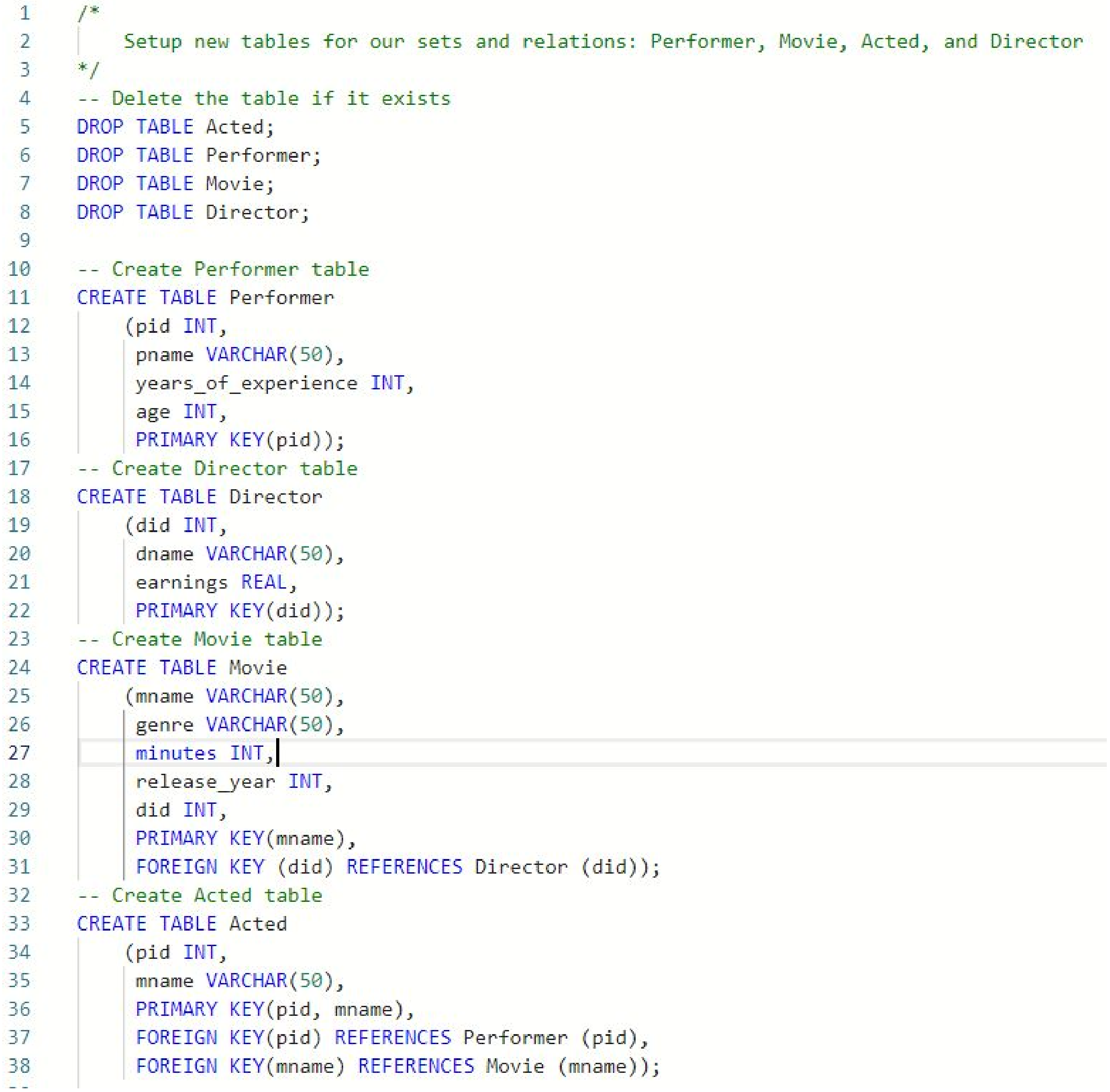
Problem 2.

