

- A. Hent ut alle filmene sortert på 'year'
- B. **SELECT** *
- C. **FROM** movies_imdb.movie
- D. **ORDER BY** year ;

Navigator: CREATE_AND_INSERT_movie... movie x

SCHEMAS

Filter objects

- movies_imdb
 - Tables
 - actor
 - director
 - genre
 - movie
 - movie_has_actor
 - movie_has_genre
 - Views
 - Stored Procedures
 - Functions
- oblig2
- oblig_1
- sakila
- sys
- videogame
- videogame_lecture5
 - Tables
 - platform
 - videogame
 - Views
 - Stored Procedures
 - Functions
- world

Administration Schemas

Information

Table: movie

Columns:

- id int PK
- title varchar(100)
- year year
- runtime int
- imdb_rating float

Result Grid

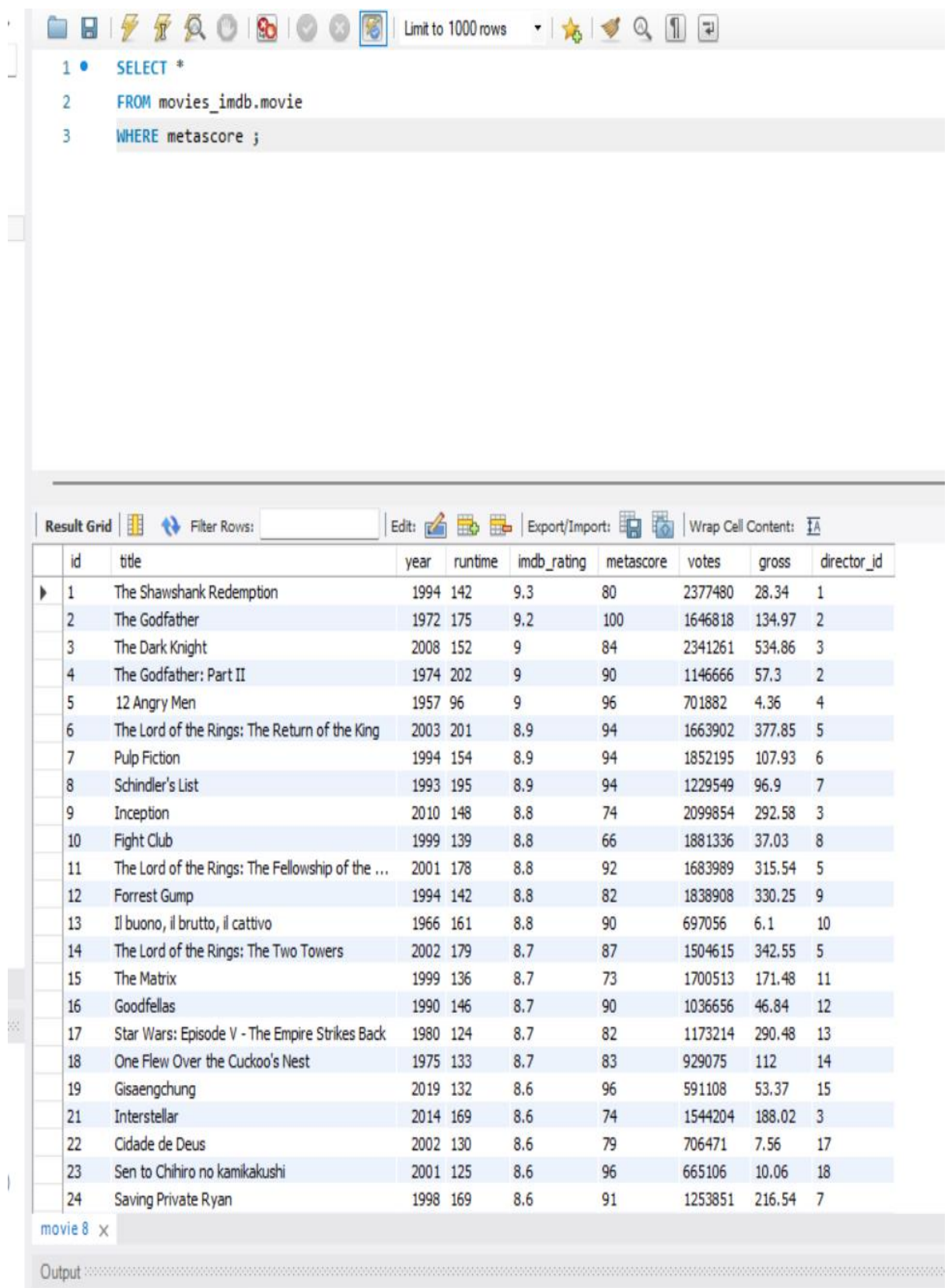
Filter Rows: Edit: Export/Import: Wrap Cell Content: Fel

