

SQL codes to answer C-executive questions

1. What is the relationship between credit grade and loan amount?

```
select avg(loan_amnt) as avg_loan_amnt, avg(funded_amnt) as avg_fund_amnt, grade, sub_grade
from loans_info as L, customers as C
where L.member_id = C.member_id
group by C.grade, C.sub_grade
order by sub_grade ;
```

Data Output

	avg_loan_amnt numeric	avg_fund_amnt numeric	grade character varying (10)	sub_grade character varying (10)
1	4885.121951219512	4885.121951219512	A	A1
2	3943.750000000000	3930.681818181818	A	A2
3	4417.187500000000	4386.093750000000	A	A3
4	5459.137055837563	5459.137055837563	A	A4
5	4759.770114942529	4759.770114942529	A	A5
6	3835.887096774194	3823.185483870968	B	B1
7	5537.068965517241	5537.068965517241	B	B2
8	4597.373188405797	4597.373188405797	B	B3
9	4732.866043613707	4732.866043613707	B	B4
10	3886.419753086420	3886.419753086420	B	B5
11	4973.824451410658	4973.824451410658	C	C1
12	3753.476821192053	3753.476821192053	C	C2
13	5650.250000000000	5650.250000000000	C	C3
14	4809.607843137255	4805.882352941176	C	C4
15	5253.925619834711	5253.925619834711	C	C5
16	6598.298429319372	6598.298429319372	D	D1
17	4775.694444444444	4775.694444444444	D	D2
18	5111.464968152866	5111.464968152866	D	D3
19	5945.600000000000	5945.600000000000	D	D4
20	7854.945054945055	7760.989010989011	D	D5
21	7936.619718309859	7936.619718309859	E	E1
22	7782.786885245902	7782.786885245902	E	E2
23	0560.069444444444	0560.069444444444	E	E3
24	8414.583333333333	8295.312500000000	E	E4
25	7782.777777777778	7782.777777777778	E	E5
26	8403.030303030303	8403.030303030303	F	F1
27	9225.000000000000	9225.000000000000	F	F2
28	0269.444444444444	0269.444444444444	F	F3
29	7821.153846153846	7821.153846153846	F	F4
30	3090.000000000000	3090.000000000000	F	F5
31	7850.000000000000	7850.000000000000	G	G1
32	9138.888888888889	9138.888888888889	G	G2
33	6116.666666666667	6116.666666666667	G	G3
34	0595.000000000000	0595.000000000000	G	G4
35	7654.166666666667	7654.166666666667	G	G5

2. When should we suggest customers to apply for individual application or for joint application?

For data analyst version:

Considering that personal status may vary because of the region distinct, we should provide the recommendation based on the geographic information.

- First, make sure that there are difference among individual application and joint application, then we could get the benchmark for each region.

- Second, when we are trying to find a benchmark for recommendation of application type, we may need to consider the geographic condition. Since people in difference place may have difference benchmark of income and lifestyle, which may also lead to varieties of satisfactory account number.

Query1:

```
SELECT * FROM
(SELECT application_type, AVG(annual_inc)::numeric(10,2) AS avg_ann_inc
FROM applications, customers, loans_info
WHERE loans_info.member_id = customers.member_id
AND applications.application_type_id = loans_info.application_type_id
GROUP BY application_type
ORDER BY avg_ann_inc) AS tb4
NATURAL JOIN
(SELECT application_type, AVG(num_sats)::numeric(5,2) AS avg_num_sats
FROM applications, accounts_info, customers, loans_info
WHERE accounts_info.account_info_id = customers.account_info_id
AND loans_info.member_id = customers.member_id
AND applications.application_type_id = loans_info.application_type_id
GROUP BY application_type
ORDER BY avg_num_sats) AS tb5
```

Output1:

	application_type character varying (20)	avg_ann_inc numeric (10,2)	avg_num_sats numeric (5,2)
1	Joint App	61649.41	11.23
2	Individual	79036.05	11.66

Query2:

```
SELECT application_type, addr_state, AVG(annual_inc)::numeric(10,2) AS avg_ann_inc
FROM applications, customers, loans_info, addresses
WHERE addresses.address_id = customers.address_id
AND loans_info.member_id = customers.member_id
AND applications.application_type_id = loans_info.application_type_id
GROUP BY application_type, addr_state
ORDER BY avg_ann_inc
```

