```
Part 1
-- drill down, display the loan amount passed for each date
select sum(l.loan_amt), d.date from loan l join date d on l.date_id = d.id group
by d.date order by d.date;
```

```
1  -- drill down, display the loan amount passed for each date
2  select sum(l.loan_amt), d.date from loan l join date d on l.date_id = d.id group by d.date order by d.date;
```

Data Output | Messages | Notifications

	sum bigint	date date
1	7120	2020-01-01
2	9978	2020-01-03
3	4870	2020-01-05
4	18838	2020-01-06
5	8293	2020-01-07
6	5603	2020-01-09
7	1567	2020-01-10
8	2602	2020-01-14
9	3706	2020-01-17

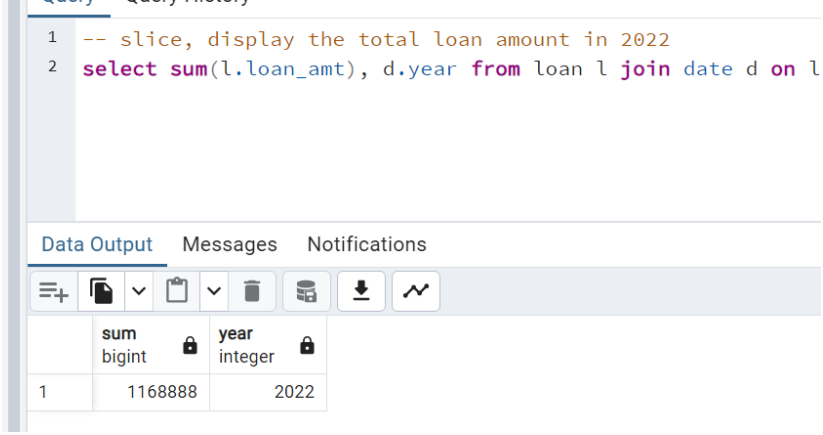
```
-- roll up, display the loan amount passed for each month
select sum(l.loan_amt), d.month, d.year from loan l join date d on l.date_id =
d.id group by d.month, d.year order by d.year, d.month asc;
```

```
1  -- roll up, display the loan amount passed for each month
2  select sum(l.loan_amt), d.month, d.year from loan l join date d on l.date_id = d.id group by d.month, d.year or
```

