Question 5

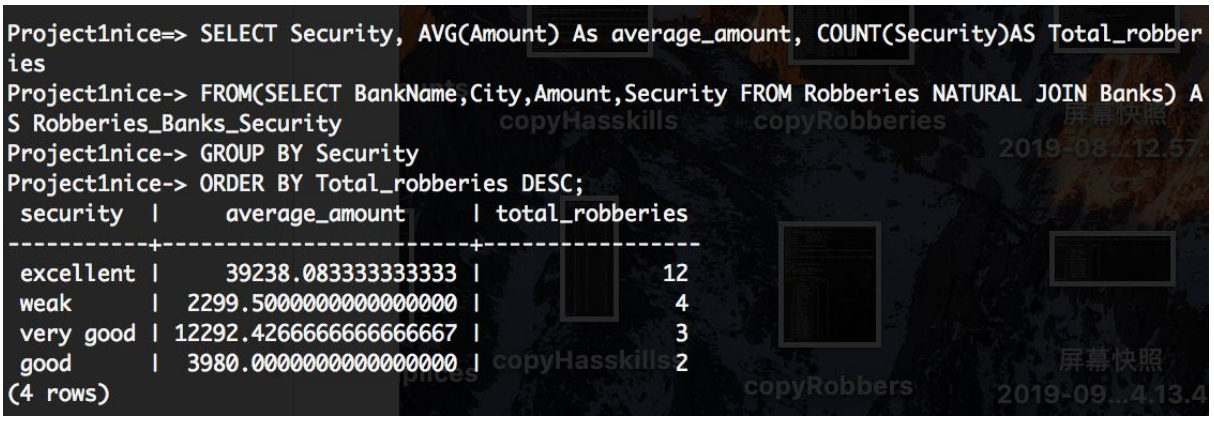
1.

```
-- View For Bank security, Name and amount  Join two table Bank and Robberies
CREATE VIEW amountWithSecurity as (
SELECT b.bankname as bankname,
b.city as city,
b.security as security,
r.amount as amount
FROM Banks b
JOIN Robberies r
ON b.bankname = r.bankname
AND b.city = r.city
ORDER BY b.security);

--Create View for total robberies number and security and average amount
From amountWithSecurity  View --
CREATE VIEW RobberTotalAmountWithSecurity as (
SELECT security as security,
COUNT(security) as Total_robberies,
AVG(amount) as average_amount
FROM amountWithSecurity
GROUP BY security
ORDER BY Total_robberies DESC);

SELECT * FROM amountWithSecurity ;

-- Nested Query --
SELECT Security, AVG(Amount) As average_amount, COUNT(Security)AS Total_robberies
FROM(SELECT BankName,City,Amount,Security FROM Robberies NATURAL JOIN Banks)
AS Robberies_Banks_Security
GROUP BY Security
ORDER BY Total_robberies DESC;
```



```
Project1nice=> SELECT Security, AVG(Amount) As average_amount, COUNT(Security)AS Total_robber
ies
Project1nice-> FROM(SELECT BankName,City,Amount,Security FROM Robberies NATURAL JOIN Banks) A
S Robberies_Banks_Security
Project1nice-> GROUP BY Security
Project1nice-> ORDER BY Total_robberies DESC;
 security  |     average_amount     | total_robberies
-----------+------------------------+-----------------
 excellent |     39238.083333333333 |              12
 weak      |  2299.5000000000000000 |               4
 very good | 12292.426666666666666667 |               3
 good      |  3980.0000000000000000 |               2
(4 rows)
```

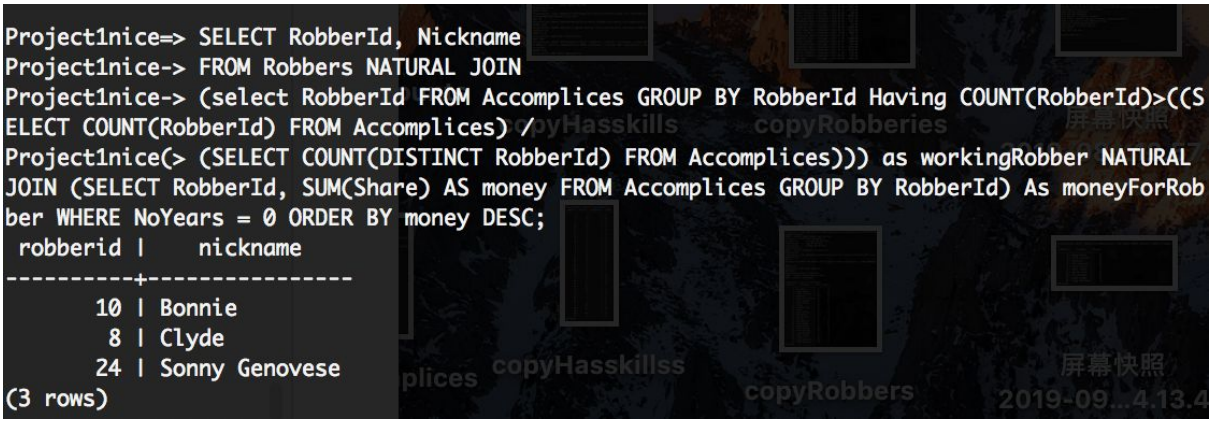
2.

```
-- Find earning of per robber by creating View
CREATE VIEW earningEach as (
select Robberid,
COUNT(Robberid) as Total_robberies,
SUM(Share) as total_earnings
from Accomplices
GROUP BY Robberid);

-- Using earningEach View , find the robbers who participated in more robberies than the
average robber
CREATE VIEW activeRobbers as (
select * from earningEach
WHERE  Total_robberies >
(select AVG (Total_robberies) as Total_robberies
from earningEach));

//View for Nickname of robber and make decreasing or by total earning
CREATE VIEW nicknames as (
select r.RobberId, Nickname
from activeRobbers a
JOIN robbers r
ON r.RobberId = a.RobberId
WHERE r.NoYears = 0
ORDER BY total_earnings DESC);

//Nested Query
SELECT RobberId, Nickname
FROM Robbers NATURAL JOIN
(select RobberId FROM Accomplices GROUP BY RobberId Having
COUNT(RobberId)>((SELECT COUNT(RobberId) FROM Accomplices) /
(SELECT COUNT(DISTINCT RobberId) FROM Accomplices))) as workingRobber
