

1. Hent ut alle filmene sortert på 'year'

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'movies_imdb' selected. The central pane shows the SQL query:

```
1 SELECT title
2 FROM movie
3 ORDER BY year;
```

The right pane shows the 'Result Grid' with a list of movie titles, sorted by year. The bottom pane shows the 'Table: movie' structure:

Table: movie

Columns:

- id (int PK)
- title (varchar(1))
- year (year)
- runtime (int)
- imdb_rating (float)
- metascore (int)
- votes (int)
- gross (float)
- director_id (int)

```
SELECT title
FROM movie
ORDER BY year;
```

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

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'movies_imdb' selected. The central pane shows the SQL query:

```
1 SELECT title, metascore
2 FROM movie
3 WHERE metascore is not null;
```

The right pane shows the 'Result Grid' with a list of movie titles and their corresponding metascores. The bottom pane shows the 'Table: movie' structure:

Table: movie

Columns:

- id (int PK)
- title (varchar(1))
- year (year)
- runtime (int)
- imdb_rating (float)
- metascore (int)
- votes (int)
- gross (float)
- director_id (int)

```
SELECT title, metascore
FROM movie
WHERE metascore is not
null;
```

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

The screenshot shows a SQL query in a file named 'SQL File 4*'. The query is:

```
1 SELECT title, name
2 FROM movie, director
3 WHERE director_id = director.id
```

The result grid displays the following data:

title	name
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
The Hobbit: The Desolation...	Peter Jackson
The Hobbit: An Unexpecte...	Peter Jackson
Pulp Fiction	Quentin Tarantino
Django Unchained	Quentin Tarantino
Inglourious Basterds	Quentin Tarantino
Reservoir Dogs	Quentin Tarantino
Kill Bill: Vol. 1	Quentin Tarantino

On the left, the 'SCHEMAS' pane shows the 'movies_imdb' database with tables 'actor', 'director', 'genre', 'movie', 'movie_has_actor', and 'movie_has_genre'. The 'director' table is selected, showing columns 'id' (int PK) and 'name' (varchar(45)).

```
SELECT title, name
FROM movie, director
WHERE director_id =
director.id
```

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

The screenshot shows a SQL query in a file named 'SQL File 4*'. The query is:

```
1 SELECT name, count(name) as number_of_movies
2 FROM movie, director
3 WHERE director_id = director.id
4 GROUP BY name
5 ORDER BY number_of_movies DESC;
```

The result grid displays the following data:

name	number_of_movies
Alfred Hitchcock	14
Steven Spielberg	13
Hayao Miyazaki	11
Martin Scorsese	10
Akira Kurosawa	10
Stanley Kubrick	9
Billy Wilder	9
Woody Allen	9
Christopher Nolan	8
Quentin Tarantino	8
David Fincher	8
Clint Eastwood	8
Ingmar Bergman	7
Rob Reiner	7
Alfonso Cuarón	7
Sergio Leone	6
Ridley Scott	6
Charles Chaplin	6
Wes Anderson	6
Richard Linklater	6
Ethan Coen	6
Francis Ford Cop...	5
Sidney Lumet	5
Peter Jackson	5
Robert Zemeckis	5
Roman Polanski	5
James Cameron	5
Denis Villeneuve	5
Ron Howard	5
Andrei Tarkovsky	5

On the left, the 'SCHEMAS' pane shows the 'movies_imdb' database. The 'director' table is selected, showing columns 'id' (int PK) and 'name' (varchar(45)).

```
SELECT name,
count(name) as
number_of_movies
FROM movie, director
WHERE director_id =
director.id
GROUP BY name
ORDER BY
number_of_movies DESC;
```

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

The screenshot shows a SQL query in a window titled 'SQL File 4*'. The query is as follows:

```
1 SELECT name, count(name) as number_of_movies, round(sum(gross)) as gross_all_movies
2 FROM movie, director
3 WHERE director_id = director.id
4 GROUP BY name
5 ORDER BY gross_all_movies DESC;
```

The 'Result Grid' shows the following data:

name	number_of_movies	gross_all_movies
Steven Spielberg	13	2478
Joe Russo	4	2205
Christopher Nolan	8	1937
James Cameron	5	1748
Peter Jackson	5	1597
Abrams	3	1423
Lee Unkrich	4	1332
Robert Zemeckis	5	1049
David Yates	3	979
Brad Bird	3	893
Ridley Scott	6	767
Quentin Tarantino	8	727
James Gunn	2	723
David Fincher	8	713
Sam Mendes	4	698
Joss Whedon	2	649
Todd Phillips	2	613
Chris Columbus	2	603
Clint Eastwood	8	593
Alfonso Cuarón	7	583
Martin Scorsese	10	538
Gareth Edwards	1	532
James Mangold	4	517
Paul Greengrass	3	511
Bryan Singer	3	474
Chris Williams	2	471
Ron Howard	5	452
Josh Cooley	1	434
Rob Minkoff	1	423
Rob Reiner	7	381

The 'Table: director' information is shown below the result grid:

Columns:
 id int PK
 name varchar(45)

```
SELECT name,
count(name) as
number_of_movies,
round(sum(gross)) as
gross_all_movies
FROM movie, director
WHERE director_id =
director.id
GROUP BY name
ORDER BY
gross_all_movies DESC;
```