	id	title	year	runtime	imdb_rating	metascore	votes	gross	director_id
▶	328	Das Cabinet des Dr. Caligari	1920	76	8.1	NULL	58203	NULL	216
	127	The Kid	1921	68	8.3	NULL	115516	5.45	40
	573	Nosferatu	1922	94	7.9	NULL	89728	NULL	214
	197	Sherlock Jr.	1924	45	8.2	NULL	43176	0.98	133
	196	The Gold Rush	1925	95	8.2	NULL	102657	5.45	40
	468	Bronenosets Potemkin	1925	75	8	97	53764	0.05	299
	327	The General	1926	67	8.1	NULL	82299	1.03	215
	126	Metropolis	1927	153	8.3	98	162045	1.24	84
	326	Sunrise: A Song of Two Humans	1927	94	8.1	NULL	47286	0.54	214
	324	La passion de Jeanne d'Arc	1928	110	8.1	NULL	48764	0.02	213
	325	The Circus	1928	72	8.1	90	30743	NULL	40
	467	All Quiet on the Western Front	1930	152	8	91	57754	3.27	298
	53	City Lights	1931	87	8.5	99	170185	0.02	40
	125	M - Eine Stadt sucht einen Mörder	1931	117	8.3	NULL	145557	0.03	84
	727	Frankenstein	1931	70	7.8	91	66051	NULL	430
	572	Freaks	1932	64	7.9	80	42543	NULL	354
	726	Scarface: The Shame of the Na...	1932	93	7.8	87	25673	NULL	432
	571	King Kong	1933	100	7.9	90	80179	10	353
	725	Duck Soup	1933	69	7.8	93	55978	NULL	431
	883	The Invisible Man	1933	71	7.7	87	31185	NULL	430
	323	It Happened One Night	1934	105	8.1	87	95393	4.36	24
	466	The Thin Man	1934	91	8	86	26958	NULL	297
	570	A Night at the Opera	1935	96	7.9	NULL	30845	2.54	352

movie 3 x

B. Hent ut alle filmene som *har* en metascore.

```
SELECT *  
FROM movies_imdb.movie  
WHERE metascore ;
```



The screenshot shows a database query interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, the SQL query is entered in a text area:

```
1 • SELECT *  
2 FROM movies_imdb.movie  
3 WHERE metascore ;
```

Below the query editor, the results are displayed in a table. The table has 10 columns: id, title, year, runtime, imdb_rating, metascore, votes, gross, and director_id. The results are sorted by metascore in descending order. The first 24 rows are visible, showing a list of movies with their respective details.

	id	title	year	runtime	imdb_rating	metascore	votes	gross	director_id
▶	1	The Shawshank Redemption	1994	142	9.3	80	2377480	28.34	1
	2	The Godfather	1972	175	9.2	100	1646818	134.97	2
	3	The Dark Knight	2008	152	9	84	2341261	534.86	3
	4	The Godfather: Part II	1974	202	9	90	1146666	57.3	2
	5	12 Angry Men	1957	96	9	96	701882	4.36	4
	6	The Lord of the Rings: The Return of the King	2003	201	8.9	94	1663902	377.85	5
	7	Pulp Fiction	1994	154	8.9	94	1852195	107.93	6
	8	Schindler's List	1993	195	8.9	94	1229549	96.9	7
	9	Inception	2010	148	8.8	74	2099854	292.58	3
	10	Fight Club	1999	139	8.8	66	1881336	37.03	8
	11	The Lord of the Rings: The Fellowship of the ...	2001	178	8.8	92	1683989	315.54	5
	12	Forrest Gump	1994	142	8.8	82	1838908	330.25	9
	13	Il buono, il brutto, il cattivo	1966	161	8.8	90	697056	6.1	10
	14	The Lord of the Rings: The Two Towers	2002	179	8.7	87	1504615	342.55	5
	15	The Matrix	1999	136	8.7	73	1700513	171.48	11
	16	Goodfellas	1990	146	8.7	90	1036656	46.84	12
	17	Star Wars: Episode V - The Empire Strikes Back	1980	124	8.7	82	1173214	290.48	13
	18	One Flew Over the Cuckoo's Nest	1975	133	8.7	83	929075	112	14
	19	Gisaengchung	2019	132	8.6	96	591108	53.37	15
	21	Interstellar	2014	169	8.6	74	1544204	188.02	3
	22	Cidade de Deus	2002	130	8.6	79	706471	7.56	17
	23	Sen to Chihiro no kamikakushi	2001	125	8.6	96	665106	10.06	18
	24	Saving Private Ryan	1998	169	8.6	91	1253851	216.54	7

Below the table, there is a tab labeled "movie 8" with a close button (X). At the bottom, there is an "Output" section with a dotted line indicating where the output would be displayed.