Output2:

	application_type character varying (20)	addr_state character varying (10)	avg_ann_inc numeric (10,2)
1	Joint App	AR	21000.00
2	Joint App	SC	21575.00
3	Joint App	NE	28000.00
4	Joint App	OK	34666.67
5	Joint App	OR	34743.00
6	Joint App	KS	39000.00
7	Joint App	ME	40000.00
8	Joint App	NH	40000.00
9	Joint App	NM	40497.25
10	Joint App	FL	41453.15
11	Joint App	LA	45000.00
12	Joint App	ID	45000.00
13	Joint App	NV	48000.00
14	Joint App	CT	49647.14
15	Joint App	OH	49648.14
16	Joint App	KY	49666.67
17	Joint App	HI	50500.00
18	Joint App	MS	54000.00
19	Joint App	AL	55000.00
20	Joint App	WY	56000.00

Query3:

```

SELECT application_type, addr_state, AVG(num_sats)::numeric(5,2) AS avg_num_sats
FROM applications, accounts_info, customers, loans_info, addresses
WHERE addresses.address_id = customers.address_id
AND accounts_info.account_info_id = customers.account_info_id
AND loans_info.member_id = customers.member_id
AND applications.application_type_id = loans_info.application_type_id
GROUP BY application_type, addr_state
ORDER BY avg_num_sats

```

Output 3:

	application_type character varying (20)	addr_state character varying (10)	avg_num_sats numeric (5,2)
1	Joint App	WV	3.00
2	Joint App	LA	4.67
3	Joint App	ID	5.00
4	Joint App	MN	5.75
5	Joint App	KS	6.00
6	Joint App	HI	6.00
7	Joint App	NJ	7.00
8	Joint App	NH	7.00
9	Joint App	MI	8.00
10	Joint App	OK	8.00
11	Joint App	OR	8.00
12	Individual	VT	8.00
13	Individual	DE	8.13
14	Joint App	NM	8.25
15	Individual	WY	8.70
16	Individual	DC	8.83
17	Joint App	RI	9.00
18	Joint App	MD	9.00
19	Individual	MT	9.00
20	Individual	HI	9.00

3. Where does Lending Club's best market locate?

Query:

*SELECT * FROM*

(

SELECT addr_state, AVG(annual_inc)::numeric(10,2) AS avg_ann_inc

FROM addresses, customers

WHERE addresses.address_id = customers.address_id

GROUP BY addr_state

ORDER BY avg_ann_inc DESC

LIMIT 10) AS tb1

NATURAL JOIN

(

SELECT addr_state, COUNT(member_id)AS num_own

FROM addresses, customers

WHERE addresses.address_id = customers.address_id

GROUP BY addr_state, home_ownership

HAVING home_ownership = 'OWN'

ORDER BY num_own DESC

LIMIT 10) AS tb2

NATURAL JOIN

(

```

SELECT addr_state, count(member_id) AS num_id
FROM addresses, customers
WHERE addresses.address_id = customers.address_id
GROUP BY addr_state
ORDER BY num_id DESC
LIMIT 10) AS tb3;

```

OUTPUT:

	addr_state character varying (10)	avg_ann_inc numeric (10,2)	num_own bigint	num_id bigint
1	CA	85070.29	59	695
2	GA	82430.75	17	148
3	NJ	87905.34	22	160
4	NY	86120.92	50	406

4. What is the trend in loan performance health for the past three years?

select

```

(WITH total_by_status AS (
SELECT SUM(loan_amnt) AS total, loan_status FROM (
    SELECT * FROM loans_info WHERE (issue_d LIKE '%2016' OR issue_d LIKE '%2017' OR issue_d LIKE '%2018%')) AS s
JOIN statuses_info ON s.status_id=statuses_info.status_id
GROUP BY loan_status)
SELECT round(bad_total/good_total,4) AS avg_ratio FROM
(SELECT SUM(total) AS bad_total FROM total_by_status
    WHERE loan_status IN ('Late (31-120 days)','In Grace Period','Late (16-30 days)','Charged Off')) AS l1,
(SELECT SUM(total) AS good_total from total_by_status
    WHERE loan_status = 'Current') AS l2),
(WITH total_by_status_2016 AS (
SELECT SUM(loan_amnt) AS total, loan_status FROM (
    SELECT * FROM loans_info WHERE issue_d LIKE '%2016') AS s
JOIN statuses_info ON s.status_id=statuses_info.status_id
GROUP BY loan_status)
SELECT round(bad_total/good_total,4) AS ratio_2016 FROM
(SELECT SUM(total) AS bad_total FROM total_by_status_2016
    WHERE loan_status IN ('Late (31-120 days)','In Grace Period','Late (16-30 days)','Charged Off')) AS l1,
(SELECT SUM(total) AS good_total from total_by_status_2016
    WHERE loan_status = 'Current') AS l2),
(WITH total_by_status_2017 AS (

