

All Together Now (Revised)

The screenshot shows a web browser window with multiple tabs. The active tab is titled "All together now | SQL" and shows a URL "campus.datacamp.com/courses/intermediate-sql/sorting-and-grouping-4?ex=12". The page content is divided into a left sidebar and a main area. The sidebar contains an "Exercise" section titled "All together now" with instructions for a SQL query. The main area features a SQL editor with the query: "SELECT release_year FROM films WHERE release_year > 1990 GROUP BY release_year;". Below the editor is a "query result" section showing "No query executed yet...". The browser's taskbar at the bottom shows various application icons and the system clock indicating 15:28 on 28/11/2024.

It's time to use much of what you've learned in one query! This is good preparation for using SQL in the real world where you'll often be asked to write more complex queries since some of the basic queries can be answered by playing around in spreadsheet applications.

In this exercise, you'll write a query that returns the average budget and gross earnings for films each year after 1990 if the average budget is greater than 60 million.

Answer for Each Instruction

Instruction 1: Select the `release_year` for each film in the `films` table, filter for records released after 1990, and group by `release_year`.

SQL Query:

```
SELECT release_year
```

```
FROM films
```

```
WHERE release_year > 1990
```

```
GROUP BY release_year;
```

Instruction 2: Modify the query to include the average budget aliased as avg_budget and average gross aliased as avg_gross for the results we have so far.

SQL Query:

```
SELECT release_year,  
       AVG(budget) AS avg_budget,  
       AVG(gross) AS avg_gross  
FROM films  
WHERE release_year > 1990  
GROUP BY release_year;
```

Instruction 3: Modify the query once more so that only years with an average budget of greater than 60 million are included.

SQL Query:

```
SELECT release_year,  
       AVG(budget) AS avg_budget,  
       AVG(gross) AS avg_gross  
FROM films  
WHERE release_year > 1990  
GROUP BY release_year  
HAVING AVG(budget) > 60000000;
```

Instruction 4: Finally, order the results from the highest average gross and limit to one.

SQL Query:

```
SELECT release_year,  
       AVG(budget) AS avg_budget,  
       AVG(gross) AS avg_gross  
FROM films  
WHERE release_year > 1990  
GROUP BY release_year  
HAVING AVG(budget) > 60000000  
ORDER BY avg_gross DESC  
LIMIT 1;
```

