SQL: Vaccine Payments vs Price Paid
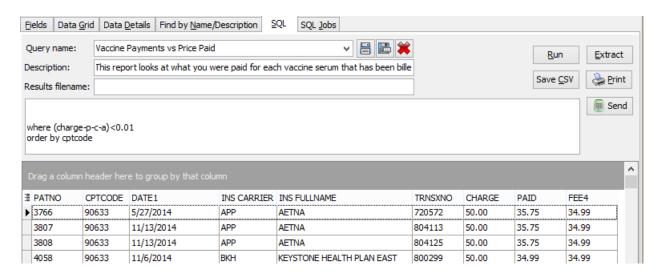
Last Modified on 01/06/2022 11:04 am EST

> ⓘ 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

This report looks at what you were paid for each vaccine serum that has been billed and completely adjudicated. It compares what you were paid to the cost of the vaccine, assuming you have put the cost per dose in FEE4 in the CPT code tables.

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

Caveats

- This SQL only shows vaccine serum payments (not vaccine administration payments).
- "FEE4" pulls from the CPT code tables. This requires you to put your current cost per dose in fee schedule #4. To use a different fee schedule, you can edit the SQL to read e.g. FEE14 or FEE19.
- Only vaccine serum codes between 90620 and 90746 appear in this report.
- Payments include both insurance-side and patient-side, and reflect actual payments (not just allowables.)
- Only completed claims (i.e. no balance) are included. That is, claims you just sent yesterday with vaccines won't appear here.
- Vaccines that you charged $0.00 or $0.01 for, i.e. VFC vaccines, are not listed here. If you charged more than $0.01 on the serum code of a VFC vaccine (i.e. you are in a state that requires you to charge VFC admin fee on the VFC product code), you will get weird results.

SQL Code: Firebird

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

Copy

```
select patno, cptcode, date1, at3.ins_carrier_code, ins_carrier.ins_fullname, trnsxno, charge, p+c as paid, cost.fee4
from
(

select patno, cptcode,  date1, trnsxno, charge, sum(payment) as p, sum(copayrecd) as c, sum(adjustment) as a from
(
select at1.patno, at1.date1, at1.trnsxno, at1.cptcode,at1.charge, at2. payment, at2.copayrecd, at2.adjustment from archive_transactions at1
inner join archive_transactions at2 on at2.txnopaid = at1.trnsxno
where at1.date1 between :start_date and :end_date
and at1.cptcode between '90620' and '90746'
and at1.archive_flag = 1
and at2.archive_flag = 1
and at1.charge > '0.01'
)
group by  patno, cptcode, date1, trnsxno, charge
) aa
left outer join archive_transactions at3 on at3.trnsxno = aa.trnsxno
left outer join ins_carrier on ins_carrier.ins_carrier_code = at3.ins_carrier_code
left outer join
(
select cptcode, fee4 from cpt_codes where cptcode between '90620' and '90746'
order by cptcode ) cost on cost.cptcode = aa.cptcode


where (charge-p-c-a)<0.01
order by cptcode
```

SQL Code: MySQL

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

Copy

Office
Practicum

```
select aa.patno, aa.cptcode, aa.date1, at3.ins_carrier_code, ins_carrier.ins_fullname, aa.trnsxno, aa.charge, p+c as paid,
 cost.fee4
from
(
select t1.patno, cptcode,  date1, trnsxno, charge, sum(payment) as p, sum(copayrecd) as c, sum(adjustment) as a from
(
select at1.patno, at1.date1, at1.trnsxno, at1.cptcode,at1.charge, at2. payment, at2.copayrecd, at2.adjustment from arc
hive_transactions at1
inner join archive_transactions at2 on at2.txnopaid = at1.trnsxno
where at1.date1 between :start_date and :end_date
and at1.cptcode between '90620' and '90746'
and at1.archive_flag = 1
and at2.archive_flag = 1
and at1.charge > '0.01'
) t1
group by  patno, cptcode, date1, trnsxno, charge
) aa
left outer join archive_transactions at3 on at3.trnsxno = aa.trnsxno
left outer join ins_carrier on ins_carrier.ins_carrier_code = at3.ins_carrier_code
left outer join
(
select cptcode, fee4 from cpt_codes where cptcode between '90620' and '90746'
order by cptcode ) cost on cost.cptcode = aa.cptcode
where (aa.charge-p-c-a)<0.01
order by cptcode
```