```

```

SELECT SUM(loan_amnt) AS total, loan_status FROM (
    SELECT * FROM loans_info WHERE issue_d LIKE "%2017") AS s
JOIN statuses_info ON s.status_id=statuses_info.status_id
GROUP BY loan_status)
SELECT round(bad_total/good_total,4) AS ratio_2017 FROM
(SELECT SUM(total) AS bad_total FROM total_by_status_2017
    WHERE loan_status IN ('Late (31-120 days)','In Grace Period','Late (16-30 days)','Charged Off')) AS I1,
(SELECT SUM(total) AS good_total from total_by_status_2017
    WHERE loan_status = 'Current') AS I2),
(WITH total_by_status_2018 AS (
    SELECT SUM(loan_amnt) AS total, loan_status FROM (
        SELECT * FROM loans_info WHERE issue_d LIKE "%2018") AS s
    JOIN statuses_info ON s.status_id=statuses_info.status_id
    GROUP BY loan_status)
    SELECT round(bad_total/good_total,4) AS ratio_2018 FROM
    (SELECT SUM(total) AS bad_total FROM total_by_status_2018
        WHERE loan_status IN ('Late (31-120 days)','In Grace Period','Late (16-30 days)','Charged Off')) AS I1,
    (SELECT SUM(total) AS good_total from total_by_status_2018
        WHERE loan_status = 'Current') AS I2) ;

```

Data Output

	avg_ratio numeric	ratio_2016 numeric	ratio_2017 numeric	ratio_2018 numeric
1	0.1683	0.5330	0.1914	0.0357

5. What loan purposes may result in poor loan performance health ?

For data analyst version:

Our main aim is to see if there are purposes that contribute to a "higher" risk whether the loan will be repaid or not. In order to explore the relationship between loan purposes and loan performance health, we consider this into two perspectives:

- The average amount of loans issued to each purpose
- The bad/good loan ratio for each loan purpose

Query1:

```

select avg(loan_amnt), purpose from loans_info
group by purpose
order by avg desc;

```

Output1:

avg numeric	purpose character varying (100)
17235.576923076923	small_business
16034.821118443904	debt_consolidation
15799.235474006116	home_improvement
15487.314471243043	credit_card
12840.889830508475	major_purchase
12136.363636363636	house
10780.780346820809	other
10123.1132075471698113	medical
9631.7073170731707317	car
9555.6818181818181818	moving
8300.0000000000000000	renewable_energy
7575.0000000000000000	wedding
5760.4838709677419355	vacation
4500.0000000000000000	educational

Query2:

```

CREATE OR REPLACE FUNCTION ratio_by_purpose(purpose varchar(20))
RETURNS numeric(8,4) AS $$
DECLARE
ratio numeric(8,4);
BEGIN
SELECT
INTO ratio
round(bad_total/good_total,4) FROM
(SELECT SUM(total) AS bad_total FROM
(SELECT SUM(loan_amnt) AS total, loan_status FROM (
SELECT * FROM loans_info WHERE loans_info.purpose =
ratio_by_purpose.purpose) AS s
JOIN statuses_info ON s.status_id=statuses_info.status_id
GROUP BY loan_status) AS total_by_status_purpose
WHERE loan_status IN ('Late (31-120 days)','In Grace Period','Late (16-30
days)','Charged Off')) AS I1,
(SELECT SUM(total) AS good_total from
(SELECT SUM(loan_amnt) AS total, loan_status FROM (
SELECT * FROM loans_info WHERE loans_info.purpose =
ratio_by_purpose.purpose) AS s
JOIN statuses_info ON s.status_id=statuses_info.status_id
GROUP BY loan_status) AS total_by_status_purpose
WHERE loan_status = 'Current') AS I2;

```

RETURN ratio;