NATURAL JOIN (SELECT RobberId, SUM(Share) AS money FROM Accomplices GROUP
BY RobberId) As moneyForRobber WHERE NoYears = 0 ORDER BY money DESC;
```



3.

```
-- 2018 information of robbed
CREATE VIEW banksInfo2018 as (
select BankName, City , Security from Banks  ,noAccounts
where (Bankname ,City) NOT IN (SELECT BankName, City FROM Robberies
WHERE(date_part('year' , Date) = '2018')));

-- Planned by robbers in 2020
CREATE VIEW Planned2020 as (
SELECT b.BankName, b.City, b.Security, p.NoRobbers, b.NoAccounts, p.PlannedDate
FROM Plans p JOIN  banksInfo2018 b on p.BankName = b.BankName AND p.City = b.City
WHERE (b.Bankname ,b.City) IN (SELECT BankName, City FROM Plans
WHERE(date_part('year' , PlannedDate) = '2020')));

*Last View--
CREATE VIEW AccountRobber as (
SELECT d.NoRobbers, d.BankName, d.City, d.noAccounts
FROM Planned2020 d JOIN Banks b ON d.BankName =b.BankName  AND
d.City = b.City
ORDER BY d.NoRobbers DESC);

SELECT * FROM AccountRobber;


// Single Query
SELECT Planned2020.Security,Planned2020.NoRobbers, Planned2020.BankName,
Planned2020.City, Planned2020.noAccounts FROM (SELECT banksInfo2018.BankName,
banksInfo2018.City, banksInfo2018.Security, p.PlannedDate, p.NoRobbers,
banksInfo2018.NoAccounts FROM Plans p JOIN  (select BankName, City ,noAccounts,
Security from Banks  where (Bankname ,City) NOT IN (SELECT BankName, City FROM
Robberies  WHERE(date_part('year' , Date) = '2018')))
 AS banksInfo2018 on p.BankName = banksInfo2018.BankName AND p.City =
banksInfo2018.City WHERE (banksInfo2018.Bankname ,banksInfo2018.City) IN (SELECT
BankName, City FROM Plans  WHERE(date_part('year' , PlannedDate) = '2020'))) AS
Planned2020;
```

```
Project1nice=> SELECT Planned2020.Security,Planned2020.NoRobbers, Planned2020.BankName, Plann
ed2020.City, Planned2020.noAccounts FROM (SELECT banksInfo2018.BankName, banksInfo2018.City,
banksInfo2018.Security, p.PlannedDate, p.NoRobbers, banksInfo2018.NoAccounts FROM Plans p JOI
N  (select BankName, City ,noAccounts, Security from Banks  where (Bankname ,City) NOT IN (SE
LECT BankName, City FROM Robberies  WHERE(date_part('year' , Date) = '2018')))
Project1nice(>  AS banksInfo2018 on p.BankName = banksInfo2018.BankName AND p.City = banksInf
o2018.City WHERE (banksInfo2018.Bankname ,banksInfo2018.City) IN (SELECT BankName, City FROM
Plans  WHERE(date_part('year' , PlannedDate) = '2020'))) AS Planned2020;
 security  | norobbers |    bankname    |  city   | noaccounts
-----------+-----------+----------------+---------+------------
 excellent |         5 | Hidden Treasure | Chicago |     999999
