Queries with expressions

In addition to querying and referencing raw column data with SQL, you can also use *expressions* to write more complex logic on column values in a query. These expressions can use mathematical and string functions along with basic arithmetic to transform values when the query is executed, as shown in this physics example.

Example query with expressions

```
SELECT particle_speed / 2.0 AS half_particle_speed
FROM physics_data
WHERE ABS(particle_position) * 10.0 > 500;
```

Each database has its own supported set of mathematical, string, and date functions that can be used in a query, which you can find in their own respective docs.
The use of expressions can save time and extra post-processing of the result data, but can also make the query harder to read, so we recommend that when expressions are used in the SELECT part of the query, that they are also given a descriptive *alias* using the AS keyword.

Select query with expression aliases

```
SELECT col_expression AS expr_description, …
FROM mytable;
```

In addition to expressions, regular columns and even tables can also have aliases to make them easier to reference in the output and as a part of simplifying more complex queries.

Example query with both column and table name aliases

```
SELECT column AS better_column_name, …
FROM a_long_widgets_table_name AS mywidgets
INNER JOIN widget_sales
  ON mywidgets.id = widget_sales.widget_id;
```

Exercise

You are going to have to use expressions to transform the BoxOffice data into something easier to understand for the tasks below.

Exercise - Tasks

1. List all movies and their combined sales in **millions** of dollars
2. List all movies and their ratings **in percent**
3. List all movies that were released on even number years