Problem 1: List the USER_ID, FNAME, LNAME of friends of the user whose id is 5.

Query:

```
SELECT DISTINCT U.USER_ID,U.FNAME,U.LNAME
FROM USERS U INNER JOIN FRIENDSHIPS F ON (U.USER_ID=F.INVITEE_ID OR
U.USER_ID = F.INVITER_ID)
WHERE (F.INVITER_ID=5 OR F.INVITEE_ID=5)
AND F.STATUS=1
AND U.USER_ID<>5;
```

Explanation:

The above query joins USERS table with FRIENDSHIPS table on U.USER_ID=F.INVITEE_ID or U.USER_ID=F.INVITER_ID through INNER JOIN and outputs the USER_ID, FNAME and LNAME of Friends of user whose id is 5 distinctly (considering the case where the user whose id is 5 is present either as the inviter or the invitee and omitting repeating values) and also omits the conditions where the user whose id is 5 can be a friend with himself. Thus, satisfying the question's requirements.

Output Obtained:

```
+---------+----------+----------+
| USER_ID | FNAME    | LNAME    |
+---------+----------+----------+
|       1 | Harry    | Prince   |
|       3 | Donald   | Trump    |
|       7 | Vladimir | Putin    |
|       8 | Diana    | Princess |
|      16 | Willa    | Holland  |
|      17 | Addison  | Timlin   |
+---------+----------+----------+
6 rows in set (0.05 sec)

MySQL [SHOME]>
```

Problem 2: List the USER_ID, FNAME, LNAME, GENDER, DAY_OF_BIRTH of all pending friends (users that invited this user as friend but have not yet been accepted) of the user whose id is 1.

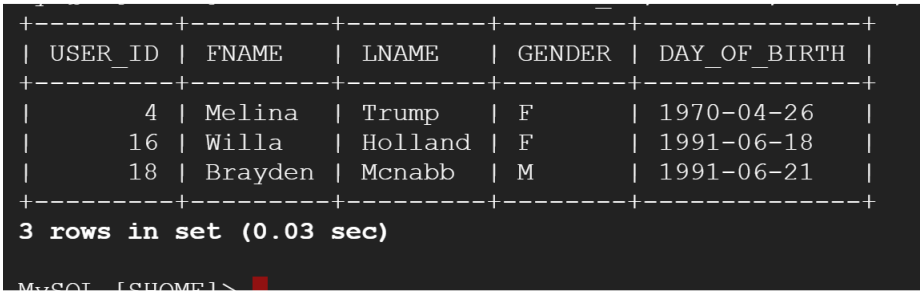
Query:

```
SELECT DISTINCT U.USER_ID,U.FNAME,U.LNAME,U.GENDER,U.DATE_OF_BIRTH AS
DAY_OF_BIRTH
FROM USERS U,FRIENDSHIPS F
WHERE U.USER_ID=F.INVITER_ID
AND F.INVITEE_ID=1
AND F.INVITER_ID<>1
AND F.STATUS=0;
```

Explanation:

The above query joins USERS table with FRIENDSHIPS table on the condition USERS.USER_ID=FRIENDSHIPS.INVITER_ID and returns the USER_ID, FNAME, LNAME, GENDER and DATE_OF_BIRTH AS DAY_OF_BIRTH distinctly (considering the case where the user whose id 1 cannot invite himself/herself since a user cannot be a friend with oneself and omitting repeating values) of all the users who invited user whose id is 1 but the friendships request has not yet been accepted.

Output Obtained:



```
+---------+---------+---------+--------+--------------+
| USER_ID | FNAME   | LNAME   | GENDER | DAY_OF_BIRTH |
+---------+---------+---------+--------+--------------+
|       4 | Melina  | Trump   | F      | 1970-04-26   |
|      16 | Willa   | Holland | F      | 1991-06-18   |
|      18 | Brayden | Mcnabb  | M      | 1991-06-21   |
+---------+---------+---------+--------+--------------+
3 rows in set (0.03 sec)

MySQL [SHOME]>
```

Problem 3: List the USER_ID, FNAME, LNAME of female mutual friends between users 1 and 2.

