Important Content Update Message

We are currently updating the OP Help Center content for the release of OP 20. OP 20 (official version 20.0.x) is the certified, 2015 Edition, version of the Office Practicum software. This is displayed in your software (**Help tab > About**) and in the Help Center tab labeled Version 20.0. We appreciate your patience as we continue to update all of our content.

SQL: Unpaid Claims with the Most Statements

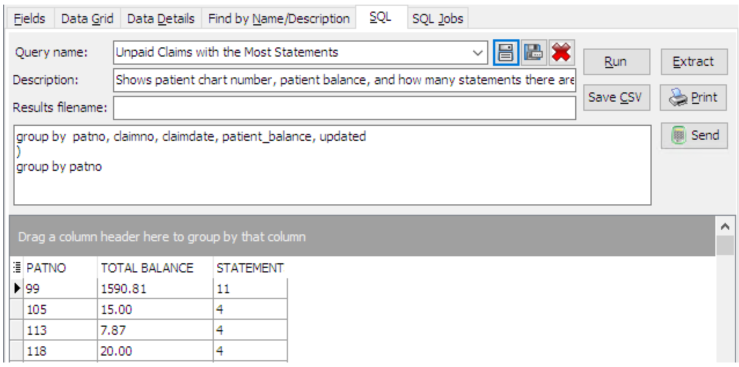
Last Modified on 12/03/2019 10:45 am EST

About

This report lists the patient chart number, patient balance, and how many statements there are for the unpaid claim with the most statements.

Example: Patient chart #123 owes three different amounts. For the first date of service, the patient owes $40, for which you have sent three statements. For a second date of service, the patient owes $20, for which you have sent two statements. For a third date of service, the patient owes $10, for which you have not yet sent a statement. This report will show that patient 123 owes $70 ($40 + $20 + $10) and has received three statements.

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



Query name: Unpaid Claims with the Most Statements

Description: Shows patient chart number, patient balance, and how many statements there are

Results filename:

```
group by  patno, claimno, claimdate, patient_balance, updated
)
group by patno
```

Drag a column header here to group by that column

PATNO	TOTAL BALANCE	STATEMENT
99	1590.81	11
105	15.00	4
113	7.87	4
118	20.00	4

Caveats

- You have to have printed the statements from within the OP software. Printing statements through OP reports does not count.
- The same claim printed twice on the same day only counts as one statement. This accounts

for misprints/accidents.

- These are grouped by patient (not family/guarantor).
- Just because you printed the statement doesn’t necessarily mean you’ve mailed it yet (or at all!).

Code

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

Copy

```
select patno, sum(patient_balance) as total_balance, max(statements_printed) as statements
from
(

select patno, claimno, claimdate, patient_balance, updated, count(statement_date) as statements_printed
from
(

select distinct claims1.invoiceno as claimno, claimdate, patno, patient_balance, updated, statement_date from claims1
left outer join  (select print_log.*, substring(cast(date1 as date) from 1 for position(' ' in cast(date1 as date)))  as statement_date from print_log where tracking_id = 'Statement' and recipient='Patient') p  on p.invoiceno = claims1.invoiceno
where patient_balance >= 0.01 and statement_date is not null
)
group by  patno, claimno, claimdate, patient_balance, updated
)
group by patno
```