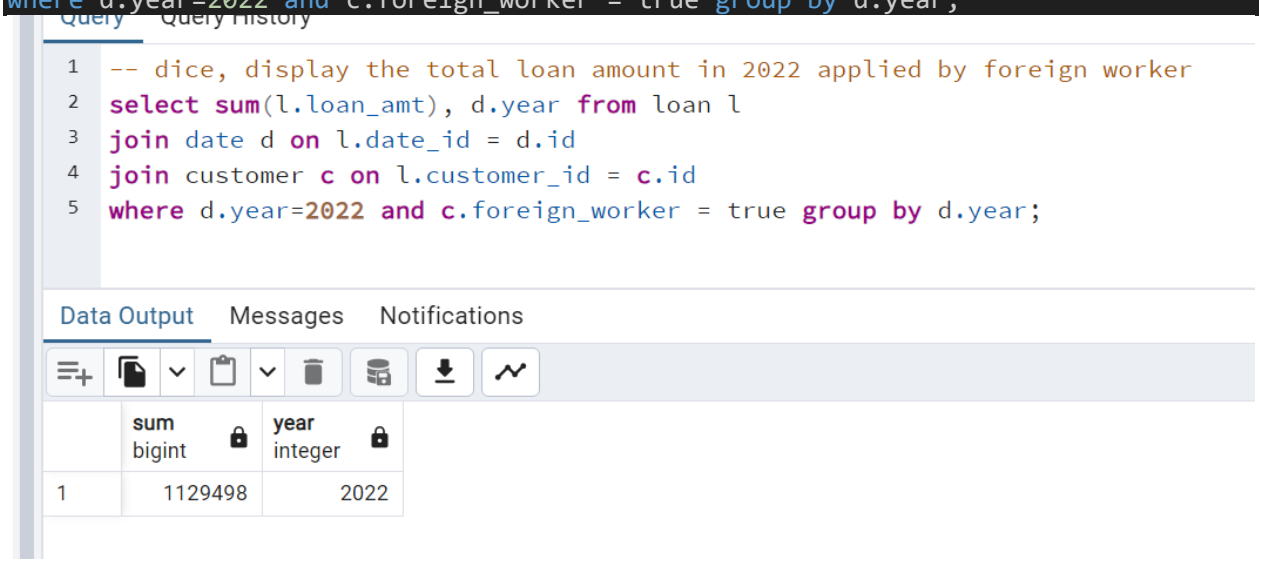
Data Output | Messages | Notifications

	sum bigint	month integer	year integer
1	102177	1	2020
2	88328	2	2020
3	96944	3	2020
4	79677	4	2020
5	146571	5	2020
6	92977	6	2020
7	70052	7	2020
8	85752	8	2020
9	65355	9	2020
10	69697	10	2020
11	119224	11	2020
12	119255	12	2020

```sql
-- slice, display the total loan amount in 2022
select sum(l.loan_amt), d.year from loan l join date d on l.date_id = d.id where
d.year=2022 group by d.year ;
```




```sql
-- dice, display the total loan amount in 2022 applied by foreign worker
select sum(l.loan_amt), d.year from loan l
join date d on l.date_id = d.id
join customer c on l.customer_id = c.id
where d.year=2022 and c.foreign_worker = true group by d.year;
```



```sql
--dice, display the total passed loan in 2022 applied by female
select sum(l.loan_amt), d.year from loan l
join date d on l.date_id = d.id
join customer c on l.customer_id = c.id
where d.year=2022 and c.gender = 'female' group by d.year;
```

```sql
--dice, display the total passed loan in 2022 applied by female
select sum(l.loan_amt), d.year from loan l
join date d on l.date_id = d.id
join customer c on l.customer_id = c.id
where d.year=2022 and c.gender = 'female' group by d.year;
```

Data Output | Messages | Notifications

	sum bigint	year integer
1	283772	2022

```sql
--combine, drill down loan amount to each specific date and slice for year 2022
select sum(l.loan_amt), d.date from loan l
join date d on l.date_id = d.id
where d.year=2022 group by d.date order by d.date;
```

```sql
--combine, drill down loan amount to each specific date and slice for year 2022
select sum(l.loan_amt), d.date from loan l
join date d on l.date_id = d.id
where d.year=2022 group by d.date order by d.date;
```

Data Output | Messages | Notifications

	sum bigint	date date
1	1050	2022-01-01
2	1047	2022-01-03
3	6314	2022-01-04
4	3496	2022-01-05
5	3609	2022-01-07
6	7860	2022-01-09
7	9881	2022-01-10
8	10366	2022-01-11
9	4660	2022-01-13

```sql
--combine, drill down total loan amount to each specific date and slice for only credit over 200
select sum(l.loan_amt), d.date, ca.status from loan l
join checking_account ca on l.checking_account_id = ca.id
join date d on l.date_id = d.id
where ca.status = 'above:200' group by d.date, ca.status order by d.date;
```

Query | Query History

```sql
select sum(l.loan_amt), d.date, ca.status from loan l
join checking_account ca on l.checking_account_id = ca.id
join date d on l.date_id = d.id
where ca.status = 'above:200' group by d.date, ca.status order by d.date;
```

Data Output | Messages | Notifications

	sum bigint	date date	status character varying (20)
1	409	2020-01-30	above:200
2	1474	2020-02-07	above:200
3	2100	2020-02-08	above:200
4	1225	2020-02-10	above:200
5	1961	2020-03-04	above:200
6	3617	2020-04-18	above:200
7	392	2020-05-07	above:200
8	781	2020-05-14	above:200
9	1881	2020-05-25	above:200

```sql
--combine, roll up total loan amount to each year and slice for only credit over 200
select sum(l.loan_amt), d.year, ca.status from loan l
join checking_account ca on l.checking_account_id = ca.id
join date d on l.date_id = d.id
where ca.status = 'above:200' group by d.year, ca.status order by d.year;
```