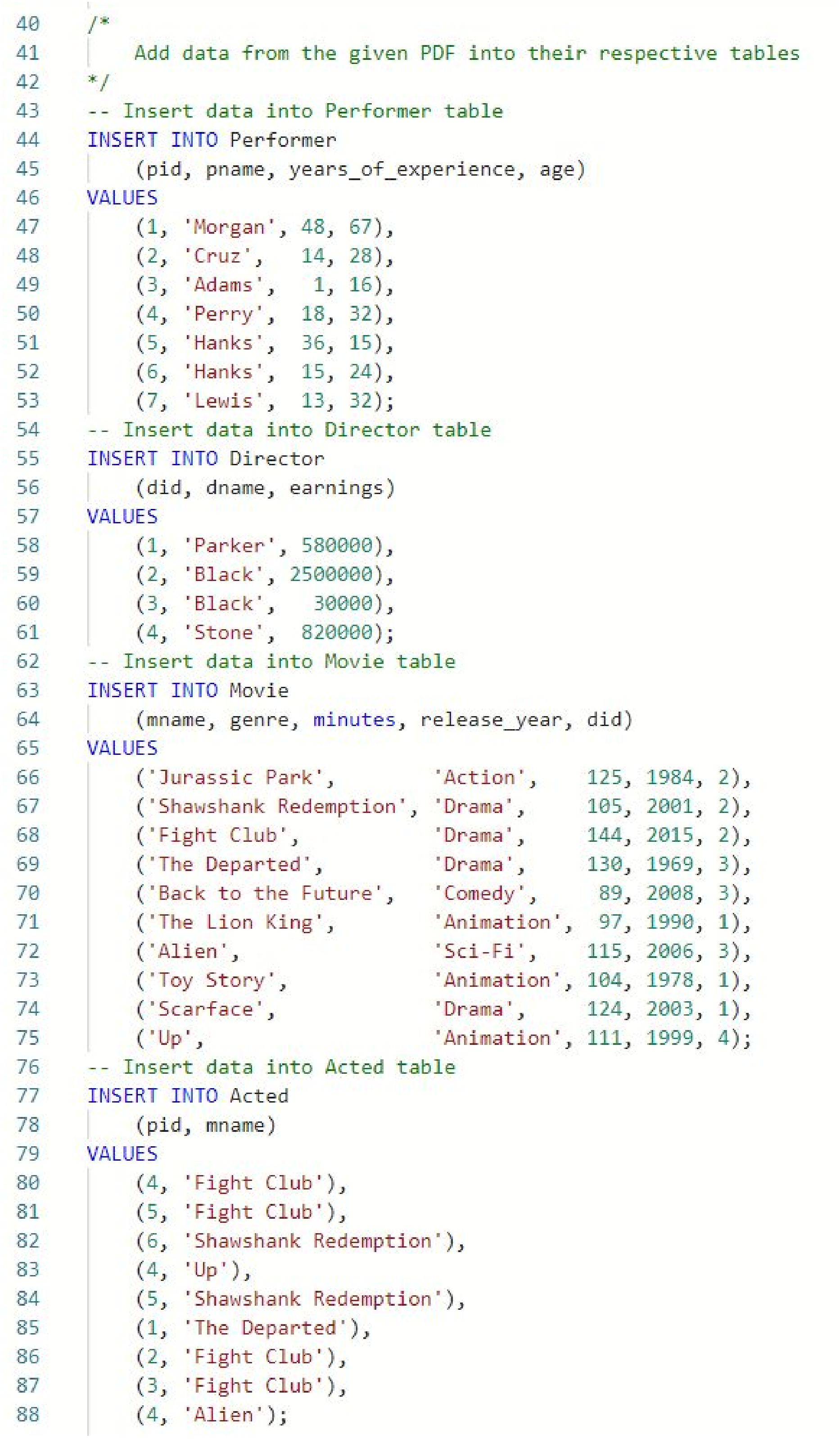
a) *Create the relations by SQL*



```
/*
    Setup new tables for our sets and relations: Performer, Movie, Acted, and Director
*/
-- Delete the table if it exists
DROP TABLE Acted;
DROP TABLE Performer;
DROP TABLE Movie;
DROP TABLE Director;

-- Create Performer table
CREATE TABLE Performer
    (pid INT,
     pname VARCHAR(50),
     years_of_experience INT,
     age INT,
     PRIMARY KEY(pid));
-- Create Director table
CREATE TABLE Director
    (did INT,
     dname VARCHAR(50),
     earnings REAL,
     PRIMARY KEY(did));
-- Create Movie table
CREATE TABLE Movie
    (mname VARCHAR(50),
     genre VARCHAR(50),
     minutes INT,
     release_year INT,
     did INT,
     PRIMARY KEY(mname),
     FOREIGN KEY (did) REFERENCES Director (did));
-- Create Acted table
CREATE TABLE Acted
    (pid INT,
     mname VARCHAR(50),
     PRIMARY KEY(pid, mname),
     FOREIGN KEY(pid) REFERENCES Performer (pid),
     FOREIGN KEY(mname) REFERENCES Movie (mname));
```