6. Hent ut alle filmer, samt tilhørende regissør og sjanger (genre)

The screenshot shows a SQL query in a window titled 'SQL File 4*'. The query is as follows:

```
1 SELECT title as movie, director.name as director, genre.name as genre
2 FROM movie, director, movie_has_genre, genre
3 WHERE director_id = director.id and movie_id = movie.id and genre_id = genre.id
```

The 'Result Grid' shows the following data:

movie	director	genre
The Shawshank Redemption	Frank Darabont	Drama
The Green Mile	Frank Darabont	Drama
The Green Mile	Frank Darabont	Crime
The Green Mile	Frank Darabont	Fantasy
The Godfather	Francis Ford Coppola	Drama
The Godfather	Francis Ford Coppola	Crime
The Godfather: Part II	Francis Ford Coppola	Drama
The Godfather: Part II	Francis Ford Coppola	Crime
Apocalypse Now	Francis Ford Coppola	Drama
Apocalypse Now	Francis Ford Coppola	Mystery
Apocalypse Now	Francis Ford Coppola	War
The Conversation	Francis Ford Coppola	Drama
The Conversation	Francis Ford Coppola	Mystery
The Conversation	Francis Ford Coppola	Thriller
The Godfather: Part III	Francis Ford Coppola	Drama
The Godfather: Part III	Francis Ford Coppola	Crime
The Dark Knight	Christopher Nolan	Drama
The Dark Knight	Christopher Nolan	Crime
The Dark Knight	Christopher Nolan	Action
Inception	Christopher Nolan	Action
Inception	Christopher Nolan	Advent...
Inception	Christopher Nolan	Sci-Fi
Interstellar	Christopher Nolan	Drama
Interstellar	Christopher Nolan	Advent...
Interstellar	Christopher Nolan	Sci-Fi
The Prestige	Christopher Nolan	Drama
The Prestige	Christopher Nolan	Mystery
The Prestige	Christopher Nolan	Sci-Fi
The Dark Knight Rises	Christopher Nolan	Action

The 'Table: movie_has_genre' information is shown below the result grid:

Columns:
 genre_id int PK
 movie_id int PK

```
SELECT title as movie,
director.name as
director, genre.name as
genre
FROM movie, director,
movie_has_genre,
genre
WHERE director_id =
director.id and
movie_id = movie.id
and genre_id = genre.id
```

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

The screenshot shows a SQL IDE interface. On the left is a 'SCHEMAS' pane with a tree view containing 'forelesninger', 'movies_imdb' (expanded), 'oblig1_1', 'oblig1_2', 'oblig2v23', and 'sys'. The 'movies_imdb' folder contains 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main editor shows a SQL query:

```
2 FROM movie, director
3 WHERE director_id = director.id
4 GROUP BY name
5 ORDER BY gross_all_movies DESC
6 limit 1
```

Below the query is a 'Result Grid' with the following data:

name	number_of_movies	gross_all_movies
Steven Spielberg	13	2478

At the bottom left, the 'Information' pane shows details for the table 'movie_has_genre':

Table: movie_has_genre

Columns: genre_id int PK, movie_id int PK

```
SELECT title as movie,
director.name as
director, genre.name as
genre
FROM movie, director,
movie_has_genre,
genre
WHERE director_id =
director.id and
movie_id = movie.id
and genre_id = genre.id
```

8. Hent ut filmer med høyest inntekt for hver sjanger

Startet med :

```
SELECT genre.name as Genre, title as Movie, gross as Gross
FROM movie, movie_has_genre, genre
WHERE movie_id = movie.id and genre_id = genre.id
```

Mangler å få gruppert per sjanger med høyest Gross, men finner ikke ut hvordan jeg kan gjøre dette..
Gi gjerne en kommentar på hvordan denne kunne vært løst.

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

```
SELECT title as Movie, director.name as Director, actor.name as Actor
FROM movie, director, movie_has_actor, actor
WHERE director_id = director.id and movie_id = movie.id and actor_id = actor.id
```

Fant ikke en måte å få filmer hvor både actor og director er spesifisert.
Gi gjerne en kommentar på hvordan denne kunne vært løst.

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

The screenshot shows a SQL IDE interface. On the left is a 'SCHEMAS' tree with a search filter. The main window displays a SQL query in a file named 'movie_has_actor'. The query is as follows:

```
1 SELECT name, count(name) as number_of_movies
2 FROM movie, movie_has_actor, actor
3 WHERE movie_id = movie.id and actor_id = actor.id
4 GROUP BY name
5 ORDER BY number_of_movies DESC;
```

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

name	number_of_movies
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
Humphrey Bogart	9
Ethan Hawke	9
Johnny Depp	9
Michael Caine	9
Scarlett Johansson	9
Aamir Khan	8
Harrison Ford	8
Bruce Willis	8
Denzel Washington	8
Morgan Freeman	7
Toshirō Mifune	7
Russell Crowe	7
Edward Norton	7
Robert Downey Jr.	7
Cary Grant	7
Mark Ruffalo	7
Jeff Bridges	7
Jake Gyllenhaal	7
Tom Cruise	7
John Wayne	7
Chris Evans	7

On the left side of the IDE, there is a 'Table: movie_has_genre' section showing its columns: 'genre_id' (int PK) and 'movie_id' (int PK).

```
SELECT name,
count(name) as
number_of_movies
FROM movie,
movie_has_actor,
actor
WHERE movie_id =
movie.id and
actor_id = actor.id
GROUP BY name
ORDER BY
number_of_movies
DESC;
```