```
CREATE SCHEMA 'social_network';


CREATE TABLE `social_network`.`person` ( // the schema specification implies not using the schema
`NickName` VARCHAR(15) NOT NULL,
`FullName` VARCHAR(45) NOT NULL,
`BirthDate` DATE NOT NULL,
PRIMARY KEY (`NickName`));


CREATE TABLE `social_network`.`post` (
`PostID` INT NOT NULL,
`Datetime` DATETIME NOT NULL,
`Content` VARCHAR(45) NOT NULL,
`PersonNick` VARCHAR(15) NOT NULL,
`WallPersonNick` VARCHAR(15) NOT NULL,
PRIMARY KEY (`PostID`),
INDEX `PersonNick_idx` (`PersonNick` ASC, `WallPersonNick` ASC) VISIBLE,
CONSTRAINT `PersonNick`
FOREIGN KEY (`PersonNick`)
REFERENCES `social_network`.`person` (`NickName` )
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `WallPersonNick`
FOREIGN KEY (`WallPersonNick`)
REFERENCES `social_network`.`person` (`NickName`)
```

```
ON DELETE CASCADE

ON UPDATE CASCADE);


CREATE TABLE `social_network`.`friends` (

`PersonNick1` VARCHAR(15) NOT NULL,

`PersonNick2` VARCHAR(15) NOT NULL,

PRIMARY KEY (`PersonNick1`, `PersonNick2`),

INDEX `PersonNick2_idx` (`PersonNick2` ASC) VISIBLE,

CONSTRAINT `PersonNick1`

FOREIGN KEY (`PersonNick1`)

REFERENCES `social_network`.`person` (`NickName`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `PersonNick2`

FOREIGN KEY (`PersonNick2`)

REFERENCES `social_network`.`person` (`NickName`)

ON DELETE CASCADE

ON UPDATE CASCADE);


CREATE TABLE `social_network`.`liketype` (

`liketype` INT NOT NULL,

`likename` VARCHAR(45) NOT NULL,

PRIMARY KEY (`liketype`));
```

```
CREATE TABLE `social_network`.`like` (
`PostId` INT NOT NULL,
`PersonNick` VARCHAR(15) NOT NULL,
`Datetime` DATETIME NOT NULL,
`Type` INT NOT NULL,
PRIMARY KEY (`PostId`, `PersonNick`),
INDEX `Type_idx` (`Type` ASC) VISIBLE,
CONSTRAINT `Type`
FOREIGN KEY (`Type`)
REFERENCES `social_network`.`liketype` (`liketype`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `PostID`
FOREIGN KEY (`PostId`)
REFERENCES `social_network`.`post` (`PostID`)
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE `social_network`.`comment` (
`CommentID` INT NOT NULL,
`PostID` INT NOT NULL,
`PersonNick` VARCHAR(15) NOT NULL,
`Datetime` DATETIME NOT NULL,
`Text` VARCHAR(45) NOT NULL,
```

```
PRIMARY KEY (`CommentID`),
INDEX `PostID_idx` (`PostID` ASC) VISIBLE,
INDEX `PersonNick_idx` (`PersonNick` ASC) VISIBLE,
CONSTRAINT `PostIDNum`
FOREIGN KEY (`PostID`)
REFERENCES `social_network`.`post` (`PostID`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `PersonNickName`
FOREIGN KEY (`PersonNick`)
REFERENCES `social_network`.`person` (`NickName`)
ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE `social_network`.`private_message` (
`MessageID` INT NOT NULL,
`PersonNick1` VARCHAR(15) NOT NULL,
`PersonNick2` VARCHAR(15) NOT NULL,
`Datetime` DATETIME NOT NULL,
`Text` VARCHAR(45) NOT NULL,
`Read` TINYINT(1) NOT NULL,
`private_messagecol` VARCHAR(45) NULL,
PRIMARY KEY (`MessageID`),
INDEX `PersonNick1_idx` (`PersonNick1` ASC) VISIBLE,
```

```
INDEX `PersonNick2_idx` (`PersonNick2` ASC) VISIBLE,
CONSTRAINT `PersonNickName1`
FOREIGN KEY (`PersonNick1`)
REFERENCES `social_network`.`person` (`NickName`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `PersonNickName2`
FOREIGN KEY (`PersonNick2`)
REFERENCES `social_network`.`person` (`NickName`)
ON DELETE CASCADE
ON UPDATE CASCADE);
```

Query a:

Because the schema representation of the liketype "wow" is 1, I simply have to look
for PostID's of posts created by Mike_34 and that have the liketype 1.

```
SELECT DISTINCT Post1.PostID
FROM social_network.post AS Post1, social_network.liketable AS likes, social_network.liketype AS
liketypes
WHERE Post1.PersonNick = 'Mike_34' AND likes.Type = 1 AND likes.PostId = Post1.PostID;
```

Query b:

The "unread" state is represented by a TINYINT where 1 represents unread and 0 represents read.

```
SELECT DISTINCT PM.MessageID, P.FullName
FROM social_network.private_message AS PM, social_network.person AS P
WHERE PM.PersonNick2 = 'Mike_34' AND PM.ReadCol = 1 AND PM.Datetime < '2018-12-23 00:00:00'
AND PM.PersonNick1 = P.NickName;
```

Query c:

```
(SELECT Per2.FullName
FROM social_network.person AS Per, social_network.person AS Per2, social_network.friends AS F
WHERE F.PersonNick1 = Per.NickName AND F.PersonNick2 = Per2.NickName AND Per.FullName = 'John
Norton')
UNION
(SELECT Per.FullName
FROM social_network.person AS Per, social_network.person AS Per2, social_network.friends AS F
WHERE F.PersonNick1 = Per.NickName AND F.PersonNick2 = Per2.NickName AND Per2.FullName = 'John
Norton');
```

Query d:

```
SELECT DISTINCT Per.FullName
FROM social_network.post AS P, social_network.person AS Per
```

```
WHERE P.PersonNick = Per.NickName AND P.PostID NOT IN (SELECT L.PostId FROM
social_network.liketable AS L) AND P.PostID NOT IN (SELECT C.PostID FROM social_network.comment
AS C);
```

Query e:

```
SELECT count(NickName) FROM social_network.person
WHERE NickName IN
        (SELECT Per.NickName
        FROM social_network.person AS Per, social_network.post AS P, social_network.liketable AS L
        WHERE P.PostId = L.PostId AND Per.NickName = L.PersonNick AND P.PersonNick = 'Mike_34')
AND NickName NOT IN
        (SELECT Per.NickName
        FROM social_network.person AS Per, social_network.post AS P, social_network.liketable AS L
        WHERE P.PostId = L.PostId AND Per.NickName = L.PersonNick AND P.PersonNick = 'Mary_56');
```

Query f:

```
SELECT DISTINCT L.PostId
FROM social_network.liketable L
WHERE NOT EXISTS (
        SELECT *
   FROM social_network.liketype LT
        WHERE NOT EXISTS (
```

```
                SELECT *
    FROM social_network.liketable L2
    WHERE (L2.PostId = L.PostId) AND (L2.Type = LT.liketype)));
```

Query g:

```
SELECT FullName, count(PostID)
FROM social_network.post AS P, social_network.person AS Per
WHERE P.WallPersonNick = Per.NickName
GROUP BY P.WallPersonNick;
```

Query h:

```
UPDATE liketable SET social_network.liketype = (SELECT liketype FROM liketype WHERE
likename = 'fun') WHERE social_network.liketype = (SELECT liketype FROM liketype WHERE
likename = 'wow');

DELETE FROM liketype WHERE likename = 'wow';
```