b) Populate the relation

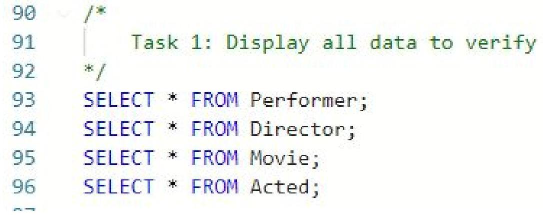


```sql
/*
    Add data from the given PDF into their respective tables
*/
-- Insert data into Performer table
INSERT INTO Performer
    (pid, pname, years_of_experience, age)
VALUES
    (1, 'Morgan', 48, 67),
    (2, 'Cruz',   14, 28),
    (3, 'Adams',   1, 16),
    (4, 'Perry',  18, 32),
    (5, 'Hanks',  36, 15),
    (6, 'Hanks',  15, 24),
    (7, 'Lewis',  13, 32);
-- Insert data into Director table
INSERT INTO Director
    (did, dname, earnings)
VALUES
    (1, 'Parker', 580000),
    (2, 'Black', 2500000),
    (3, 'Black',   30000),
    (4, 'Stone',  820000);
-- Insert data into Movie table
INSERT INTO Movie
    (mname, genre, minutes, release_year, did)
VALUES
    ('Jurassic Park',        'Action',    125, 1984, 2),
    ('Shawshank Redemption', 'Drama',     105, 2001, 2),
    ('Fight Club',           'Drama',     144, 2015, 2),
    ('The Departed',         'Drama',     130, 1969, 3),
    ('Back to the Future',   'Comedy',     89, 2008, 3),
    ('The Lion King',        'Animation',  97, 1990, 1),
    ('Alien',                'Sci-Fi',    115, 2006, 3),
    ('Toy Story',            'Animation', 104, 1978, 1),
    ('Scarface',             'Drama',     124, 2003, 1),
    ('Up',                   'Animation', 111, 1999, 4);
-- Insert data into Acted table
INSERT INTO Acted
    (pid, mname)
VALUES
    (4, 'Fight Club'),
    (5, 'Fight Club'),
    (6, 'Shawshank Redemption'),
    (4, 'Up'),
    (5, 'Shawshank Redemption'),
    (1, 'The Departed'),
    (2, 'Fight Club'),
    (3, 'Fight Club'),
    (4, 'Alien');
```

c) Implement the SQL queries

1.