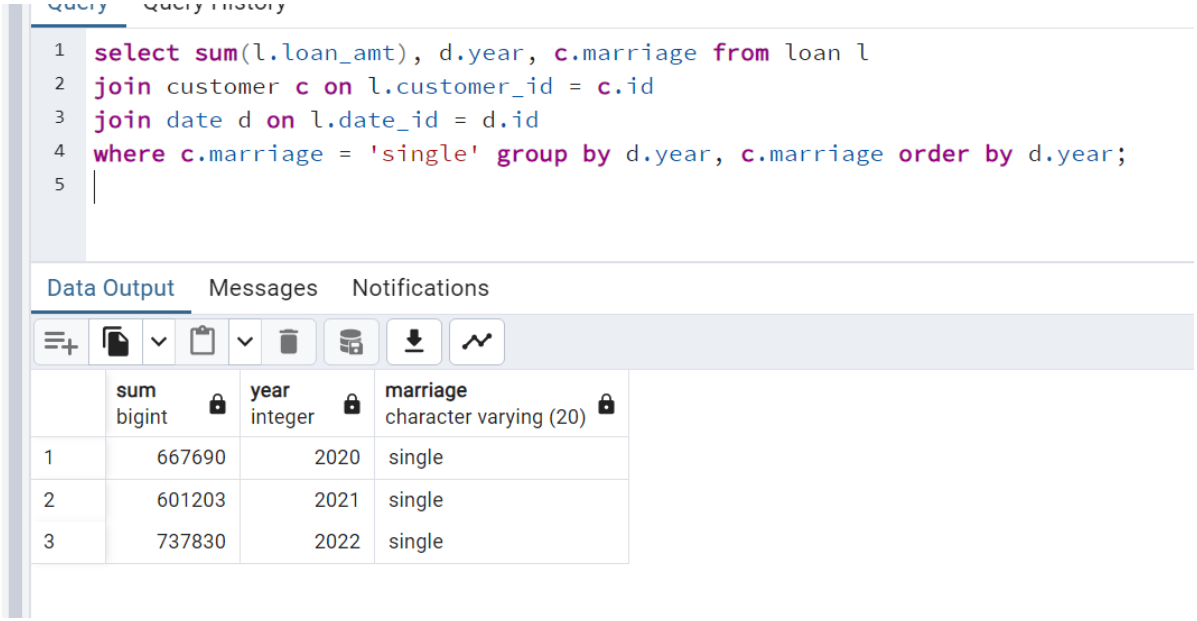
Query | Query History

```sql
select sum(l.loan_amt), d.year, ca.status from loan l
join checking_account ca on l.checking_account_id = ca.id
join date d on l.date_id = d.id
where ca.status = 'above:200' group by d.year, ca.status order by d.year;
```

Data Output | Messages | Notifications

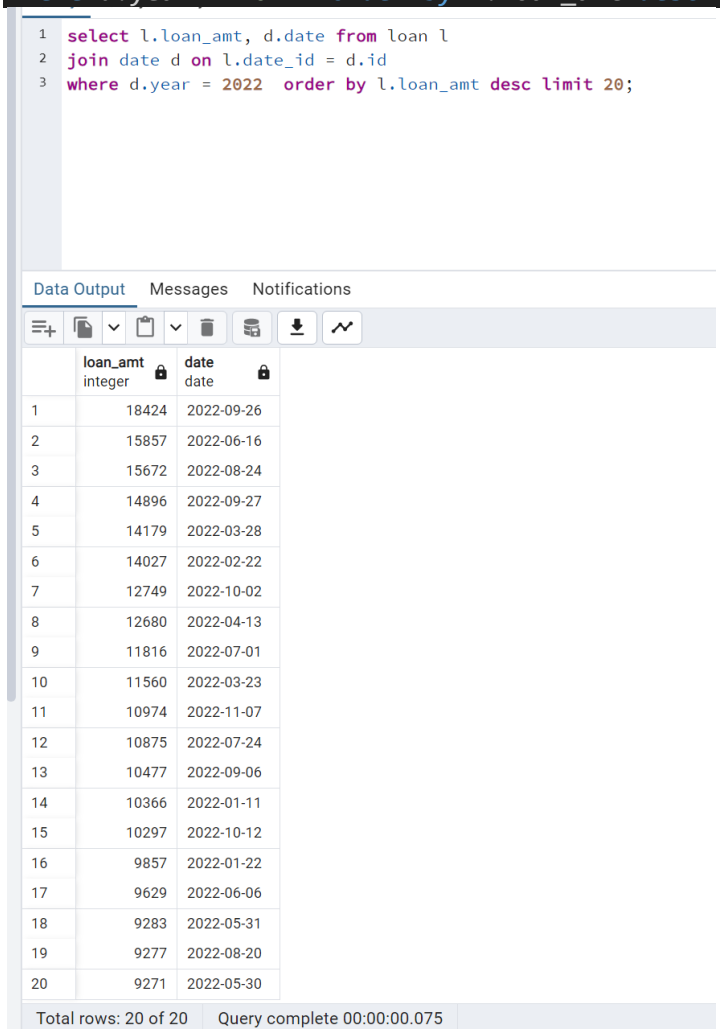
	sum bigint	year integer	status character varying (20)
1	51905	2020	above:200
2	42335	2021	above:200
3	42952	2022	above:200

```sql
--combine, roll up total loan amount to each year and slice for only single
customer
select sum(l.loan_amt), d.year, c.marriage from loan l
join customer c on l.customer_id = c.id
join date d on l.date_id = d.id
where c.marriage = 'single' group by d.year, c.marriage order by d.year;
```



