

# Fixing Blue or White Boxes in SQL

Last Modified on 04/18/2024 1:14 pm EDT

## Overview

As OP has improved its SQL execution to run more efficiently, SQL queries must be tweaked to continue running. Users are noticing that they will see blue or white boxes in SQL report columns where text used to appear. You can fix the SQL code so that the text will appear as expected. If you try the steps below and still need further assistance, please contact OP Support.

Drag a column header here to group by that column

PATNO	FNAME	LNAME	DATE1	MDM_PLAN	CODES
			1/4/2021	[Blue Box]	[Blue Box]
			1/4/2021	[Blue Box]	[Blue Box]
			1/4/2021	[Blue Box]	[Blue Box]
			1/1/2021	[Blue Box]	[Blue Box]
			1/5/2021	[Blue Box]	[Blue Box]

## Edit the SQL

1. Within the Database Viewer, identify the column(s) that are affected. In the screen shot above, the **MDM\_PLAN** and **CODES** columns display the boxes. The other columns (PATNO, FNAME, LNAME, and DATE1) are not affected.
2. Examine the SQL code and find where the column(s) are first referenced. The two columns in the example are highlighted in red.

Query name: [new query] Save Print Close Run Extract

Description:

Results filename:

Save CSV Print Send

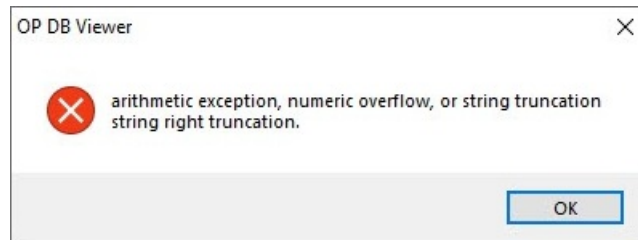
```
select enc_note.patno, fname, lname, enc_note.date1, mdm_plan, codes from enc_note
inner join register on register.patno = enc_note.patno
left outer join (select patno, date1, list(cptcode,',') as codes from archive_transactions where
date1 between :start_date and :end_date and archive_flag = 1 and cptcode not in ('1','2','3','4')
group by patno, date1) a on a.patno = enc_note.patno and a.date1 = enc_note.date1
```

3. For each affected column in your SQL (ex: **COLUMN NAME**), make the following change: **column\_name** becomes **cast(column\_name as char(2000)) as column\_name**. In the example above, the column **MDM\_PLAN** will be changed to **cast(mdm\_plan as char(2000)) as mdm\_plan** and the **CODES** column will be changed to **cast(codes as char(2000)) as codes**.

```
select enc_note.patno, fname, lname, enc_note.date1, cast(mdm_plan as char(2000)) as mdm_plan,
cast(codes as char(2000)) as codes from enc_note
inner join register on register.patno = enc_note.patno
left outer join (select patno, date1, list(cptcode,',') as codes from archive_transactions where date1
between :start_date and :end_date and archive_flag = 1 and cptcode not in ('1','2','3','4') group by
patno, date1) a on a.patno = enc_note.patno and a.date1 = enc_note.date1
where enc_note.date1 between :start_date and :end_date
```

4. Click **Run**. You should see text as expected in the columns that you changed. In this example, the **MDM\_PLAN** and **CODES** columns should now display data.
5. Click the **Save SQL Query text** button to save the edits you made to the SQL query.

**Note:** If you get the following error message when running the edited SQL, follow the steps below.



Raise the character limit of the cast for the affected columns in the SQL you are editing:

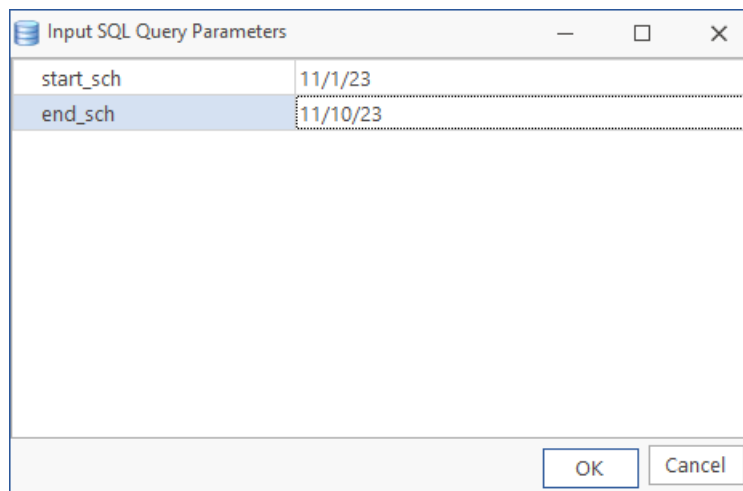
- Change **cast(column\_name as char(2000)) as column\_name** to **cast(column\_cast as char(5000)) as column\_name**.
- Continue raising the character limit as needed until the SQL runs with no error.

If you get any other message, check your work to look for any typos you may have made in your SQL edit.

## Date Format

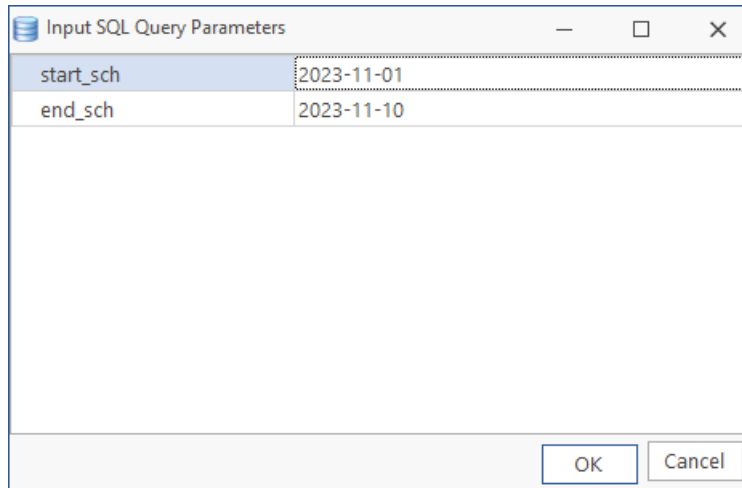
Keep in mind that, when entering run time dates in the Database Viewer, **MySQL requires a different date format than Firebird.**

In Firebird, while using the Database Viewer, we enter dates using the `mm/dd/yy` format. For example, if a manager wants to run a schedule report for the first 10 days of November 2023, she might enter these parameters at run time:



Parameter	Value
start_sch	11/1/23
end_sch	11/10/23

However, once the practice has converted to MySQL, dates must be entered using the `yyyy-mm-dd` format. Here's the same date range entered using the required MySQL format:



The screenshot shows a dialog box titled "Input SQL Query Parameters" with a table containing two rows of data. The first row has "start\_sch" and "2023-11-01". The second row has "end\_sch" and "2023-11-10". At the bottom right of the dialog are "OK" and "Cancel" buttons.

Parameter	Value
start_sch	2023-11-01
end_sch	2023-11-10

If a MySQL user forgets and uses the `mm/dd/yy` convention, the report will behave unexpectedly. Most commonly it will return no results. In other cases, it may return results outside of the date range requested.