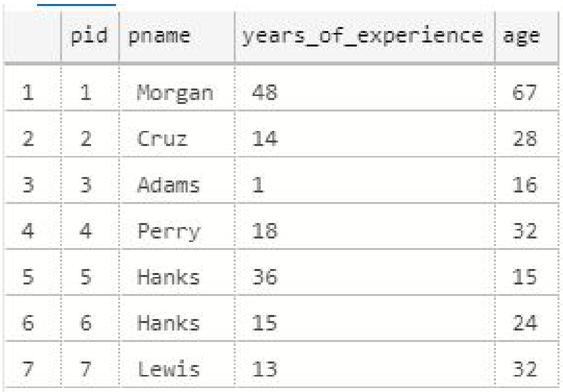
Query



```
/*
    Task 1: Display all data to verify
*/
SELECT * FROM Performer;
SELECT * FROM Director;
SELECT * FROM Movie;
SELECT * FROM Acted;
```

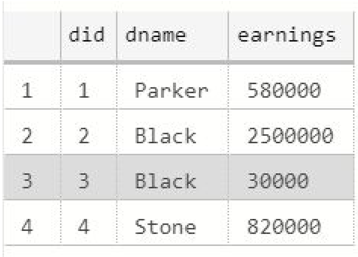
Output

Performer



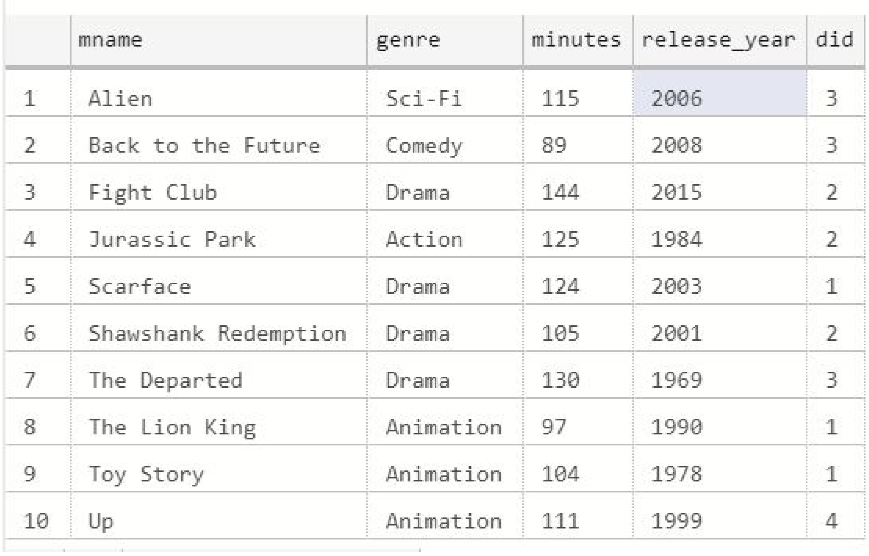
	pid	pname	years_of_experience	age
1	1	Morgan	48	67
2	2	Cruz	14	28
3	3	Adams	1	16
4	4	Perry	18	32
5	5	Hanks	36	15
6	6	Hanks	15	24
7	7	Lewis	13	32

Director



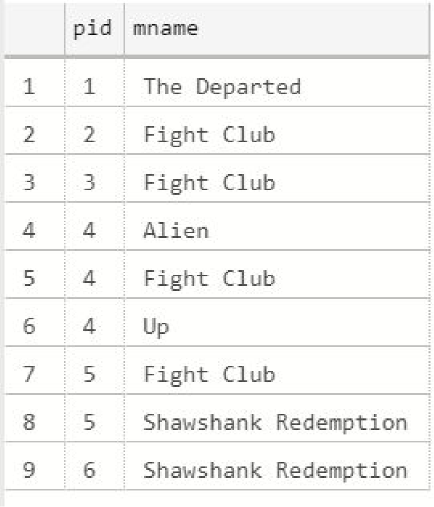
	did	dname	earnings
1	1	Parker	580000
2	2	Black	2500000
3	3	Black	30000
4	4	Stone	820000