Part2

```sql
-- iceburg display the 20 biggest loan passed in 2022
select l.loan_amt, d.date from loan l
join date d on l.date_id = d.id
where d.year = 2022  order by l.loan_amt desc limit 20;
```



```sql
select l.loan_amt, d.date from loan l
join date d on l.date_id = d.id
where d.year = 2022  order by l.loan_amt desc limit 20;
```

Data Output | Messages | Notifications

	loan_amt integer	date date
1	18424	2022-09-26
2	15857	2022-06-16
3	15672	2022-08-24
4	14896	2022-09-27
5	14179	2022-03-28
6	14027	2022-02-22
7	12749	2022-10-02
8	12680	2022-04-13
9	11816	2022-07-01
10	11560	2022-03-23
11	10974	2022-11-07
12	10875	2022-07-24
13	10477	2022-09-06
14	10366	2022-01-11
15	10297	2022-10-12
16	9857	2022-01-22
17	9629	2022-06-06
18	9283	2022-05-31
19	9277	2022-08-20
20	9271	2022-05-30

Total rows: 20 of 20 Query complete 00:00:00.075

```sql
-- windowing, display the rank by each loan amount partitioned by applicant job
type
WITH loan_avg AS(
  SELECT l.loan_amt,
         c.job_type,
         ROUND(AVG(l.loan_amt) OVER (PARTITION BY c.job_type), 2) AS
avg_loan_amt_by_job_type
  FROM loan l
  JOIN customer c ON l.customer_id = c.id)

SELECT loan_amt,
       job_type,
       avg_loan_amt_by_job_type,
       RANK() OVER (PARTITION BY job_type ORDER BY loan_amt) AS loan_amt_rank
FROM loan_avg;
```

Query | Query History

```sql
3          c.job_type,
4          ROUND(AVG(l.loan_amt) OVER (PARTITION BY c.job_type), 2) AS avg_
5    FROM loan l
6    JOIN customer c ON l.customer_id = c.id)
7
8  SELECT loan_amt,
9         job_type,
10        avg_loan_amt_by_job_type,
11        RANK() OVER (PARTITION BY job_type ORDER BY loan_amt) AS loan_amt_
12 FROM loan_avg;
```

Data Output | Messages | Notifications

	loan_amt integer	job_type character varying (30)	avg_loan_amt_by_job_type numeric	loan_amt_rank bigint
1	629	management_or_self_emp	5435.49	1
2	1050	management_or_self_emp	5435.49	2
3	1107	management_or_self_emp	5435.49	3
4	1164	management_or_self_emp	5435.49	4
5	1199	management_or_self_emp	5435.49	5
6	1209	management_or_self_emp	5435.49	6
7	1231	management_or_self_emp	5435.49	7
8	1238	management_or_self_emp	5435.49	8
9	1278	management_or_self_emp	5435.49	9
10	1337	management_or_self_emp	5435.49	10

```sql
--window clause, show the rank by each loan amount partitioned by allicant
marriage status
WITH loan_avg AS(
  SELECT l.loan_amt,
         c.marriage,
         ROUND(AVG(l.loan_amt) OVER W, 2) AS avg_loan_amt_by_job_type
  FROM loan l
  JOIN customer c ON l.customer_id = c.id
  WINDOW W AS (PARTITION BY c.marriage ORDER BY l.loan_amt))

SELECT loan_amt,
       marriage,
       avg_loan_amt_by_job_type,
       RANK() OVER W AS loan_amt_rank
FROM loan_avg
WINDOW W AS (PARTITION BY marriage ORDER BY loan_amt);
```

