

Names: Urja Soni, Chance Potter

Implementation Report:

We have only implemented code to answer questions related to movies database.

We used the parsed tree from the part 1, and to make it easier to reach each element, we saved them into an array called tags. In order to determine if the given question is Yes/No question or Wh question, we had to look at the POS tags generated by the code. We noticed that All the yes/no question either started with “SQ” or “S” and the wh questions started with “SBRQ”. Therefore, we had condition which will check the first second element of the array and if it was “SQ”/ “S” then we called a function called yesNoQuestion() otherwise call the whQuestion() function.

In yesNoQuestion():

Since it is a yes no question, we sent the select line equal to Count (*), which will return the number of occurrences certain things happens in a database. The code uses different arrays for person and movie. And based on the POS tags on the tree, if there were two NNP in a row, we didn't add them into nous as they would most likely be a movie. Similarly, the code iterated through entire array to find noun, person or a year. If the tags included “CD” that would be for year.

If one of the noun was “Oscar” the select line gets changed to the table Oscar. We have a set variables for person, movie, a_type(for award type) and year. Depending what the question is asking, we are using those variables for specific queries.

In whQuestion():

As mentioned above, the code iterates through the array called tags, and checking whether the question is looking for movie name, actor/actress name or year. Based on the question Boolean value would either be true or false. This also gets the name of the movie, person or the year if they are present in the question.

If getyear value equals to true then select like in from table Oscar, if the directed value equals to true then select line is set to table Person, if findMovie value is true then select line is

from the table movie. As mentioned previously, we built different queries and used values from the questions to build them.

Initially we ran the queries on sqlite separately since we didn't connect to database in the beginning. Once we had all the queries working, we then wrote the code to connect to the database. For Wh questions, the answer would simply be what the queries would return. However, in order for yes/no questions there is additional code which needed to be written. If the return value from the query equals to 0, that means that no answer was found on the database, therefore the answer to that question would be "No". If the return value from the query was greater than 0, that means that there was at least one data retrieved which fulfilled the condition of the question and therefore the answer to the question would be "Yes".

Once all the queries have been retrieved successfully, the code goes about and closes the database.

Sample Output:

<QUESTION>

Is Kubrick a director?

<QUERY>

Select Count(*)

FROM Person P INNER JOIN Director D on P.id = D.director_id

WHERE P.name LIKE '%Kubrick%';

<ANSWER>

Yes

<QUESTION>

Is Mighty Aphrodite by Allen?

<QUERY>

Select Count(*)

FROM Person P INNER JOIN Movie M on P.id = M.id

WHERE P.name LIKE '%Allen%' AND M.name LIKE '%Mighty%';

<ANSWER>

No

<QUESTION>

Was Loren born in Italy?

<QUERY>

Select Count(*)

FROM Person P

WHERE P.name LIKE '%Loren%' AND pob LIKE '%Italy%';

<ANSWER>

Yes

<QUESTION>

Was Birdman the best movie in 2015?

<QUERY>

Select Count(*)

FROM movie INNER JOIN oscar on movie.id=oscar.movie_id

WHERE movie.name LIKE '%Birdman%' AND type LIKE '%BEST-PICTURE%' AND
oscar.year LIKE '%2015%';

<ANSWER>

Yes

<QUESTION>

Did Neeson star in Schindler's List?

<QUERY>

Select Count(*)

FROM Person P INNER JOIN Movie M on P.id = M.id

WHERE P.name LIKE '%Neeson%' AND M.name LIKE '%List%';

<ANSWER>

No

<QUESTION>

Did Swank win the oscar in 2000?

<QUERY>

Select Count(*)

FROM oscar INNER JOIN person on oscar.person_id=person.id INNER JOIN actor on
person.id=actor.actor_id

WHERE person.name LIKE '%Swank%' AND oscar.year LIKE '%2000%';

<ANSWER>

Yes

<QUESTION>

Did a French actor win the oscar in 2012?

<QUERY>

Select Count(*)

FROM oscar INNER JOIN person on oscar.person_id=person.id

WHERE pob LIKE '%France%' AND oscar.year LIKE '%2012%';

<ANSWER>

Yes

<QUESTION>

Did a movie with Neeson win the oscar for best film?

<QUERY>

Select Count(*)

FROM oscar INNER JOIN person on oscar.person_id=person.id

WHERE person.name LIKE '%Neeson%' AND type LIKE '%BEST_PICTURE%';

<ANSWER>

No

<QUESTION>

Who directed Schindler's List?

<QUERY>

SELECT P.name FROM PERSON P

INNER JOIN Director D on P.id = D.director_id INNER JOIN MOVIE M on D.movie_id = M.id

WHERE M.name LIKE '%Schindler%';

<ANSWER>

Steven Spielberg

<QUESTION>

Who won the oscar for best actor in 2005?

<QUERY>

```
SELECT DISTINCT P.name FROM PERSON P
INNER JOIN ACTOR A on P.id = A.actor_id INNER JOIN OSCAR O on P.id = O.person_id
WHERE O.year = 2005 AND O.type = "BEST-ACTOR";
```

<ANSWER>

Jamie Foxx

<QUESTION>

Who directed the best movie in 2010?

<QUERY>

```
SELECT P.name FROM PERSON P
INNER JOIN Director D on P.id = D.director_id INNER JOIN OSCAR O on D.movie_id =
O.movie_id
WHERE O.type = "BEST-PICTURE" and O.year = 2010;
```

<ANSWER>

Kathryn Bigelow

<QUESTION>

Which American actress won the oscar in 2012?

<QUERY>

```
SELECT DISTINCT P.name FROM PERSON P
```

INNER JOIN ACTOR A on P.id = A.actor_id INNER JOIN OSCAR O on P.id = O.person_id
WHERE O.year = 2012 AND O.type = "BEST-ACTRESS";

<ANSWER>

Meryl Streep

<QUESTION>

Which movie won the oscar in 2000?

<QUERY>

SELECT M.name FROM MOVIE M
INNER JOIN OSCAR O ON M.id = O.movie_id
WHERE O.type = "BEST-PICTURE" AND O.year = 2000;

<ANSWER>

American Beauty

<QUESTION>

When did Blanchett win an oscar for best actress?

<QUERY>

SELECT O.year FROM OSCAR O
INNER JOIN PERSON P on P.id = O.person_id
WHERE O.type = "BEST-ACTRESS" and P.name LIKE '%Blanchett%';

<ANSWER>

2014