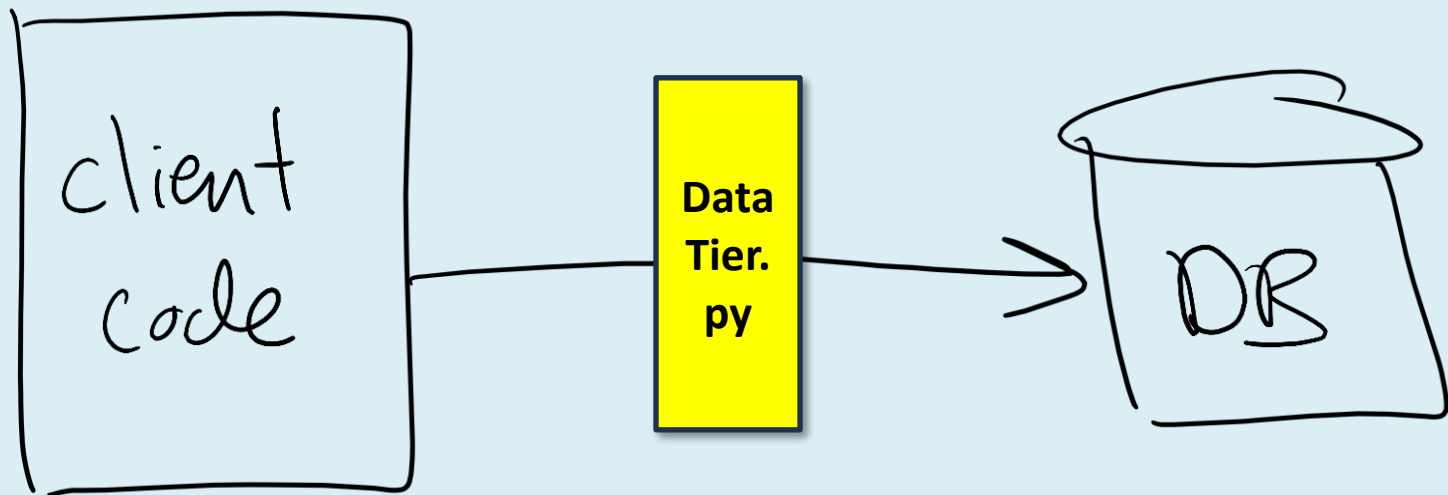
rating = input("What rating do you give 'The Matrix'? ")

# insert rating:
sql = """
    insert into ratings(movie_id, rating) values(603, ?);
"""

dbCursor.execute(sql, [rating])
inserted = dbCursor.rowcount

if inserted > 0:
    dbConn.commit()
else:
    dbConn.rollback()
```


Data tier



Executing actions (insert, update, delete)

```
sql = "INSERT INTO Ratings(Movie_ID, Rating) VALUES(603, 10);"
added = datatier.perform_action(dbConn, sql)
if added != 1:
    print("error!")
```

```
#
# perform_action: executes action query and returns the # of
#                 rows modified; returns -1 on an error.
#
def perform_action(dbConn, sql, parameters=[]):
    dbCursor = dbConn.cursor()

    try:
        # try to execute, and if successful commit the changes
        # and return the # of rows modified by the query:
        dbCursor.execute(sql, parameters)
        dbConn.commit()
        return dbCursor.rowcount

    except Exception as e:
        # if it fails, print an error msg and return -1:
        dbConn.rollback()
        logging.error('perform_action failed:')
        logging.error(e)
        return -1

    finally:
        # cleanup code that gets executed either way:
        dbCursor.close()
```

That's it, thank you!