C. Hent ut filmer og hvem som har regissert filmen (director)

```
SELECT movie.title, director.name as director
FROM movie
INNER JOIN director ON movie.director_id = director.id
```

The screenshot shows a database management tool interface. On the left, a sidebar displays a tree view of database objects under the 'movies_imdb' schema, including tables like 'actor', 'director', 'genre', 'movie', 'movie_has_actor', and 'movie_has_genre'. The main area is split into two panes. The top pane shows a SQL query:

```
1 • SELECT movie.title, director.name as director
2 FROM movie
3 INNER JOIN director ON movie.director_id = director.id
```

 The bottom pane shows the 'Result Grid' with columns 'title' and 'director'. It lists 20 rows of movie titles and their directors. The bottom left pane shows the 'Schema' tab for the 'director' table, indicating it has two columns: 'id' (int PK) and 'name' (varchar(45)).

Result Grid

title	director
The Shawshank Redemption	Frank Darabont
The Green Mile	Frank Darabont
The Godfather	Francis Ford Coppola
The Godfather: Part II	Francis Ford Coppola
Apocalypse Now	Francis Ford Coppola
The Conversation	Francis Ford Coppola
The Godfather: Part III	Francis Ford Coppola
The Dark Knight	Christopher Nolan
Inception	Christopher Nolan
Interstellar	Christopher Nolan
The Prestige	Christopher Nolan
The Dark Knight Rises	Christopher Nolan
Memento	Christopher Nolan
Batman Begins	Christopher Nolan
Dunkirk	Christopher Nolan
12 Angry Men	Sidney Lumet
Network	Sidney Lumet
Dog Day Afternoon	Sidney Lumet
The Verdict	Sidney Lumet
Serpico	Sidney Lumet
The Lord of the Rings: The ...	Peter Jackson
The Lord of the Rings: The ...	Peter Jackson
The Lord of the Rings: The ...	Peter Jackson

Schema: director

Columns:

- id int PK
- name varchar(45)

D. Hent navnet og antallet filmer en regissør har regissert, sortert synkende på antallet

```
SELECT director.name, sum(movie.director_id) as directed_movies
FROM director INNER JOIN movie ON director.id = movie.director_id
GROUP BY director.id
ORDER BY directed_movies DESC
```

The screenshot shows a database IDE interface. The top pane displays a SQL query: `SELECT director.name, sum(movie.director_id) as directed_movies FROM director INNER JOIN movie ON director.id = movie.director_id GROUP BY director.id ORDER BY directed_movies DESC`. The bottom pane shows the 'Result Grid' with the following data:

	name	directed_movies
▶	Woody Allen	2502
	Alfonso Cuarón	2268
	Matthew Vaughn	1768
	Don Siegel	1506
	Howard Hawks	1440
	Fred Zinnemann	1388
	James Whale	1290
	Rob Reiner	1288
	Ang Lee	1272
	Terrence Malick	1260
	John Hughes	1245
	Jim Jarmusch	1233
	Spike Lee	1088
	Chris Columbus	1086
	Ethan Coen	1050
	George Cukor	1050
	Federico Fellini	1020
	Richard Linklater	1008
	Hutton	1006
	John Ford	1000
	Jacques Annaud	988
	Robert Altman	976
	Pedro Almodóvar	975

The interface also shows a 'Result 7' tab and an 'Output' pane at the bottom.

- E. Hent navnet og antallet filmer en regissør har regissert, samt totalinntekten (gross) for disse filmene, sortert synkende på inntekten

```
SELECT director.name, sum(movie.director_id) as directed_movies,  
sum(movie.gross) as total_gross  
FROM director INNER JOIN movie ON director.id = movie.director_id  
GROUP BY director.id  
ORDER BY total_gross DESC;
```

