View 1: Computes a join of at least three tables

```
SELECT *
FROM Filter as F INNER JOIN Tweet as T
ON F.Text_ID = T.Text_ID
INNER JOIN Users as U
ON T.User_ID = U.User_ID
```

View 2: Uses nested queries with the ANY or ALL operator and uses a GROUP BY clause

```
SELECT DisplayName
FROM Users
WHERE User_ID = ANY (SELECT User_ID FROM Tweet WHERE User_ID > 100000)
GROUP BY DisplayName
```

View 3: A correlated nested query

```
SELECT AVG (Favourites_Count)
FROM Filter, Tweet
WHERE Filter.Text_ID = Tweet.Text_ID
```

View 4: Uses a FULL JOIN

```
SELECT st
FROM Tweet
FULL OUTER JOIN Users
ON Tweet.Text_ID = Users.User_ID
```

View 5: Uses nested queries with any of the set operations UNION, EXCEPT, or INTERSECT

```
SELECT User_ID FROM Tweet
UNION
SELECT User_ID FROM Users
ORDER BY User_ID
```

View 6: Computes average retweets count

```
SELECT AVG (Retweet_Count)
FROM Filter, Tweet
WHERE Filter.Text_ID = Tweet.Text_ID
```

View 7: computes highest reply count

```
SELECT COUNT (DISTINCT Reply_Count)
FROM Filter
```

View 8: Finds highest amount of tweets in a single day

```
SELECT COUNT (DISTINCT Text_ID)
FROM Tweet
GROUP BY DateCreated
```

View 9: Latest tweet

```
SELECT *
FROM Tweet
ORDER BY DateCreated DESC
LIMIT 1
```

View 10: Count total tweets

```
SELECT COUNT (Text_ID)
From Tweet
```

View 11: Message containing China

```
SELECT message
FROM Tweet
WHERE message LIKE '%China%'
```

View 12: Total Tweets

```
SELECT COUNT (Text_ID)
From Tweet
```

View 13: Get Tweets from User ID

```
SELECT *
FROM Tweet
WHERE "Text_ID" = %s' , [Tweet_ID]
```