Movie



	mname	genre	minutes	release_year	did
1	Alien	Sci-Fi	115	2006	3
2	Back to the Future	Comedy	89	2008	3
3	Fight Club	Drama	144	2015	2
4	Jurassic Park	Action	125	1984	2
5	Scarface	Drama	124	2003	1
6	Shawshank Redemption	Drama	105	2001	2
7	The Departed	Drama	130	1969	3
8	The Lion King	Animation	97	1990	1
9	Toy Story	Animation	104	1978	1
10	Up	Animation	111	1999	4

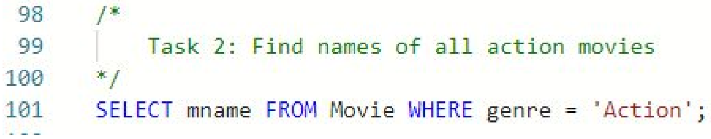
Acted



	pid	mname
1	1	The Departed
2	2	Fight Club
3	3	Fight Club
4	4	Alien
5	4	Fight Club
6	4	Up
7	5	Fight Club
8	5	Shawshank Redemption
9	6	Shawshank Redemption

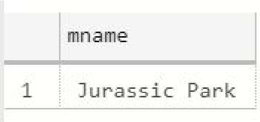
2.

Query



```
/*
    Task 2: Find names of all action movies
*/
SELECT mname FROM Movie WHERE genre = 'Action';
```

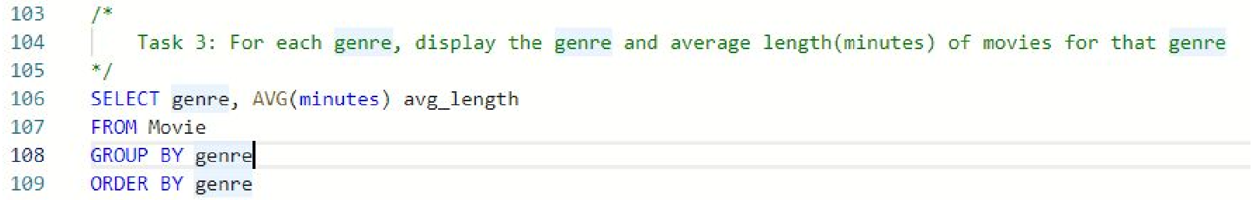
Output



	mname
1	Jurassic Park

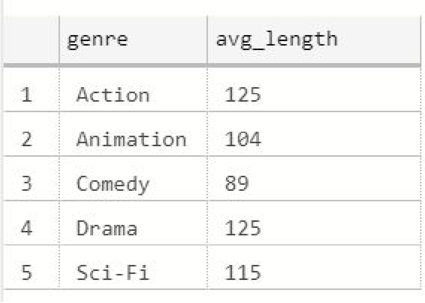
3.

Query



```
/*
    Task 3: For each genre, display the genre and average length(minutes) of movies for that genre
*/
SELECT genre, AVG(minutes) avg_length
FROM Movie
GROUP BY genre
ORDER BY genre
```

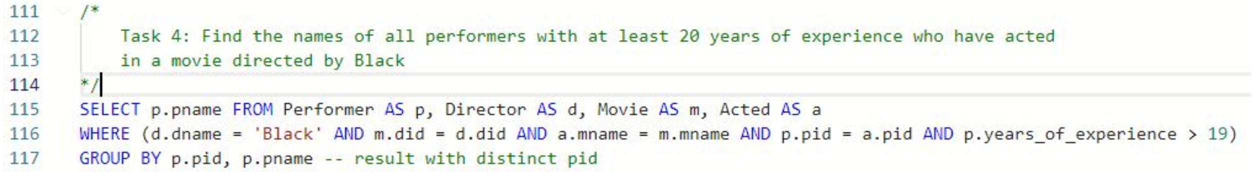
Output



	genre	avg_length
1	Action	125
2	Animation	104
3	Comedy	89
4	Drama	125
5	Sci-Fi	115

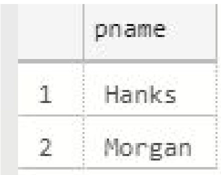
4.