END; \$\$

LANGUAGE plpgsql;

```
SELECT (SELECT ratio_by_purpose('small_business') AS ratio_small_bus),  
      (SELECT ratio_by_purpose('debt_consolidation') AS ratio_debt_consl),  
        (SELECT ratio_by_purpose('house') AS ratio_house),  
      (SELECT ratio_by_purpose('home_improvement') AS ratio_home_imp),  
      (SELECT ratio_by_purpose('credit_card') AS ratio_credit_card),  
      (SELECT ratio_by_purpose('medical') AS ratio_med),  
  
      (SELECT ratio_by_purpose('other') AS ratio_other),  
  
      (SELECT ratio_by_purpose('car') AS ratio_car),  
      (SELECT ratio_by_purpose('moving') AS ratio_moving),  
      (SELECT ratio_by_purpose('major_purchase') AS ratio_maj_purchase)  
  
;
```

Output2:

ratio_small_bus numeric	ratio_debt_consl numeric	ratio_house numeric	ratio_home_imp numeric	ratio_credit_card numeric	ratio_med numeric	ratio_other numeric	ratio_car numeric	ratio_moving numeric	ratio_maj_purchase numeric
0.8779	0.3552	0.5072	0.3766	0.2219	0.1942	0.2872	0.6359	0.2983	0.3492

6. Which region has the highest bad/good loan ratio?

For data analyst:

Query:

```
select bad_total,good_total, bad_by_state.addr_state, round(bad_total/good_total,4) as ratio from  
(select sum(sum) as bad_total, addr_state from(  
select sum(loan_amnt),addr_state,loan_status from loans_info  
join customers on loans_info.member_id=customers.member_id  
join addresses on customers.address_id=addresses.address_id  
join statuses_info on loans_info.status_id=statuses_info.status_id  
group by (addr_state,loan_status)) as s  
where loan_status IN ('Late (31-120 days)','In Grace Period','Late (16-30 days)','Charged Off')  
group by addr_state) as bad_by_state  
  
join  
(select sum(sum) as good_total, addr_state from(  
select sum(loan_amnt),addr_state,loan_status from loans_info  
join customers on loans_info.member_id=customers.member_id  
join addresses on customers.address_id=addresses.address_id
```



```

join statuses_info on loans_info.status_id=statuses_info.status_id
group by (addr_state,loan_status)) as s
where loan_status = 'Current'
group by addr_state) as good_by_state
on bad_by_state.addr_state = good_by_state.addr_state
order by ratio desc;

```

Data Output

	bad_total numeric	good_total numeric	addr_state character varying (10)	ratio numeric
1	37500	5000	AK	7.5000
2	37450	32400	NE	1.1559
3	156625	255025	KY	0.6142
4	188775	332475	LA	0.5678
5	111175	200250	AR	0.5552
6	216450	432850	AL	0.5001
7	61550	131675	HI	0.4674
8	383650	850775	NC	0.4509
9	511275	1166050	OH	0.4385
10	1268825	2931250	TX	0.4329
11	376475	926075	VA	0.4065
12	196525	490500	NV	0.4007
13	21325	54400	VT	0.3920
14	123250	315025	OR	0.3912
15	1009400	2618000	NY	0.3856
16	360875	955225	PA	0.3778
17	827300	2194275	FL	0.3770
18	93475	252700	OK	0.3699
19	1543275	4264825	CA	0.3619
20	242650	712625	WA	0.3405
21	138575	455750	MN	0.3041
22	25000	85300	UT	0.2931

23	291000	996050	GA	0.2922
24	209200	723750	CO	0.2891
25	349725	1288975	IL	0.2713
26	72800	272575	KS	0.2671
27	200450	762425	MO	0.2629
28	214350	852625	MI	0.2514
29	196325	788250	MA	0.2491
30	111000	519275	WI	0.2138
31	15500	74000	WY	0.2095
32	256925	1247500	NJ	0.2060
33	87400	431450	TN	0.2026
34	125775	623800	CT	0.2016
35	11000	60575	DE	0.1816
36	170400	939875	MD	0.1813
37	113675	627000	AZ	0.1813
38	20000	110400	ME	0.1812
39	15850	89000	SD	0.1781
40	106600	605625	SC	0.1760
41	131075	756800	IN	0.1732
42	22000	150200	NH	0.1465
43	23025	163200	RI	0.1411
44	14375	115425	DC	0.1245
45	14000	131225	MT	0.1067
46	13400	225750	NM	0.0594