CREATE_AND_INSERT_movie... movie x movie director

Limit to 1000 rows

```
1 • SELECT director.name, sum(movie.director_id) as directed_movies, sum(movie.gross) as total_gross  
2 FROM director INNER JOIN movie ON director.id = movie.director_id  
3 GROUP BY director.id  
4 ORDER BY total_gross DESC;
```

or
nre

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	name	directed_movies	total_gross
▶	Steven Spielberg	91	2478.129991531372
	Joe Russo	192	2205.0399780273438
	Christopher Nolan	24	1937.449993133545
	James Cameron	175	1748.2400283813477
	Peter Jackson	25	1597.3099975585938
	Abrams	918	1423.1699829101562
	Robert Zemeckis	45	1049.4400024414062
	David Yates	474	978.9500122070312
	Lee Unkrich	510	916.6100158691406
	Brad Bird	762	893.1800193786621
	Ridley Scott	180	766.7100028991699
	Quentin Tarantino	48	727.0399951934814
	James Gunn	454	722.989990234375
	David Fincher	64	713.3199996948242
	Sam Mendes	228	698.1399841308594
	Joss Whedon	474	648.7900295257568
	Todd Phillips	84	612.77001953125
	Chris Columbus	1086	603.3399963378906
	Clint Eastwood	944	592.7100048065186
	Alfonso Cuarón	2268	582.6399965286255
	Martin Scorsese	120	538.3099994659424
	Gareth Edwards	363	532.1799926757812
	James Mangold	564	517.0299987792969

Result 9 x

Output

Action Output

#	Time	Action	Message
---	------	--------	---------

F. Download all films, as well as associated director and genre (genre)

```
SELECT director.name as director, movie.title, genre.name as genre
FROM movie INNER JOIN director ON movie.director_id = director_id
INNER JOIN movie_has_genre ON movie.id = movie_has_genre.movie_id
INNER JOIN genre ON movie_has_genre.genre_id = genre.id;
```

The screenshot shows a database management tool interface. The top toolbar includes icons for database operations. The main window is titled 'CREATE_AND_INSERT_movie...' and contains a SQL query. The query is as follows:

```
1 • SELECT director.name as director, movie.title, genre.name as genre
2 FROM movie INNER JOIN director ON movie.director_id = director_id
3 INNER JOIN movie_has_genre ON movie.id = movie_has_genre.movie_id
4 INNER JOIN genre ON movie_has_genre.genre_id = genre.id;
```

Below the query editor, the 'Result Grid' is displayed, showing a table with three columns: 'director', 'title', and 'genre'. The table contains 20 rows of data, all with 'History' as the genre. The first row is highlighted with a mouse cursor.

director	title	genre
Frank Darabont	Apollo 13	History
Frank Darabont	Enemy at the Gates	History
Frank Darabont	Bridge of Spies	History
Frank Darabont	Dark Waters	History
Frank Darabont	Zulu	History
Frank Darabont	A Man for All Seasons	History
Frank Darabont	Empire of the Sun	History
Frank Darabont	The Last Emperor	History
Frank Darabont	Malcolm X	History
Frank Darabont	Black Hawk Down	History
Frank Darabont	Joyeux Noël	History
Frank Darabont	The Last King of Scot...	History
Frank Darabont	Frost/Nixon	History
Frank Darabont	The Longest Day	History
Frank Darabont	The Man Who Would...	History
Frank Darabont	The Killing Fields	History
Frank Darabont	Mississippi Burning	History
Frank Darabont	Glory	History
Frank Darabont	The Boy in the Stripe...	History
Frank Darabont	The Trial of the Chic...	History
Frank Darabont	Das weiße Band - Ein...	History
Frank Darabont	Straight Outta Comp...	History
Frank Darabont	Under sandet	History

At the bottom of the interface, there is an 'Output' section with a dropdown menu set to 'Action Output'.

G. Hent ut film og regissør for den filmen som har gitt høyest inntekt

```
SELECT movie.title, director.name as director
FROM movie INNER JOIN director ON movie.director_id = director.id
ORDER BY movie.gross DESC LIMIT 1;
```

The screenshot shows a database management tool interface. At the top, there are tabs for 'CREATE_AND_INSERT_movie...', 'movie', and 'director'. Below the tabs is a toolbar with various icons, including a 'Limit to 1000 rows' dropdown. The main area displays a SQL query in a text editor, numbered 1 to 4. The query is:
1 SELECT movie.title, director.name as director
2 FROM movie INNER JOIN director ON movie.director_id = director.id
3 ORDER BY movie.gross DESC LIMIT 1
4
Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, a 'Wrap Cell Content' checkbox, and a 'Fetch rows' button. The result grid shows a single row of data with two columns: 'title' and 'director'. The data is:
title: Star Wars: Episode VII - The Force Awakens
director: Abrams

title	director
Star Wars: Episode VII - The Force Awakens	Abrams

H. Hent ut filmer med høyest inntekt for hver sjanger

```
SELECT movie.title, max(movie.gross), genre.name
FROM movie INNER JOIN movie_has_genre ON movie.id =
movie_has_genre.movie_id
INNER JOIN genre ON movie_has_genre.genre_id = genre.id
GROUP BY genre.name ;
```