 weak      |         2 | Bad Bank        | Chicago |       6000
(2 rows)
```

4.

```
*--View Security and Robber Id
CREATE VIEW SecurityWithRobberIdas (
SELECT DISTINCT a.RobberId as RobberId,
b.Security as Security
FROM Banks b
JOIN Accomplices a
ON b.bankname = a.bankname
AND b.city = a.city
ORDER BY security);

-- now this view will display the robberid and skill id next to the security level:
CREATE VIEW securitywithSkillId as (
SELECT h.RobberId as RobberId,
h.SkillId as SkillId,
s.security as security
FROM hasSkills h
Join SecurityWithRobberIdas s
ON h.robberId = s.RobberId);

shows SkillId
CREATE VIEW ShowsDescription as (
SELECT s.Security as Security,
s.RobberId as RobberId,
d.Description as Description
FROM securitywithSkillId S
JOIN skills d
ON s.skillid = d.skillid);

show NickName
CREATE VIEW ShowsNickName as (
SELECT s.security as security,
s.description as description,
r.nickname as nickname
FROM Robbers r
JOIN ShowsDescription s
ON r.robberid = s.robberid
GROUP BY s.security,description,nickname
```

```
ORDER BY s.security  ASC);



select * from ShowsNickName ;

// Query

SELECT w.security as security,
w.description as description,
r.nickname as nickname
FROM Robbers r
JOIN (SELECT j.security as security,
j.robberid as robberid,
s.description as description
FROM (SELECT h.robberid as robberid,
h.skillid as skillid,
k.security as security
FROM hasSkills h
Join (SELECT DISTINCT a.robberid as robber_id,
b.security as security
FROM Banks b
JOIN Accomplices a
ON b.bankname = a.bankname
AND b.city = a.city
ORDER BY security) k
ON h.robberid = k.robber_id) j
JOIN skills s ON j.skillid = s.skillid) w
ON r.robberid = w.robberid;
```

```
Project1nice=> SELECT w.security as security,
Project1nice-> w.description as description,
Project1nice-> r.nickname as nickname
Project1nice-> FROM Robbers r
Project1nice-> JOIN (SELECT j.security as security,
Project1nice(> j.robberid as robberid,
Project1nice(> s.description as description
Project1nice(> FROM (SELECT h.robberid as robberid,
Project1nice(> h.skillid as skillid,
Project1nice(> k.security as security
Project1nice(> FROM hasSkills h
Project1nice(> Join (SELECT DISTINCT a.robberid as robber_id,
Project1nice(> b.security as security
Project1nice(> FROM Banks b
Project1nice(> JOIN Accomplices a
Project1nice(> ON b.bankname = a.bankname
Project1nice(> AND b.city = a.city
Project1nice(> ORDER BY security) k
Project1nice(> ON h.robberid = k.robber_id) j
Project1nice(> JOIN skills s ON j.skillid = s.skillid) w
Project1nice-> ON r.robberid = w.robberid;
 security  |  description   |     nickname