Query | Query History

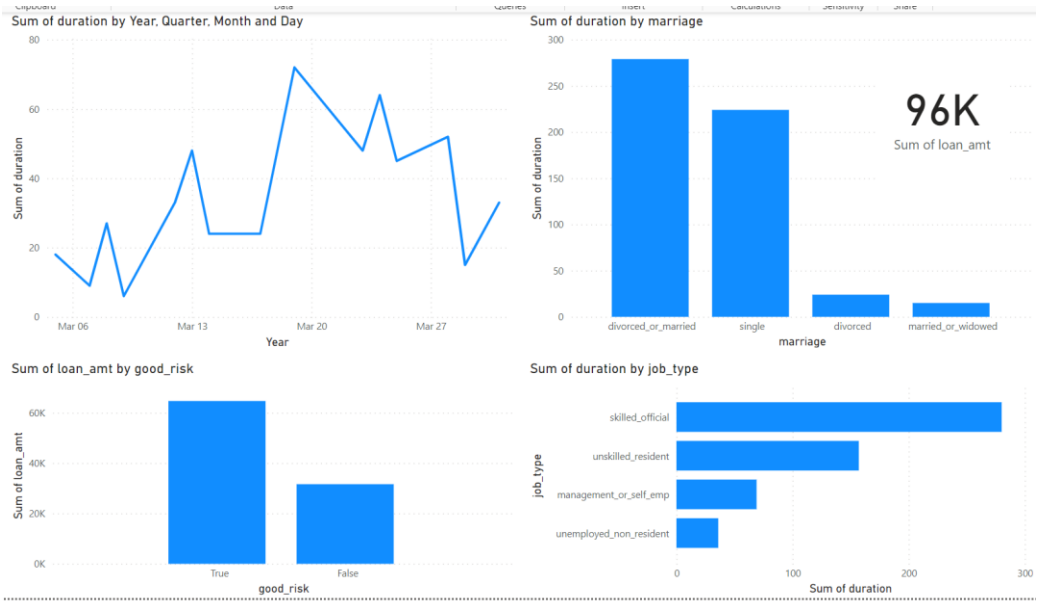
```sql
WITH loan_avg AS(
  SELECT l.loan_amt,
         c.marriage,
         ROUND(AVG(l.loan_amt) OVER W, 2) AS avg_loan_amt_by_job_type
  FROM loan l
  JOIN customer c ON l.customer_id = c.id
  WINDOW W AS (PARTITION BY c.marriage ORDER BY l.loan_amt))

SELECT loan_amt,
       marriage,
       avg_loan_amt_by_job_type,
       RANK() OVER W AS loan_amt_rank
FROM loan_avg
WINDOW W AS (PARTITION BY marriage ORDER BY loan_amt);
```

Data Output | Messages | Notifications

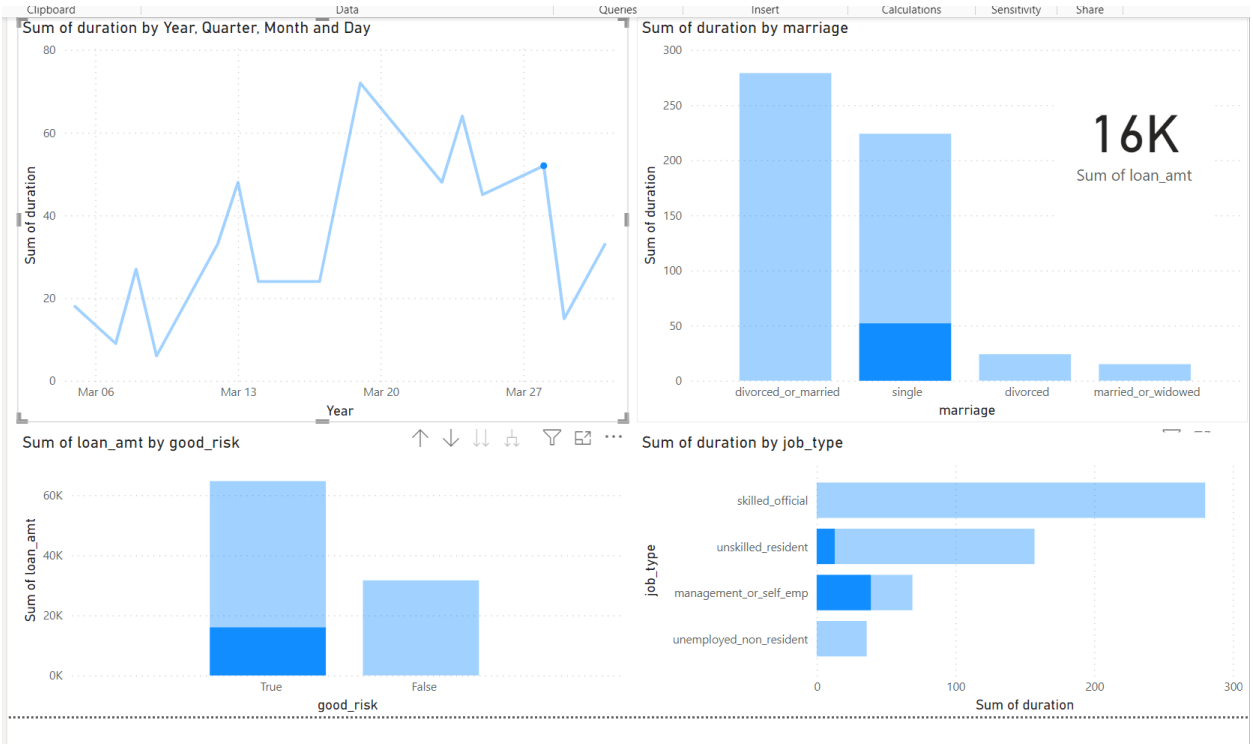
	loan_amt integer	marriage character varying (20)	avg_loan_amt_by_job_type numeric	loan_amt_rank bigint
1	640	divorced	640.00	1
2	975	divorced	807.50	2
3	1126	divorced	913.67	3
4	1158	divorced	974.75	4
5	1223	divorced	1024.40	5
6	1262	divorced	1064.00	6
7	1275	divorced	1094.14	7
8	1323	divorced	1122.75	8
9	1338	divorced	1146.67	9
10	1372	divorced	1169.20	10
11	1449	divorced	1194.64	11
12	1521	divorced	1221.83	12
13	1543	divorced	1246.54	13