Query



```
/*
    Task 4: Find the names of all performers with at least 20 years of experience who have acted
    in a movie directed by Black
*/
SELECT p.pname FROM Performer AS p, Director AS d, Movie AS m, Acted AS a
WHERE (d.dname = 'Black' AND m.did = d.did AND a.mname = m.mname AND p.pid = a.pid AND p.years_of_experience > 19)
GROUP BY p.pid, p.pname -- result with distinct pid
```

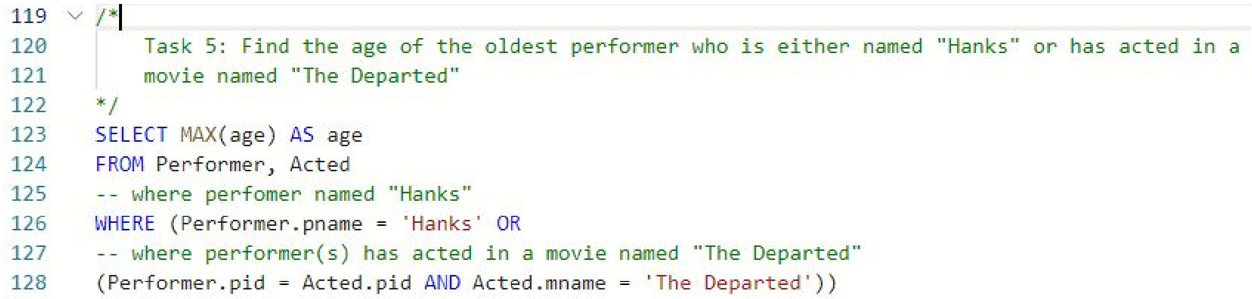
Output



	pname
1	Hanks
2	Morgan

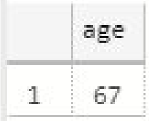
5.

Query



```
/*
    Task 5: Find the age of the oldest performer who is either named "Hanks" or has acted in a
    movie named "The Departed"
*/
SELECT MAX(age) AS age
FROM Performer, Acted
-- where perfomer named "Hanks"
WHERE (Performer.pname = 'Hanks' OR
-- where performer(s) has acted in a movie named "The Departed"
(Performer.pid = Acted.pid AND Acted.mname = 'The Departed'))
```

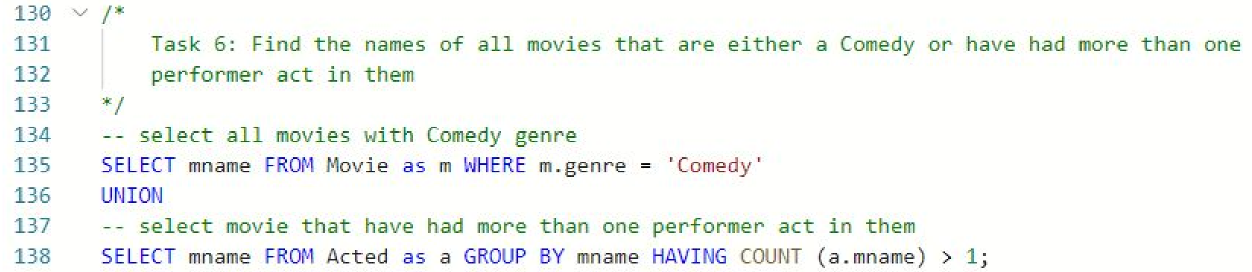
Output



	age
1	67

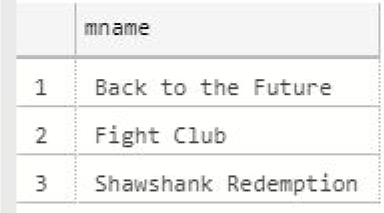
6.