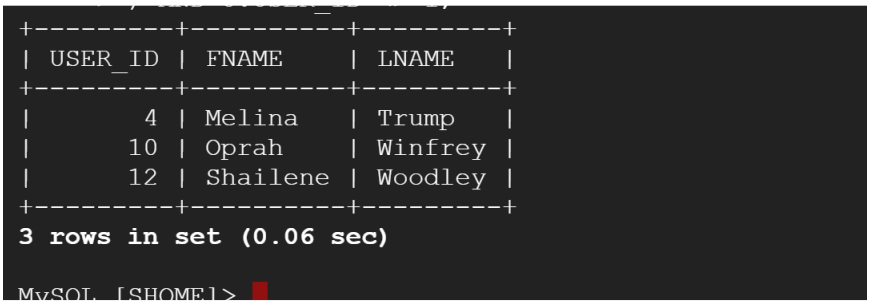
Query:

```
SELECT DISTINCT U.USER_ID,U.FNAME,U.LNAME
FROM USERS U INNER JOIN FRIENDSHIPS F ON (U.USER_ID=F.INVITEE_ID OR
U.USER_ID = F.INVITER_ID)
WHERE (F.INVITER_ID=1 OR F.INVITEE_ID=1)
AND F.STATUS=1
AND U.GENDER='F'
AND U.USER_ID IN
(
SELECT DISTINCT U.USER_ID
FROM USERS U INNER JOIN FRIENDSHIPS F ON (U.USER_ID=F.INVITEE_ID OR
U.USER_ID = F.INVITER_ID)
WHERE (F.INVITER_ID=2 OR F.INVITEE_ID=2)
AND F.STATUS=1
AND U.GENDER='F'
AND U.USER_ID<>2
) AND U.USER_ID <> 1;
```

Explanation:

The above query returns the USER_ID, FNAME, LNAME of the users who are female mutual friends of users whose id's are 1 and 2 distinctively (omitting repeating values) by joining USERS and FRIENDSHIPS tables on the condition USER.USER_ID=FRIENDSHIPS.INVITER_ID and USER.USER_ID =FRIENDSHIPS.INVITEE_ID (ensuring that the user whose id is 1 or 2 can be either an INVITER or an INVITEE) and also ensures that the condition that the user whose id is 1 or 2 cannot be considered (since a user cannot be friend with himself/herself). Thus, satisfying the requirement's of the question in hand.

Output Obtained:



```
+---------+----------+---------+
| USER_ID | FNAME    | LNAME   |
+---------+----------+---------+
|       4 | Melina   | Trump   |
|      10 | Oprah    | Winfrey |
|      12 | Shailene | Woodley |
+---------+----------+---------+
3 rows in set (0.06 sec)

MySQL [SHOME]>
```

Problem 4: List the USER_ID of female users who were born after '1990-12-20' and commented on posts of USER_ID=10. Show their friends count in a separate column.

Query:

```
SELECT U.USER_ID,COUNT(F.STATUS) AS FRIENDS_COUNT
FROM USERS U
INNER JOIN COMMENTS C ON C.USER_ID=U.USER_ID
INNER JOIN POSTS P ON P.POST_ID=C.POST_ID
LEFT OUTER JOIN FRIENDSHIPS F ON ((U.USER_ID=F.INVITEE_ID OR U.USER_ID =
F.INVITER_ID) AND F.STATUS=1)
WHERE U.GENDER='F'
AND P.USER_ID=10
AND U.DATE_OF_BIRTH > '1990-12-20'
GROUP BY U.USER_ID;
```

Explanation:

The above query returns the USER_ID, Friendships counts of female users born after '1990-12-20' and commented on post of user whose id is 10 by joining the tables USERS, COMMENTS through inner join on the condition that COMMENTS.USER_ID=USERS.USER_ID and the tables POSTS, COMMENTS through inner join on the condition that POSTS.POST_ID=COMMENTS.POST_ID and the above resultant tables with FRIENDSHIPS table on the condition that USERS.USER_ID = FRIENDSHIPS.INVITEE_ID through left outer join and provides the COUNT (aggregate function) of friends of each of the above users by the condition USERS.USER_ID = FRIENDSHIPS.INVITEE_ID or USERS.USER_ID = FRIENDSHIPS.INVITER_ID (ensuring that the friends of the commenters can either be invitee or inviter and also ensures the friendship is in accepted status by checking that the FRIENDSHIPS.STATUS=1 for all such friendships. Thus, satisfying the question's requirements.

