SQL: Anticipated Payment

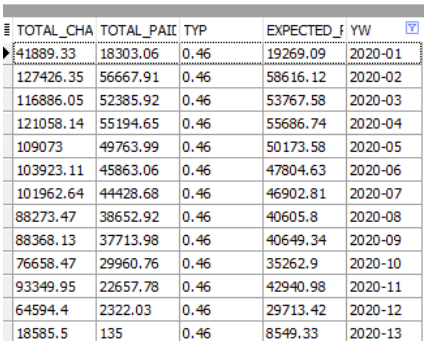
Last Modified on 01/06/2022 9:49 am EST

There are **two versions** of this SQL: Firebird and MySQL. The **MySQL version** only applies to clients who are **BETA testing MySQL**. All other Practices should continue to use the Firebird version of this code. Click to expand and copy the code you need. If you are unsure which code to use, please check with your Practice Administrator.

About

Estimate what your payments will be in light of severely reducing your charges. The SQL can be used for SBA loans to substantiate decreased future payment for present and past work.

A sample image of this SQL report run in the Database Viewer is shown below:



TOTAL_CHA	TOTAL_PAII	TYP	EXPECTED_I	YW
41889.33	18303.06	0.46	19269.09	2020-01
127426.35	56667.91	0.46	58616.12	2020-02
116886.05	52385.92	0.46	53767.58	2020-03
121058.14	55194.65	0.46	55686.74	2020-04
109073	49763.99	0.46	50173.58	2020-05
103923.11	45863.06	0.46	47804.63	2020-06
101962.64	44428.68	0.46	46902.81	2020-07
88273.47	38652.92	0.46	40605.8	2020-08
88368.13	37713.98	0.46	40649.34	2020-09
76658.47	29960.76	0.46	35262.9	2020-10
93349.95	22657.78	0.46	42940.98	2020-11
64594.4	2322.03	0.46	29713.42	2020-12
18585.5	135	0.46	8549.33	2020-13

Caveats

Read all of the below before running the SQL.

- Use a 6-12 month period ending with the current day when prompted to enter the begin and end dates. Larger practices should use a smaller date range.
- This report will take a long time to run. You may consider using the Extract option to save the report to a file on your computer.
- See below for an explanation of each column using the above image as an example.
  - **YW**: This is the year-week of the year, for example, 2020-01 if the first week of 2020 and 2019-52 is the last week of 2019.
  - **TOTAL_CHARGES**: This is the total amount charged for dates of service in the *YW* column.
  - **TOTAL_PAID-TO_DATE**: The column will display the amount you have been paid, so far, for the work you did with DOS week *YW*. For example, using the above image, the practice charged $41,889.33 the first week in January 2020. The practice was paid $18.303.06 at some point thereafter. This is not the same as what was posted during the week *YW*, which would have been mostly work done in December.
  - **TYP**: This column will display the typical collections ration computed for the period you have run the report, minus the last 120 days. This would be total paid to date / total charges.
  - **EXPECTED_PAYMENT**: This column will display the *TOTAL_CHARGES* times the *TYP*. In other words, if you were paid at your typical collections ratio for those charges, what would you have been paid? This number is not going to be precise with *TOTAL_PAID*. *TOTAL_PAID* shows what you actually collected for that work. *EXPECTED_PAYMENT* shows an estimate based on prior performance. The more recent you are in time, i.e. if 120 days have not passed yet, the more discrepant these two numbers are. For example, you can see that for the weeks of Janaury, this practice has collected about $1000.00 below the estimated week; this is usually late insurance payments or corrections or late patient payments coming in. The more recent weeks, e.g. March weeks, are substantially lower, because many of those claims have not yet processed. The *EXPECTED_PAYMENT* column is what the practice can

reasonable expect to be paid based on their charge work for every week of the year.

SQL Code: Firebird

To highlight and copy the code below to your clipboard, simply click the **Copy** button.

Copy

```
with overview as (
select patno, trnsxno, date1, charge, p,  (extract(year from date1) || '-' || lpad(extract(week from date1),2,0)) as yw from
archive_transactions at1
left outer join (select txnopaid, sum(payment + copayrecd) as p from archive_transactions where date1>= :Start_date g
roup by txnopaid) at2 on at2.txnopaid = at1.trnsxno
 where at1.date1 between :Start_date and :end_date and at1.txnopaid = 0
),
tpc as
(select  round(sum(p)/sum(charge),2) as typ from overview where date1 <= :end_date - 120)
select total_charge, total_paid_to_date, typ, round( total_charge * (select typ from tpc) ,2) as expected_payment, yw fro
m (
select round(sum(charge),2) as total_charge, round(sum(p),2) as total_paid_to_date, (select typ from tpc) as typ,  yw fro
m overview group by yw
)
```

SQL Code: MySQL

To highlight and copy the code below to your clipboard, simply click the **Copy** button.

Copy

```
select total_charge, total_paid_to_date, typ, round( total_charge * (select typ from
(select  round(sum(p)/sum(charge),2) as typ from (
select patno, trnsxno, date1, charge, p,  (extract(year from date1) || '-' || lpad(extract(week from date1),2,0)) as yw from
archive_transactions at1
left outer join (select txnopaid, sum(payment + copayrecd) as p from archive_transactions where date1>= :start_date g
roup by txnopaid) at2 on at2.txnopaid = at1.trnsxno
 where at1.date1 between :start_date and :end_date and at1.txnopaid = 0
) overview where date1 <= date_add(cast(:end_date as date),interval -120 day) )
tpc) ,2) as expected_payment, yw from (
select round(sum(charge),2) as total_charge, round(sum(p),2) as total_paid_to_date, (select typ from
(select  round(sum(p)/sum(charge),2) as typ from (
select patno, trnsxno, date1, charge, p,  (extract(year from date1) || '-' || lpad(extract(week from date1),2,0)) as yw from
archive_transactions at1
left outer join (select txnopaid, sum(payment + copayrecd) as p from archive_transactions where date1>= :start_date g
roup by txnopaid) at2 on at2.txnopaid = at1.trnsxno
 where at1.date1 between :start_date and :end_date and at1.txnopaid = 0
) overview where date1 <= date_add(cast(:end_date as date),interval -120 day) )
tpc) as typ,  yw from
(
select patno, trnsxno, date1, charge, p,  (extract(year from date1) || '-' || lpad(extract(week from date1),2,0)) as yw from
archive_transactions at1
left outer join (select txnopaid, sum(payment + copayrecd) as p from archive_transactions where date1>= :start_date g
roup by txnopaid) at2 on at2.txnopaid = at1.trnsxno
 where at1.date1 between :start_date and :end_date and at1.txnopaid = 0
)
 overview group by yw
)t1
```