Query



```
/*
    Task 6: Find the names of all movies that are either a Comedy or have had more than one
    performer act in them
*/
-- select all movies with Comedy genre
SELECT mname FROM Movie as m WHERE m.genre = 'Comedy'
UNION
-- select movie that have had more than one performer act in them
SELECT mname FROM Acted as a GROUP BY mname HAVING COUNT (a.mname) > 1;
```

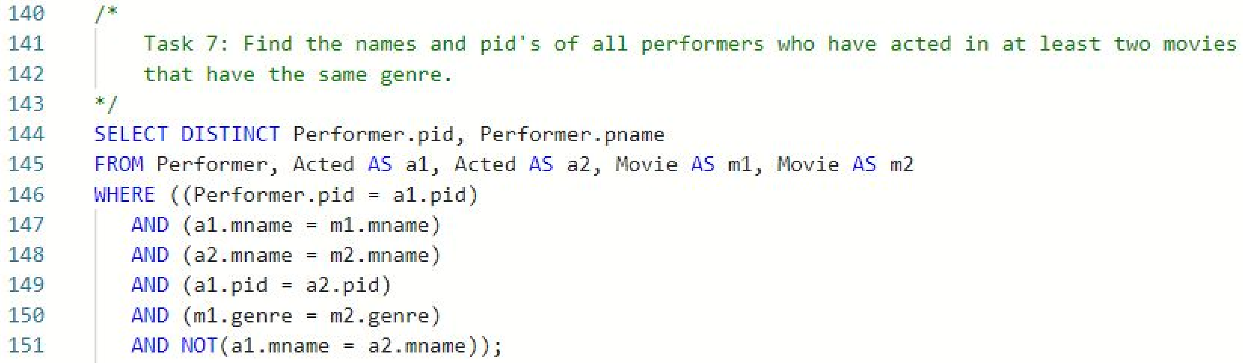
Output



	mname
1	Back to the Future
2	Fight Club
3	Shawshank Redemption

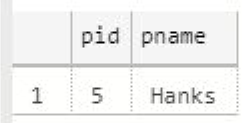
7.

Query



```
/*
    Task 7: Find the names and pid's of all performers who have acted in at least two movies
    that have the same genre.
*/
SELECT DISTINCT Performer.pid, Performer.pname
FROM Performer, Acted AS a1, Acted AS a2, Movie AS m1, Movie AS m2
WHERE ((Performer.pid = a1.pid)
   AND (a1.mname = m1.mname)
   AND (a2.mname = m2.mname)
   AND (a1.pid = a2.pid)
   AND (m1.genre = m2.genre)
   AND NOT(a1.mname = a2.mname));
```