PartB

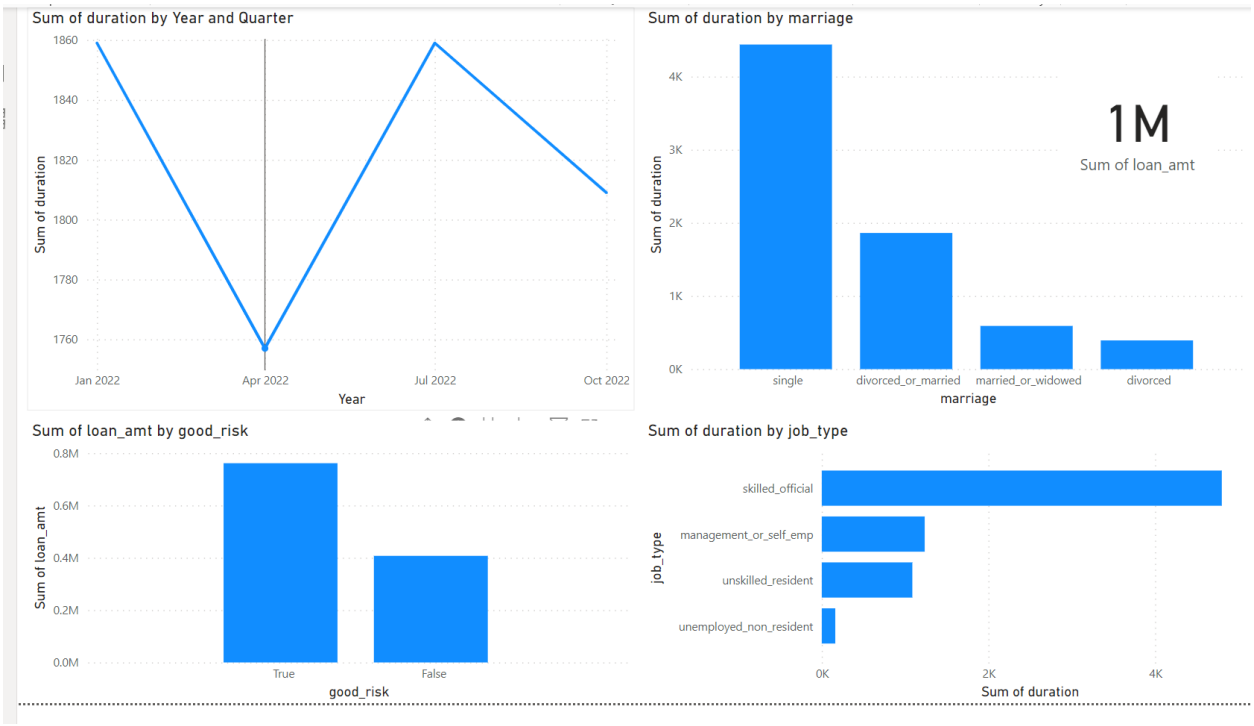
Drill down to a month (date hierarchy: year – month - day). Display the loan amount for each day in March and the sum in the month.