The screenshot shows a database query editor with a query window and a results window. The query window contains the following SQL query:

```
1 • SELECT movie.title, max(movie.gross), genre.name
2 FROM movie INNER JOIN movie_has_genre ON movie.id = movie_has_genre.movie_id
3 INNER JOIN genre ON movie_has_genre.genre_id = genre.id
4 GROUP BY genre.name ;
5
```

The results window displays the query results in a grid. The grid has four columns: title, max(movie.gross), and name. The results are as follows:

	title	max(movie.gross)	name
▶	The Shawshank Redemption	858.37	Drama
	The Godfather	534.86	Crime
	The Dark Knight	936.66	Action
	Schindler's List	255.96	Biography
	Il buono, il brutto, il cattivo	184.21	Western
	Gisaengchung	434.04	Comedy
	The Lord of the Rings: The Return of the King	936.66	Adventure
	Sen to Chihiro no kamikakushi	608.58	Animation
	Psycho	232.91	Horror
	Se7en	293.51	Mystery
	Sunset Blvd.	10.46	Film-Noir
	Star Wars: Episode V - The Empire Strikes Back	760.51	Fantasy
	Sen to Chihiro no kamikakushi	435.11	Family
	Gisaengchung	335.45	Thriller
	Forrest Gump	659.33	Romance
	Inception	936.66	Sci-Fi
	Saving Private Ryan	216.54	War
	Whiplash	216.43	Music
	Anand	80.5	Musical
	Bacheha-Ye aseman	255.96	Sport
	Schindler's List	198.68	History

The results window also includes a toolbar with options like "Result Grid", "Filter Rows", "Export", and "Wrap Cell Content".

1. Hent ut hvilke filmer Christopher Nolan har regissert, som også Christian Bale har spilt i

```
SELECT movie.title
FROM movie INNER JOIN movie_has_actor ON movie.id =
movie_has_actor.movie_id
INNER JOIN actor ON movie_has_actor.actor_id = actor.id
INNER JOIN director ON movie.director_id = director.id
WHERE director.name = "Christopher Nolan" AND actor.name = "Christian
Bale";
```

The screenshot shows a database query editor with a toolbar at the top containing icons for file operations, execution, and search. The query text is as follows:

```
1 • SELECT movie.title
2 FROM movie INNER JOIN movie_has_actor ON movie.id = movie_has_actor.movie_id
3 INNER JOIN actor ON movie_has_actor.actor_id = actor.id
4 INNER JOIN director ON movie.director_id = director.id
5 WHERE director.name = "Christopher Nolan" AND actor.name = "Christian Bale" ;
6
```

Below the editor is a results panel with a toolbar for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The results are displayed in a table:

	title
▶	The Dark Knight
	The Prestige
	The Dark Knight Rises
	Batman Begins

J. Hent ut skuespillerne og antallet filmer de har spilt i, sortert synkende på antallet

```
SELECT actor.name, COUNT(movie_has_actor.actor_id) as movie_numbers
FROM movie INNER JOIN movie_has_actor ON movie.id =
movie_has_actor.movie_id
INNER JOIN actor ON movie_has_actor.actor_id = actor.id
GROUP BY actor.name
ORDER BY movie_numbers DESC
```

The screenshot shows a database query editor with the following SQL query:

```
1 • SELECT actor.name, COUNT(movie_has_actor.actor_id) as movie_numbers
2 FROM movie INNER JOIN movie_has_actor ON movie.id = movie_has_actor.movie_id
3 INNER JOIN actor ON movie_has_actor.actor_id = actor.id
4 GROUP BY actor.name
5 ORDER BY movie_numbers DESC
6
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has two columns: 'name' and 'movie_numbers'. The results are sorted in descending order of the number of movies.

name	movie_numbers
Robert De Niro	17
Tom Hanks	14
Al Pacino	13
Brad Pitt	12
Clint Eastwood	12
Christian Bale	11
Leonardo DiCaprio	11
Matt Damon	11
James Stewart	10
Michael Caine	9
Scarlett Johansson	9
Ethan Hawke	9
Humphrey Bogart	9
Johnny Depp	9
Bruce Willis	8
Harrison Ford	8
Denzel Washington	8
Aamir Khan	8
Morgan Freeman	7
Ian McKellen	7
Tom Cruise	7
Edward Norton	7
Christopher Plummer	7