-----------+----------------+------------------
 weak      | Planning       | Al Capone
 weak      | Safe-Cracking  | Al Capone
 weak      | Preaching      | Al Capone
 very good | Planning       | Al Capone
 very good | Safe-Cracking  | Al Capone
 very good | Preaching      | Al Capone
 excellent | Planning       | Al Capone
 excellent | Safe-Cracking  | Al Capone
 excellent | Preaching      | Al Capone
 very good | Explosives     | Bugsy Malone
 excellent | Lock-Picking   | Lucky Luchiano
 excellent | Driving        | Lucky Luchiano
 very good | Guarding       | Anastazia
 excellent | Guarding       | Anastazia
 excellent | Planning       | Mimmy The Mau Mau
 excellent | Driving        | Mimmy The Mau Mau
 weak      | Lock-Picking   | Dutch Schulz
 weak      | Driving        | Dutch Schulz
 excellent | Lock-Picking   | Dutch Schulz
 excellent | Driving        | Dutch Schulz
 weak      | Planning       | Clyde
 weak      | Lock-Picking   | Clyde
 weak      | Scouting       | Clyde
 excellent | Planning       | Clyde
 excellent | Lock-Picking   | Clyde
 excellent | Scouting       | Clyde
 excellent | Preaching      | Bonnie
 excellent | Safe-Cracking  | Meyer Lansky
 very good | Safe-Cracking  | Moe Dalitz
 good      | Money Counting | Mickey Cohen
 good      | Money Counting | Kid Cann
 weak      | Planning       | Boo Boo Hoff
 excellent | Planning       | Boo Boo Hoff
```

5.

```
//Create view average share for each city
CREATE VIEW CityAverageShare as (
select BankName,city, SUM(share)/COUNT(*) as average_share
from accomplices
group by BankName,city,robberydate);
//Create view average share of Chicago
CREATE VIEW ChicagoAverageShare as (
select city, SUM(average_share)/COUNT(*) as average_share
from CityAverageShare
WHERE city = 'Chicago'
group by city);
//Create view average Share of others
CREATE VIEW OtherAverageShare as (
select city, SUM(average_share)/COUNT(*) as average_share
from CityAverageShare
WHERE city <> 'Chicago'
group by city);
//Combine two view using UNION
CREATE VIEW GroupCityAvgShare as(
SELECT * FROM ChicagoAverageShare UNION SELECT * FROM OtherAverageShare);

//Single Query
select city, SUM(average_share)/COUNT(*) as average_share
from (select BankName,city, SUM(share)/COUNT(*) as average_share
from accomplices
group by BankName,city,robberydate) ASCityAverageShare
WHERE city <> 'Chicago'
group by city UNION select city, SUM(average_share)/COUNT(*) as average_share
from(select BankName,city, SUM(share)/COUNT(*) as average_share
from accomplices
group by BankName,city,robberydate) AS CityAverageShare
WHERE city = 'Chicago'
group by city;
```