```
/* query 1
   search the steam game released after 2020, with Action tag
 */
SELECT name FROM GAME G
JOIN DESCRIPTION D ON G.app_id = D.app_id
WHERE D.genre like 'Action' AND D.release_dt > DATE("2000-00-00")

/* query 2
   search the description of the game with name "Day of Defeat"
 */
SELECT *
FROM GAME G
JOIN DESCRIPTION
WHERE name like '%Day of Defeat%'

/* query 3
   show the game and the number of its recommendation in ascending
order
 */
SELECT name, COUNT(*) as count
FROM (SELECT g.app_id as appid, g.name, r.recommended
FROM GAME g INNER JOIN REVIEW r
ON g.app_id = r.app_id) as tb
WHERE recommended like 'True'
GROUP BY name
ORDER BY count

/*
 query 4:
 return all the users who have a review on the game they do not own
*/
SELECT user_id
FROM USER u JOIN REVIEW r ON u.user_id = r.author_id
WHERE EXISTS
(
  SELECT * FROM OWN_GAME o
  WHERE o.a
);

/* query 5
   Search for top 10 games which are most played by users from
Philadelphia
 */
SELECT name, COUNT(*) AS philadelphia_users
FROM GAME g JOIN OWN_GAME o ON g.app_id = o.app_id
JOIN USER u ON o.user_id = u.user_id
WHERE location = "Philadelphia"
GROUP BY name, g.app_id
ORDER BY COUNT(*)
LIMIT 10

/* query 6
   Search for games that positive_ratings are 10 times higher than
   negative ratings and allow windows platform. Also the release
```

```
date
   of the game should be later than 2000. Finally, sort the result
by
   number of positive_ratings.
 */
SELECT d.app_id AS app_id, name, positive_ratings
FROM DESCRIPTION d JOIN GAME g on d.app_id = g.app_id
WHERE positive_ratings > 10 * negative_ratings
AND platform LIKE '%windows%'
AND release_dt >= DATE("2000-01-01")
ORDER BY positive_ratings DESC

/* query 7
   Search for games that users in Boston play and are released
   after 2000.Then return the name of the game and the rate
   of positive reviews
 */
 WITH temp(app_id) AS(
 SELECT app_id FROM OWN_GAME o JOIN USER u ON o.user_id = u.user_id
 )
 SELECT name, SUM(IF(recommended = True, 1, 0)) / COUNT(recommended)
AS rate_pos
 FROM GAME g JOIN DESCRIPTION d ON g.app_id = d.app_id
JOIN REVIEW r ON g.app_id = r.app_id JOIN temp t ON g.app_id =
t.app_id
WHERE release_dt >= DATE("2000-01-01")
GROUP BY name
HAVING COUNT(DISTINCT recommended) > 0
ORDER BY rate_pos

/*
 query 8:
 search the game with app_id, name, short_description, with the
number of positive review
 */

SELECT D.app_id, C.name, C.positivereview, D.short_description
FROM DESCRIPTION D JOIN (SELECT app_id, name, positivereview
                       FROM GAME G
                                JOIN (SELECT app_name, count(review)
AS positivereview
                                      FROM REVIEW
                                      GROUP BY app_id, app_name,
recommended
                                      HAVING recommended like
'True') AS R
                                     ON G.name = R.app_name) AS C
ON D.app_id = C.app_id;

/*
 query 9:
 return all the users who have a review on the game they do not own
*/
SELECT user_id
```

```
FROM USER u JOIN REVIEW r ON u.user_id = r.author_id
WHERE EXISTS
(
  SELECT * FROM OWN_GAME o
  WHERE o.a
);

/*
 query 9
 return the name and number of reviews of the game which is owned by
people from Beijing and was released after 1990
*/

WITH temp(name) AS(
SELECT name FROM GAME g JOIN OWN_GAME o ON g.app_id = o.app_id
JOIN USER u ON o.user_id = u.user_id
WHERE location <> "Beijing"
)
SELECT name, num_review
FROM
(
SELECT name, COUNT(DISTINCT review_id) AS num_review
FROM GAME g LEFT JOIN REVIEW r ON g.app_id = r.app_id
LEFT JOIN DESCRIPTION d ON g.app_id= d.app_id
WHERE year(release_dt) >= year(1990)
GROUP BY name) t
WHERE name NOT IN
(
    SELECT * FROM temp
)

/*
 query 10:
 select the name and a random review of the game where the game
genre with Action
 */
SELECT C.name, C.review
FROM DESCRIPTION D JOIN (SELECT app_id, name, review
                       FROM GAME G
                                JOIN (SELECT app_name, review
                                      FROM REVIEW
                                      GROUP BY app_id, app_name) AS
R
                                     ON G.name = R.app_name) AS C
ON D.app_id = C.app_id
GROUP BY name, genre
HAVING genre like '%Action%';
```