

SQL: Zero-Paid Claims Per Month

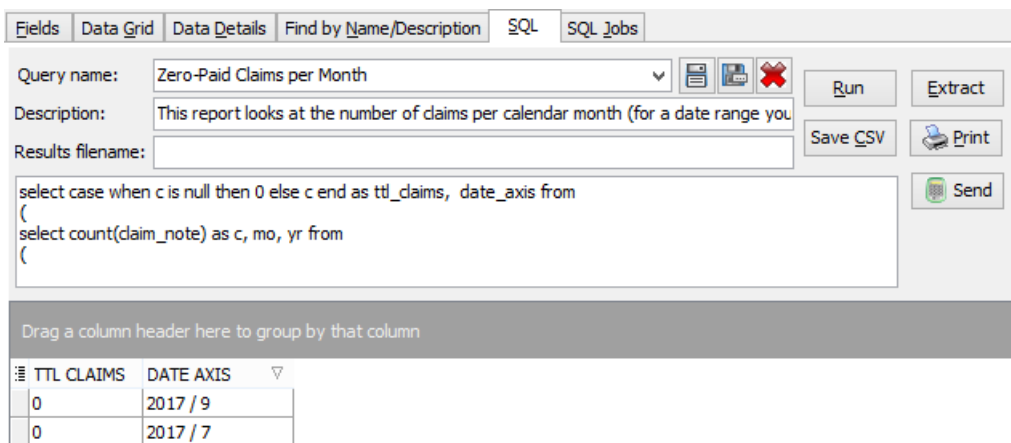
Last Modified on 01/06/2022 11:08 am EST

i 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

This report looks at the number of claims per calendar month (for a date range you specify) that were completely written off on the insurance-side. This is a good general monitor for performance around timely filing and appeals for your billing department.

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



The screenshot shows a Database Viewer window with the following details:

- Query name: Zero-Paid Claims per Month
- Description: This report looks at the number of claims per calendar month (for a date range you specify)
- Results filename: (empty)
- SQL Code:

```
select case when c is null then 0 else c end as ttl_claims, date_axis from  
(  
select count(claim_note) as c, mo, yr from  
(
```
- Buttons: Run, Extract, Save CSV, Print, Send
- Table below the query:

TTL CLAIMS	DATE AXIS
0	2017 / 9
0	2017 / 7

Caveats

This SQL looks only for adjudicated claims where the total adjustment amount is equal to the total charge amount. The total charge amount must be at least \$0.01.

SQL Code: Firebird

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

Copy

```
select case when c is null then 0 else c end as ttl_claims, date_axis from
(
select count(claim_note) as c, mo, yr from
(
select extract(month from claimdate) as mo, extract(year from claimdate) as yr, claimdate, claim_note from
(
select claimdate, claim_note, invoiceno, charge, payment, adjustment from claims1 where charge=adjustment and clai
mdate between :start_date and :end_date and charge >= 0.01
order by claimdate
)
) a
group by a.yr, a.mo
) b

right outer join
(
select ( yr || ' / ' || mo) as date_axis, mo, yr from
(
select distinct extract(month from date1) as mo, extract(year from date1) as yr from archive_transactions where date1 b
etween :start_date and :end_date
order by yr, mo
)
) t on t.yr=b.yr and t.mo=b.mo
```

SQL Code: MySQL

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

Copy

```
select case when c is null then 0 else c end as ttl_claims, date_axis from
(
select count(claim_note) as c, mo, yr from
(
select extract(month from claimdate) as mo, extract(year from claimdate) as yr, claimdate, claim_note from
(
select claimdate, claim_note, invoiceno, charge, payment, adjustment from claims1 where charge=adjustment and clai
mdate between :start_date and :end_date and charge >= 0.01
order by claimdate
) t2
) a
group by a.yr, a.mo
) b
right outer join
(
select ( yr || ' / ' || mo) as date_axis, mo, yr from
(
select distinct extract(month from date1) as mo, extract(year from date1) as yr from archive_transactions where date1 b
etween :start_date and :end_date
order by yr, mo
) t1
) t on t.yr=b.yr and t.mo=b.mo
```