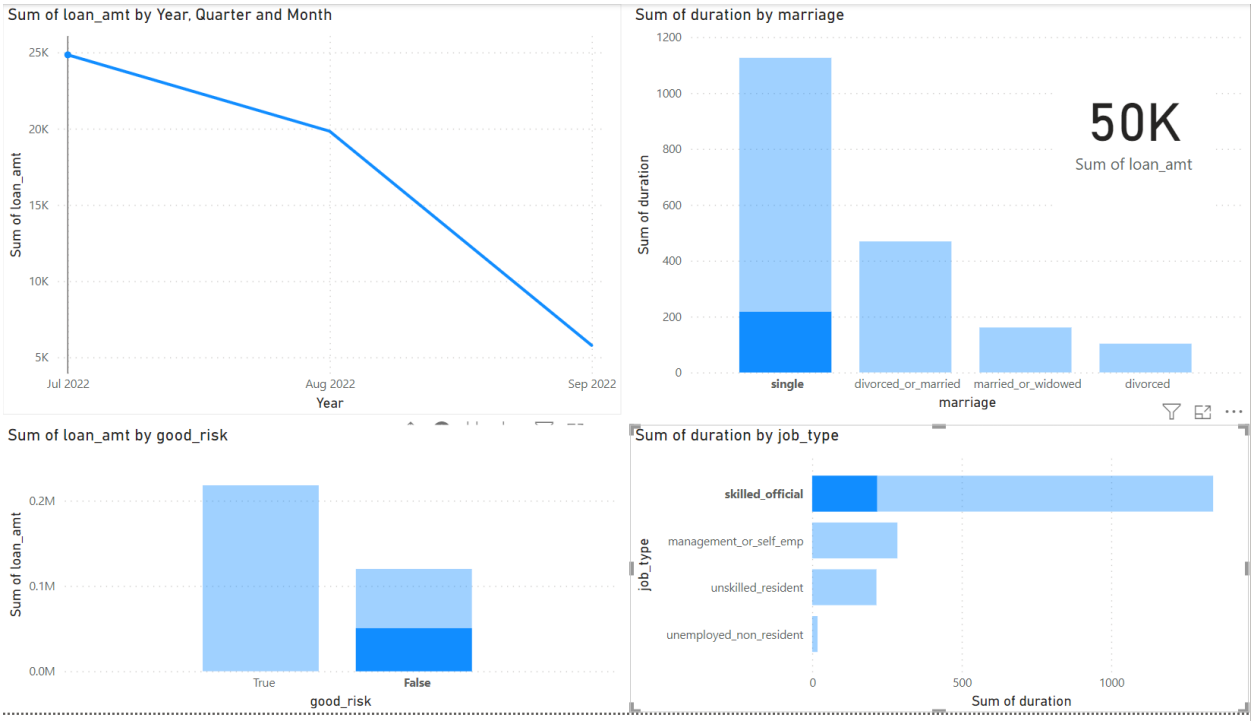
Drill down to a day (date hierarchy: year – month - day). Display the loan amount for 28 March 2022.



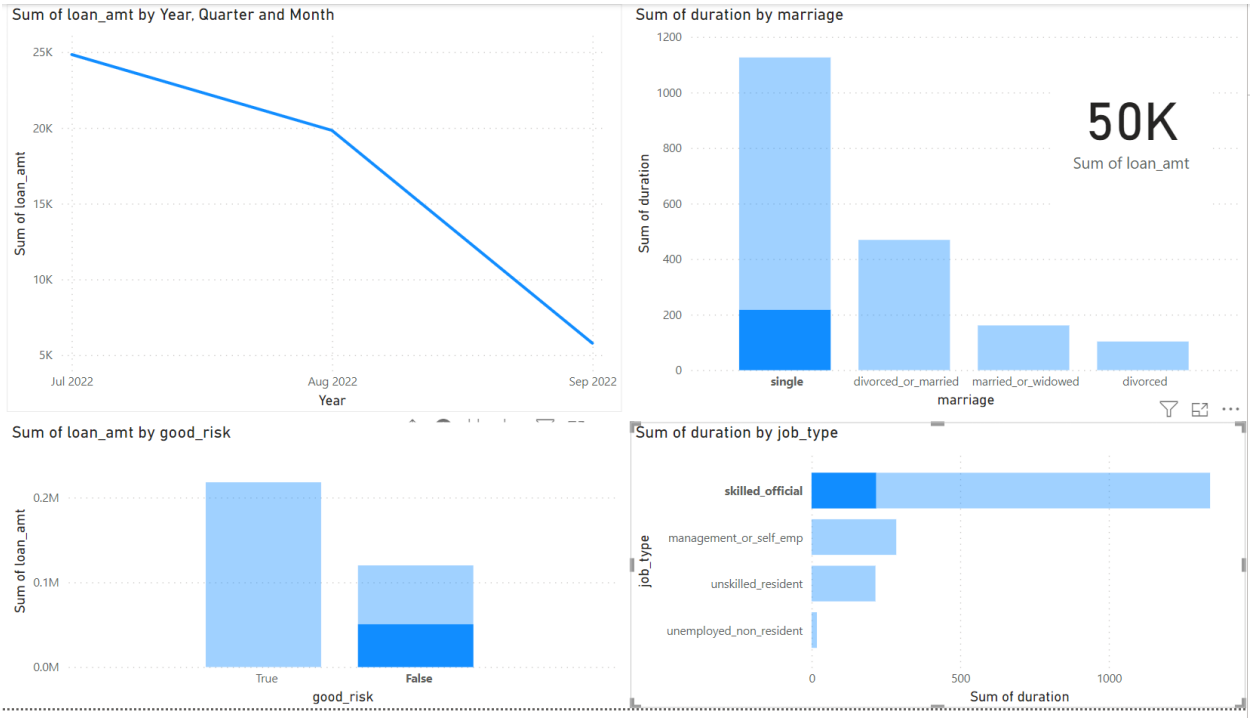
Rollup to a year (date hierarchy: year – month - day). Display the loan amount in 2022.



