

Article

[Eduard Lebedyuk](#) · Dec 19, 2016 3m read

Tips & Tricks - SQL queries management

If you have a lot of SQL queries to external systems (or even internal dynamic ones), you can encounter something like this:

```

Set SQL =      "SELECT "_"
                "c.cid                                AS Id, "_"
                "c.nid                                AS Nid, "_"
                "FROM_UNIXTIME(c.created)             AS Created, "_"
                "c.uid                                AS Uid, "_"
                "IFNULL(vv.average,0)                 AS AvgVote, "_"
                "IFNULL(vv.amount,0)                 AS VotesAmount, "_"
                "body.comment_body_value              AS Text, "_"
                "'comment'                             AS Type "_"
                "FROM comment AS c "_"
                "LEFT OUTER JOIN node ON node.nid = c.nid "_"
                "INNER JOIN field_data_field_forum_ref AS ref ON ref.entity_id = node
.nid "_"
                "LEFT OUTER JOIN field_data_comment_body AS body ON c.cid = body.ent
ty_id "_"
                "LEFT OUTER JOIN (SELECT entity_id , SUM(value) AS average, COUNT(1)
AS amount "_"
                                "FROM votingapi_vote "_"
                                "WHERE entity_type = 'comment' "_"
                                "GROUP BY entity_id) AS vv ON vv.entity_id = c.cid
 "_"
                ....
                + 100 more lines of SQL code omitted, but you get the idea

```

Looks awful?

Yes.

But that's not the only problem. Queries in class code also:

- Make it harder to read and understand source code
- No SQL code highlighting
- If you want to execute the query in any other tool, you'll need to copy it to external editor and do a bunch of Find&Replaces and probably reformat the query after that

I was plagued by this problem, before developing the solution that I'd like to share.

Class definition can have a [Query](#) element declared and it can be a basic class query or a custom class query ([more on that](#)). Basic class query text gets checked during compilation, but you can write anything in custom class query body and it would compile.

Here I created a separate class to hold custom queries and a method to return query text by it's name:

```

Class Utils.SQL [ Abstract ]
{

    /// Return query text by name. Removes linebreaks if removeNL is 1
    /// write ##class(Utils.SQL).getSQL("Comments")
    ClassMethod getSQL(name As %String, removeNL = {$$$NO}) As %String
    {
        #dim sc As %Status = $$$OK

        set query = ##class(%Dictionary.QueryDefinition).IDKEYOpen($classname(), name,, .
sc)
        throw:$$$ISERR(sc) ##class(%Exception.StatusException).CreateFromStatus(sc)

        set sql = query.SqlQuery
        set:(removeNL = $$$YES) sql = $replace(sql, $$$NL, " ")
        return sql
    }

    /// Get comments from mysql db
    Query Comments() As %Query
    {
        SELECT
            c.cid                AS Id,
            c.nid                AS Nid,
            FROM_UNIXTIME(c.created) AS Created,
            c.uid                AS Uid,
            IFNULL(vv.average, 0) AS AvgVote,
            IFNULL(vv.amount, 0) AS VotesAmount,
            body.comment_body_value AS Text,
            'comment'           AS Type
        FROM comment AS c
        LEFT OUTER JOIN node ON node.nid = c.nid
        INNER JOIN field_data_field_forum_ref AS ref ON ref.entity_id = node.nid
        LEFT OUTER JOIN field_data_comment_body AS body ON c.cid = body.entity_id
        LEFT OUTER JOIN (SELECT
                            entity_id,
                            SUM(value) AS average,
                            COUNT (1) AS amount
                        FROM votingapi_vote
                        WHERE entity_type = 'comment'
                        GROUP BY entity_id) AS vv ON vv.entity_id = c.cid
    WHERE
        node.status = 1
        AND node.type IN ('code_package', 'documentation', 'learning_track', 'video', 'post
        ')
    GROUP BY c.cid
    }
}

```

And in code, you can get the query text by executing:

```
Set SQL = ##class(Utils.SQL).getSQL("Comments")
```

Also, there is SQL code highlighting. And, of course, you can just copy/paste this sql to any other tool without any hassle.

How do you manage your SQL queries?

[GitHub](#).

[#Beginner](#) [#Caché](#) [#SQL](#)

Source URL: <https://community.intersystems.com/post/tips-tricks-sql-queries-management>