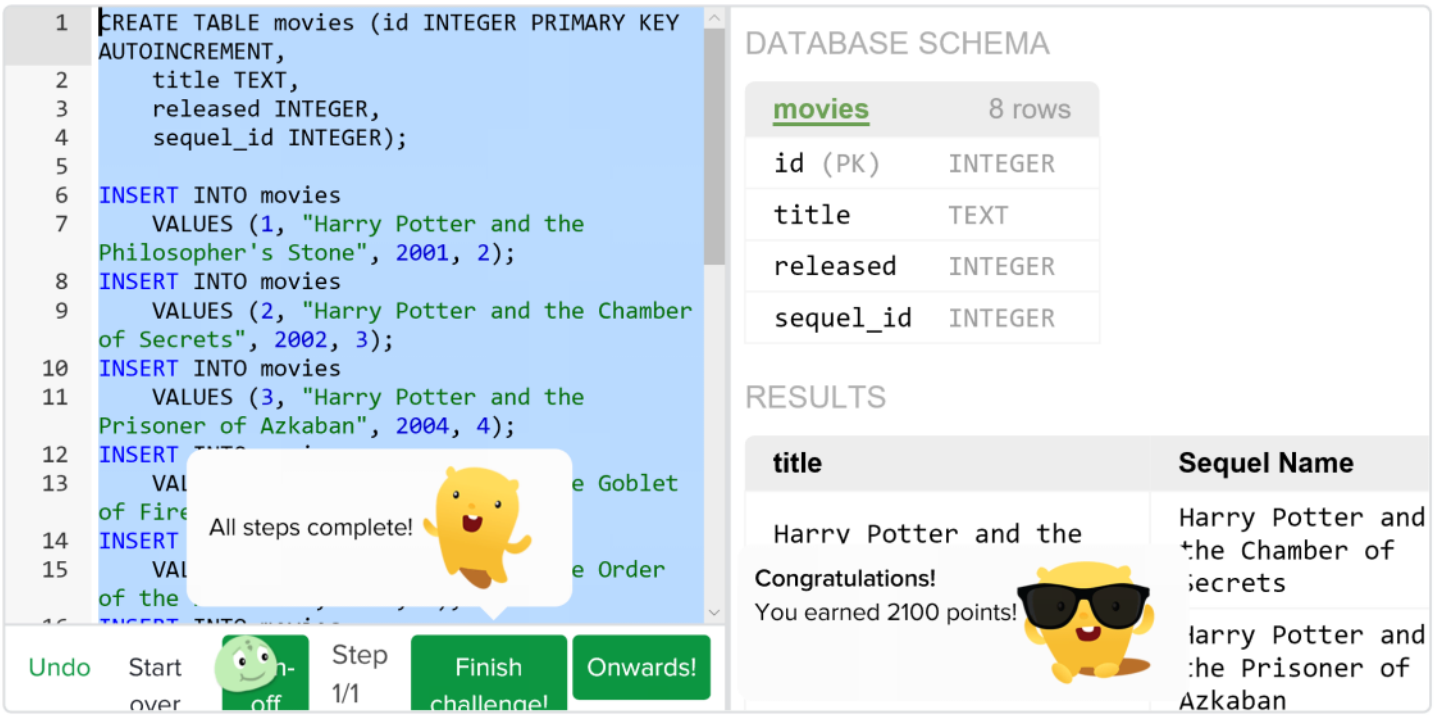
Challenge 12: Sequels in SQL

Friday, April 7, 2017 12:55 PM



Screen clipping taken: 4/22/2017 11:30 PM

Step 1 We've created a table with all the 'Harry Potter' movies, with a sequel_id column that matches the id of the sequel for each movie. Issue a SELECT that will show the title of each movie next to its sequel's title (or NULL if it doesn't have a sequel).	CREATE TABLE movies (id INTEGER PRIMARY KEY AUTOINCREMENT, title TEXT, released INTEGER, sequel_id INTEGER); INSERT INTO movies VALUES (1, "Harry Potter and the Philosopher's Stone", 2001, 2); INSERT INTO movies VALUES (2, "Harry Potter and the Chamber of Secrets", 2002, 3); INSERT INTO movies VALUES (3, "Harry Potter and the Prisoner of Azkaban", 2004, 4); INSERT INTO movies VALUES (4, "Harry Potter and the Goblet of Fire", 2005, 5); INSERT INTO movies VALUES (5, "Harry Potter and the Order of the Phoenix ", 2007, 6); INSERT INTO movies VALUES (6, "Harry Potter and the Half-Blood Prince", 2009, 7); INSERT INTO movies VALUES (7, "Harry Potter and the Deathly Hallows – Part 1", 2010, 8); INSERT INTO movies VALUES (8, "Harry Potter and the Deathly Hallows – Part 2", 2011, NULL);

```
/* SELECT movies.title, sequels.title
FROM movies
LEFT OUTER JOIN movies sequels
ON movies.id = sequels.sequel_id;
*/

SELECT movies.title, sequel.title AS "Sequel Name"
FROM movies
LEFT OUTER JOIN movies AS sequel
ON sequel.id = movies.sequel_id;

/* SELECT movies.title, sequels.title as "Sequel
Name" FROM movies
JOIN movies sequels
ON movies.id = sequels.sequel_id;
ORDER BY movies.released;
*/
```