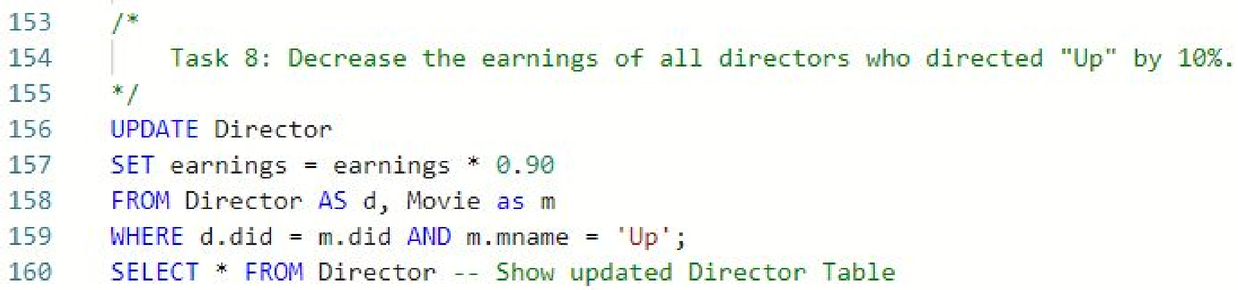
Output



	pid	pname
1	5	Hanks

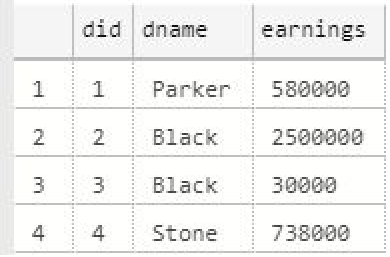
8.

Query



```
/*
    Task 8: Decrease the earnings of all directors who directed "Up" by 10%.
*/
UPDATE Director
SET earnings = earnings * 0.90
FROM Director AS d, Movie as m
WHERE d.did = m.did AND m.mname = 'Up';
SELECT * FROM Director -- Show updated Director Table
```

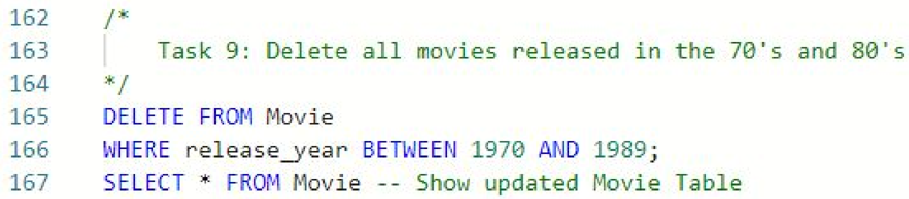
Output



	did	dname	earnings
1	1	Parker	580000
2	2	Black	2500000
3	3	Black	30000
4	4	Stone	738000

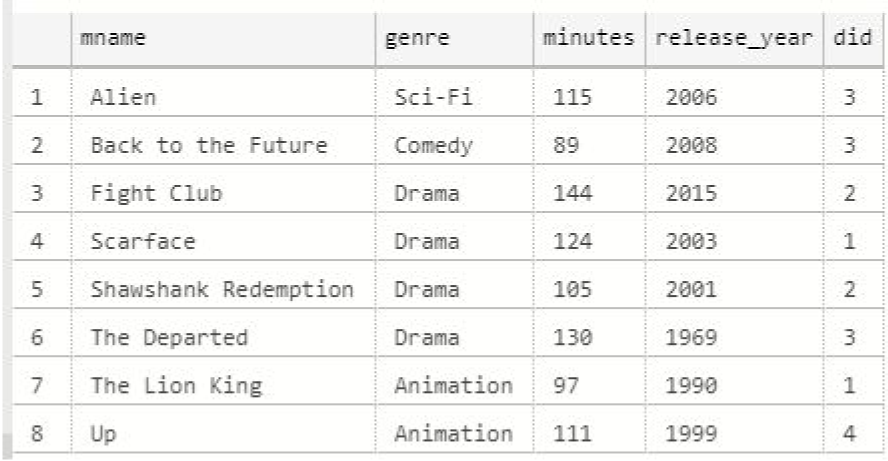
9.

Query



```
/*
    Task 9: Delete all movies released in the 70's and 80's
*/
DELETE FROM Movie
WHERE release_year BETWEEN 1970 AND 1989;
SELECT * FROM Movie -- Show updated Movie Table
```

Output



	mname	genre	minutes	release_year	did
1	Alien	Sci-Fi	115	2006	3
2	Back to the Future	Comedy	89	2008	3
3	Fight Club	Drama	144	2015	2
4	Scarface	Drama	124	2003	1
5	Shawshank Redemption	Drama	105	2001	2
6	The Departed	Drama	130	1969	3
7	The Lion King	Animation	97	1990	1
8	Up	Animation	111	1999	4