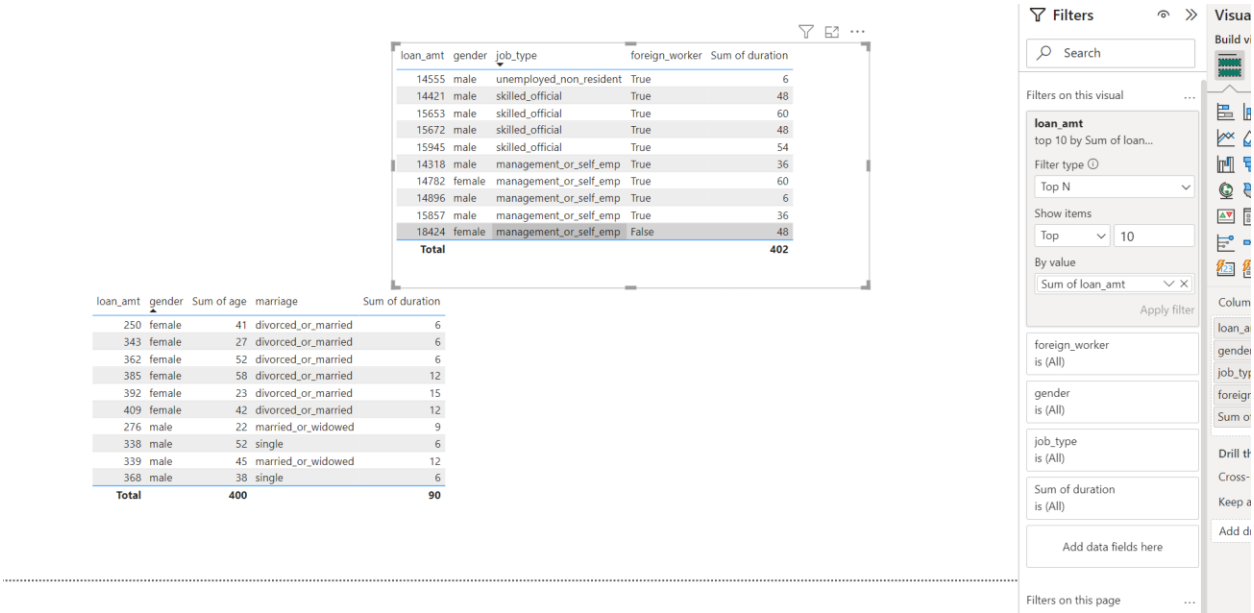
Drill down to a month (date hierarchy: year – month - day), slice marriage dimension. Display the loan amount in August 2022 for single applicant.



Drill down to a month (date hierarchy: year – month - day), dice by risk and job dimension. Display the loan amount in August 2022 for skilled official applicant with bad risk.



Top 10 biggest loan applicant information and bottom 10 smallest loan applicants information



loan_amt	gender	job_type	foreign_worker	Sum of duration
14555	male	unemployed_non_resident	True	6
14421	male	skilled_official	True	48
15653	male	skilled_official	True	60
15672	male	skilled_official	True	48
15945	male	skilled_official	True	54
14318	male	management_or_self_emp	True	36
14782	female	management_or_self_emp	True	60
14896	male	management_or_self_emp	True	6
15857	male	management_or_self_emp	True	36
18424	female	management_or_self_emp	False	48
Total				**402**

loan_amt	gender	Sum of age	marriage	Sum of duration
250	female	41	divorced_or_married	6
343	female	27	divorced_or_married	6
362	female	52	divorced_or_married	6
385	female	58	divorced_or_married	12
392	female	23	divorced_or_married	15
409	female	42	divorced_or_married	12
276	male	22	married_or_widowed	9
338	male	52	single	6
339	male	45	married_or_widowed	12
368	male	38	single	6
Total		**400**		**90**