1. [10pts] For all detectors whose first name is 'Ellen', list their user id, the date and time when they became a detector, and their email address. Expected result row(s): 2
   a) [7pts] SQL Query:
   
   ```sql
   SELECT userid, detector_since, email
   FROM Users natural join Detector
   WHERE name_first = 'Ellen'
   ```
   
   b) [3pts] Result:
   
<table>
<thead>
<tr>
<th>userid</th>
<th>detector_since</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>2011-09-21 14:47:12</td>
<td><a href="mailto:pedro@aol.com">pedro@aol.com</a></td>
</tr>
<tr>
<td>20</td>
<td>2004-08-22 19:47:12</td>
<td><a href="mailto:rob@gmail.com">rob@gmail.com</a></td>
</tr>
</tbody>
</table>

2. [10pts] For those articles that have gotten at least one “like” score of 0 (i.e., articles that have received 0 as their “like” score at least once), print their user id, their article id and the date and time when the article was posted. Expected result row(s): 2 (Ignore last row with null values)
   a) [7pts] SQL Query:
   
   ```sql
   SELECT userid, articleid, posting_datetime
   FROM Article
   WHERE EXISTS (SELECT *
   FROM Has_read
   WHERE like_score = 0 AND Article.userid = article_userid
   AND Article.articleid = article_articleid)
   ```
   
   b) [3pts] Result:
   
<table>
<thead>
<tr>
<th>userid</th>
<th>articleid</th>
<th>posting_datetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>2016-10-24 02:01:12</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2015-07-09 01:00:00</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
3. [10pts] Print title and quality articles associated with the topic 'Politics' that have received the current highest “like” score. Expected result row(s): 1

a) [7pts] SQL Query:

```sql
SELECT A.title, A.quality
FROM Article as A NATURAL JOIN Topic, Has_read as H
WHERE A.userid = H.article_userid AND A.articleid = H.article_articleid AND topic = 'Politics'
    AND like_score = (SELECT MAX(like_score) FROM Has_read, Topic T
                        WHERE topic = "Politics" AND article_articleid = articleid
                        AND article_userid = T.userid)
```

b) [3pts] Result:

```
| title      | quality      |
+ ---------- + ------------ |
| Inauguration of President Trump | Clean       |
```

4. [10pts] Print the publisher name, user id and article id of articles that were published by a poster working for a publisher that is based in either 'San Francisco' or 'Irvine'. Expected result row(s): 4

a) [7pts] SQL Query:

```sql
SELECT name, userid, articleid
FROM Publisher NATURAL JOIN Poster NATURAL JOIN Article
WHERE addr_city= 'San Francisco'
UNION
SELECT name, userid, articleid
FROM Publisher NATURAL JOIN Poster NATURAL JOIN Article
WHERE addr_city= 'Irvine'
```

b) [3pts] Result:

```
| name      | userid | articleid |
+ --------- + ------- + ---------- |
| Angel News | 6      | 1         |
| Irvine Post | 3      | 1         |
```
5. [10pts] Print the publisher id, name and website of publishers that have a poster holding a degree from 'University of California, Irvine' and a poster holding a degree from 'Harvard University'. (This could be different posters or one poster who went to both schools.) Expected result row(s): 1

a) [7pts] SQL Query:

```
SELECT publisherid, name, website
FROM Publisher P1 NATURAL JOIN Poster NATURAL JOIN Users NATURAL JOIN Degree as D1
WHERE D1.school = 'University of California, Irvine' AND EXISTS (SELECT * FROM Publisher P2 NATURAL JOIN Poster NATURAL JOIN Users NATURAL JOIN Degree as D2 WHERE p1.publisherid = p2.publisherid AND D2.school = 'Harvard University')
```

b) [3pts] Result:

```
<table>
<thead>
<tr>
<th>publisherid</th>
<th>name</th>
<th>website</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Irvine Post</td>
<td><a href="http://www.irvinepost.com">www.irvinepost.com</a></td>
</tr>
</tbody>
</table>
```

6. [10pts] Print the user id, first name, last name, and job title of posters who have recommended an article that they have not read. Expected result row(s): 2

a) [7pts] SQL Query:

```
SELECT userid, name_first, name_last, jobtitle
FROM Users as U NATURAL JOIN Poster
WHERE EXISTS(SELECT * FROM Recommendation as R
        WHERE R.from_userid = U.userid
        AND NOT EXISTS(SELECT * FROM Has_read H
            WHERE H.userid = U.userid
            AND H.article_userid = R.article_userid
            AND H.article_articleid = R.article_articleid))
```
b) [3pts] Result:

<table>
<thead>
<tr>
<th>userid</th>
<th>name_first</th>
<th>name_last</th>
<th>jobtitle</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Andrew</td>
<td>James</td>
<td>Blogger</td>
</tr>
<tr>
<td>4</td>
<td>Richard</td>
<td>Cooper</td>
<td>Sports Writer</td>
</tr>
</tbody>
</table>

7. [10pts] List, in descending order, the 3 most avid readers (showing their user ids and number-of-articles-read counts) among users. (Hint: Avid readers are the ones that read the most.)

Expected result row(s): 3

a) [7pts] SQL Query:

```sql
SELECT userid, COUNT(*) FROM Has_read
GROUP BY userid ORDER BY COUNT(*) DESC
LIMIT 3
```

b) [3pts] Result:

<table>
<thead>
<tr>
<th>userid</th>
<th>COUNT(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
</tbody>
</table>

8. [10pts] Due to a bug in the application, some data are now dirty and need to be cleaned. Find the user id, article id, and title of articles that have quality value of 'Clean' but that deserve a lesser quality value. (Hint: Look at item 13 in HW 1 for the quality criterion.) Expected result row(s): 1 (Ignore last row with null values)

a) [7pts] SQL Query:

```sql
SELECT userid, articleid, title FROM Article
WHERE quality = 'Clean'
AND (userid, articleid) IN(SELECT article_userid, article_articleid
FROM Reports
GROUP BY article_userid, article_articleid
HAVING COUNT(*) > 5)
```
9. [10pts] Find the fastest user(s) in terms of posting articles after signing up on the website (i.e., based on the time difference between their joining time and posting time) and print their user id and account. Expected result row(s): 1

a) [7pts] SQL Query:
```
SELECT userid, account
FROM Users NATURAL JOIN Article
ORDER BY TIMEDIFF(posting_datetime,user_since)
LIMIT 1 /* Q9 */
```

b) [3pts] Result:
```
+ ----------- + ------------ +
| userid      | articleid      | title      |
+ ----------- + ------------ +
| 3           | 2              | iPhone 8 to be announced in September |
| NULL        | NULL           | NULL       |
+ ----------- + ------------ +
```

10. [10pts] Print the user id, article id, and title of articles that have been reported by all detectors. Expected result row(s): 1 (Ignore last row with null values)

a) [7pts] SQL Query:
```
SELECT userid, articleid, title FROM Article A
WHERE NOT EXISTS (SELECT userid FROM Detector D
    WHERE NOT EXISTS (SELECT userid FROM Reports WHERE article_userid = A.userid
        AND article_articleid = A.articleid
        AND D.userid = userid)) /* Q10 */
```
b) [3pts] Result:

+ ----------- + -------------- + ---------- +
| userid      | articleid      | title      |
+ ----------- + -------------- + ---------- +
| 6           | 1              | 2009 Warmest Year on Record |
| NULL        | NULL           | NULL       |
+ ----------- + -------------- + ---------- +