Output Obtained:

```
+---------+---------------+
| USER_ID | FRIENDS_COUNT |
+---------+---------------+
|      11 |             0 |
|      12 |             5 |
|      16 |             2 |
|      17 |             2 |
+---------+---------------+
4 rows in set (0.04 sec)

MySQL [SHOME]>
```

Problem 5: List the user ids of up to 10 pairs of users where their distances are exactly 2 (i.e., they do not have direct friendship and share at least one common friend).

Query:

```
SELECT T1.USER1,T1.USER2
FROM (SELECT U1.USER_ID USER1,U2.USER_ID USER2 FROM USERS U1 INNER JOIN
USERS U2 ON U1.USER_ID>U2.USER_ID) T1,( SELECT U.USER_ID,F.* FROM USERS U
INNER JOIN FRIENDSHIPS F ON (U.USER_ID=F.INVITEE_ID OR U.USER_ID =
F.INVITER_ID)) T2
WHERE (T2.INVITER_ID= T1.USER1 OR T2.INVITEE_ID=T1.USER1)
AND T2.STATUS=1 AND T2.USER_ID IN
(
SELECT DISTINCT U.USER_ID
FROM USERS U INNER JOIN FRIENDSHIPS F ON (U.USER_ID=F.INVITEE_ID OR
U.USER_ID = F.INVITER_ID)
WHERE (F.INVITER_ID=T1.USER2 OR F.INVITEE_ID=T1.USER2)
AND F.STATUS=1
AND U.USER_ID<>T1.USER2
) AND T2.USER_ID <> T1.USER1
GROUP BY T1.USER1,T1.USER2
HAVING COUNT(*)>=1
LIMIT 10;
```

Explanation:

The above query lists the USER_ID of upto 10 pairs of users who do not share direct friendships but has friendships distance of exactly 2 levels and shares atleast 1 common friends by joining the USERS and FRIENDSHIPS table based on above conditions. Thus satisfying the question's requirements.

Output Obtained:

```
+-------+-------+
| USER1 | USER2 |
+-------+-------+
|     2 |     1 |
|     3 |     1 |
|     3 |     2 |
|     4 |     3 |
|     5 |     1 |
|     5 |     2 |
|     5 |     3 |
|     5 |     4 |
|     6 |     1 |
|     6 |     2 |
+-------+-------+
10 rows in set (0.04 sec)
```

Problem 6: List the user ids of up to 10 pairs of users where one is male and the other is female, and each comments on the other's posts at least 5 times.

Query:

```
SELECT DISTINCT USER1,USER2
FROM (SELECT U1.USER_ID USER1,U2.USER_ID USER2 FROM POSTS P INNER JOIN
COMMENTS C ON (P.POST_ID=C.POST_ID AND P.USER_ID <> C.USER_ID) RIGHT OUTER
JOIN USERS U1 ON P.USER_ID=U1.USER_ID LEFT OUTER JOIN USERS U2 ON
C.USER_ID=U2.USER_ID
WHERE U1.GENDER <> U2.GENDER
GROUP BY U1.USER_ID,U2.USER_ID
HAVING COUNT(*)>=5) T1
WHERE T1.USER1>T1.USER2
AND EXISTS (SELECT * FROM (SELECT U1.USER_ID AS USER1,U2.USER_ID USER2
FROM POSTS P INNER JOIN COMMENTS C
ON (P.POST_ID=C.POST_ID AND P.USER_ID <> C.USER_ID) RIGHT OUTER JOIN USERS
U1
ON P.USER_ID=U1.USER_ID LEFT OUTER JOIN USERS U2
ON C.USER_ID=U2.USER_ID
WHERE U1.GENDER <> U2.GENDER
GROUP BY U1.USER_ID,U2.USER_ID
HAVING COUNT(*)>=5) T2
WHERE T2.USER1=T1.USER2 AND T2.USER2=T1.USER1)
LIMIT 10;
```

Explanation:

The above query lists upto 10 pairs (one female and the other male) of users by joining POSTS, COMMENTS table through inner join with mentioned conditions and joins USERS tables through RIGHT OUTER JOIN, LEFT OUTER JOIN with mentioned conditions and ensures that the the pair are in such a way that one is female and the other is male and each comments on the other's posts atleast 5 times. Thus, satisfying the question's requirements.

Output Obtained:

```
+-------+-------+
| USER1 | USER2 |
+-------+-------+
|     4 |     1 |
+-------+-------+
1 row in set (0.05 sec)

MySQL [SHOME]>
```