

## Article

[Oliver Wilms](#) · Dec 24, 2021 4m read

## InterSystems SQL Outbound Adapter - How fast can it go?

One of my colleagues had developed an interface in Health Connect (HealthShare 2019.1) to add large amounts of data to an external SQL Server database. The data comes from many text files with delimited rows and data for one table per file. There is a business process to read a file line by line and send an Insert Request to an operation. The request contains an Insert statement like ' Insert into TABLE columns (col1, col2, ... colZ) values (val1, val2, ... valZ). ' The Health Connect operation utilizes Outbound SQL Adapter to insert one row into a table per request. You can probably imagine this process takes quite a long time. I believe it took about 4 hours to insert 200000 rows into one table. That is 50000 rows in one hour, which is 3600 seconds, or about 14 rows per second.

I saw on Open Exchange that Guillaume Rongier had developed Grongier.SQL.OutboundAdapter. I understand this code uses an Insert statement like ' Insert into TABLE columns (col1, col2, col3) values (?, ?, ?) ' paired with an array consisting of number of rows, number of columns per row, data values and data types. I had to tweak the code a bit to make it run in HealthShare 2019.1. I learned a lot about InterSystems SQL classes by debugging the code. It did not really work for me with reliability. I think I tried to insert a batch of 5 rows into two different tables. It worked with one table, but with the other table it inserted only one row and the return value indicated no row had been inserted.

I believe it was Dmitry Maslennikov who suggested to me that I could use an Insert statement like ' Insert into Table columns (col1, col2, col3) values (valA1, valA2, valA3), (valB1, valB2, valB3), (valC1, valC2, valC3). ' I refactored the business process to make one request per text file and the operation processed the stream line by line. I liked the idea to have all the values in the query statement, because I could use the standard InterSystems SQL Adapter with it. I observed that I could insert about 30 rows per second when I combined the values from two rows into one Insert statement. I was surprised that it did not make any significant difference if I inserted 20 columns per row or 90 columns per row.

It made sense to me that I could insert about twice as many rows per second when I inserted two rows per insert compared to inserting rows one by one. I wanted to see how fast I could insert rows if I combined more rows into one insert statement. Here are the results:

Batch Size	Insert Rate
3	45.90
4	57.03
5	62.73
6	78.42
7	94.10
8	101.73

When I tried to insert rows by ten at a time (Batch Size = 10), the processing time was the same as nine rows per insert. I was able to insert nearly 130 rows per second, almost ten times what the original process provided.

I also need to delete data before inserting new data. For some tables we delete all the rows, for some tables we delete only the rows that are being updated. I am not sure why we ' Delete TABLE where col1 = val1 ' and then ' Insert into TABLE columns (col1, ...) values (valNew1, ...). ' I think this could probably be accomplished quicker by an update statement. However, I was pleased to see I could delete thousands of rows in less than one second with a single delete statement like ' Delete TABLE where SiteID = 123. ' I have not yet collected metrics comparing deleting individual rows with inserting individual rows.

The refactored process refers to UPDATE logic when it uses an Insert statement like ' Insert into Table columns (col1, col2, col3) values (valA1, valA2, valA3), (valB1, valB2, valB3), (valC1, valC2, valC3) ' AFTER deleting any rows matching the rows to be inserted. We have identified key columns per table to identify records for the delete. Assuming a table uses key columns col1, col2, and col3, I will combine the key values from multiple rows using OR condition like shown below for a batch size of three rows:

Delete TABLE where (col1 = valA1 AND col2 = valA2 AND col3 = valA3) OR (col1 = valB1 AND col2 = valB2 AND col3 = valB3) OR (col1 = valC1 AND col2 = valC2 AND col3 = valC3)

[#Performance](#) [#SQL](#) [#Health Connect](#) [#HealthShare](#)

---

Source URL: <https://community.intersystems.com/post/intersystems-sql-outbound-adapter-how-fast-can-it-go>