[MCSQ]: Multilingual Corpus of Survey Questionnaires[1]

Users' Manual

[1] The [MCSQ]: Multilingual Corpus of Survey Questionnaires is an open-access research resource.
If you use part of the code, data, manual and/or findings to inspire your own scientific work, please cite the article: Zavala-Rojas, D., Sorato, D., Hareide, L., & Hofland, K. (forthcoming 2020). The [MCSQ]: Multilingual Corpus of Survey Questionnaires. *Meta: Journal Des Traducteurs*.

SSHOC, "Social Sciences and Humanities Open Cloud", has received funding from the EU Horizon 2020 Research and Innovation Programme (2014-2020); H2020-INFRAEOSC-04-2018, under the agreement No. 823782

Table of Contents

Introduction

The [MCSQ]: Multilingual Corpus of Survey Questions (MCSQ) is a multilingual corpus of survey items from different studies. In its first version (Ada Lovelace), it comprises datasets from ESS[2](rounds 1 to 6) and EVS[3] (rounds 3 and 4) in the English source language and their translations into Catalan, Czech, French (produced for France, Switzerland, Belgium, and Luxembourg), German (produced for Austria, Germany, Switzerland, and Luxembourg), Norwegian, Portuguese, Spanish and Russian (produced for Israel, Latvia, Lithuania, Russia, Ukraine, Estonia).

This document comprises a guide to MCSQ users interested in making searches in the database. MCSQ was implemented as a PostgreSQL[4] Entity-Relationship (ER) database, therefore the searches must be made in SQL[5] language. In this document, we provide an overview of the database structure and provide a comprehensive explanation of how to do such search queries, in order to facilitate the usage of the database for users unfamiliar with SQL.

The structure of this document is organized as follows: firstly we present a brief overview of the MCSQ structure. Afterward, we show how to connect to the database where users can search for information. Then, we introduce basic concepts about the syntax of the search queries and show several examples of searches that can be made in the database. Lastly, we provide contact information to solve user inquiries.

[2] https://www.europeansocialsurvey.org/
[3] https://europeanvaluesstudy.eu/
[4] https://www.postgresql.org/
[5] SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system.

MCSQ Structure

The MCSQ is an Entity-Relationship (ER) database, in which the data stored is represented regarding *entities*, also referred to as *tables*, and the relationship between them. In order to make searches in the database, we use Structured Query Language (SQL), which is a domain-specific language for ER databases. To make a consult we construct a query that represents the information we want to retrieve from the tables contained in the database. The tables contain *attributes*, also referred to as *columns*. The columns contain the necessary information to describe the entity.

The main tables of the MCSQ database are the *survey_item* table and its substructures *introduction*, *instruction*, *request*, *response*, and *alignment*. The *survey_item* table holds the structural information about survey item observations (our unit of analysis), such as the sequence in which the substructures appear, which item types compose a given survey item, the item names, etc. The text for a given survey item can be found in the *introduction*, *instruction*, *request*, and *response* tables. Also, the *alignment* table shows the survey item texts, but only for correspondence with the source text purposes. Therefore, for a given language there may be less segments in the *alignment* table than in the *survey_item* table. The complete ER diagram of the MCSQ database, describing all of its tables and attributes is depicted in Figure 1:

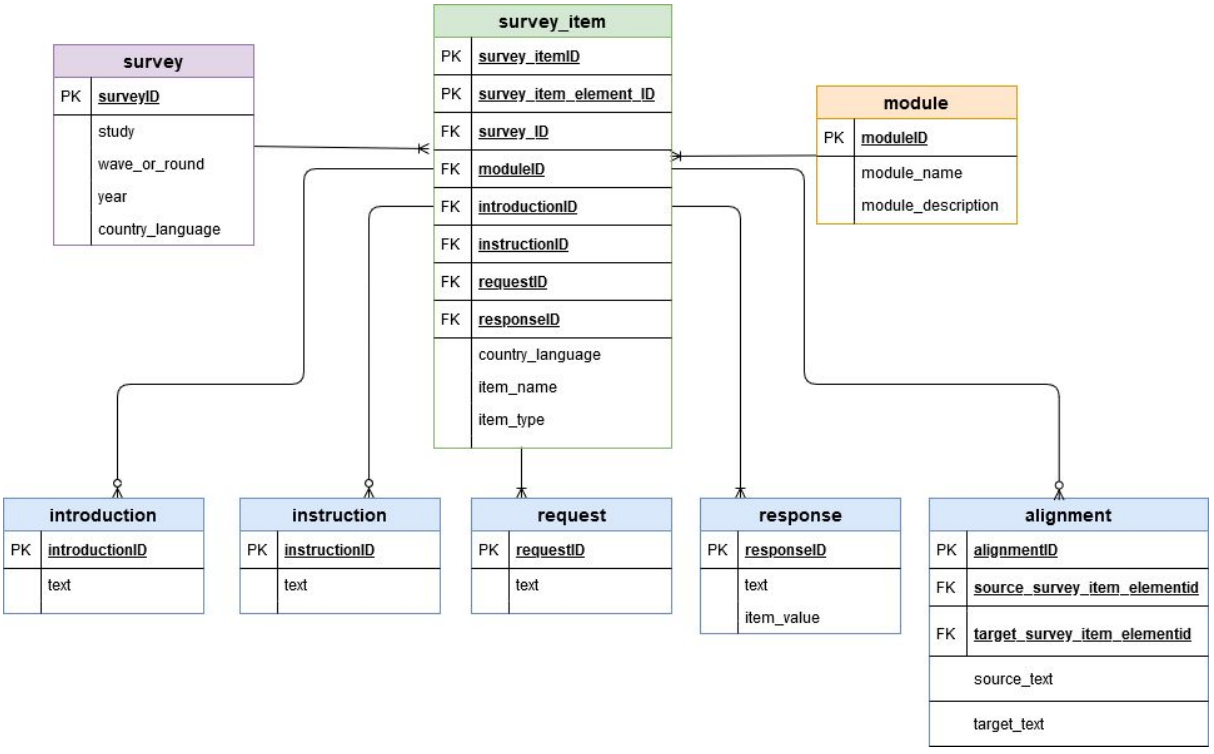


Figure 1: MCSQ ER diagram.

The MCSQ database was implemented in a way that for each language and study, only unique segment texts are included in the *introduction*, *instruction*, *request*, and *response* tables. There may be segments with small variations from one to another (differences in punctuation, word order, etc). Examples of several types of queries that can be done by the user are shown in the next sections.

Connecting to the database and running queries

To access the MCSQ database, enter in the page https://www.upf.edu/web/mcsq/ and click on 'Access Database':

Figure 2: Access to the MCSQ database in the MCSQ website.

There, you will find a login page that looks like this:

Figure 3: MCSQ login screen.

Credentials for the read only MCSQ user are available on request for the SSHOC team. Please send an email to the electronic addresses below to get the access keys.

Danielly.sorato@upf.edu and diana.zavala@upf.edu

Click on 'Servers' and you should see the structure of the objects contained in MCSQ:

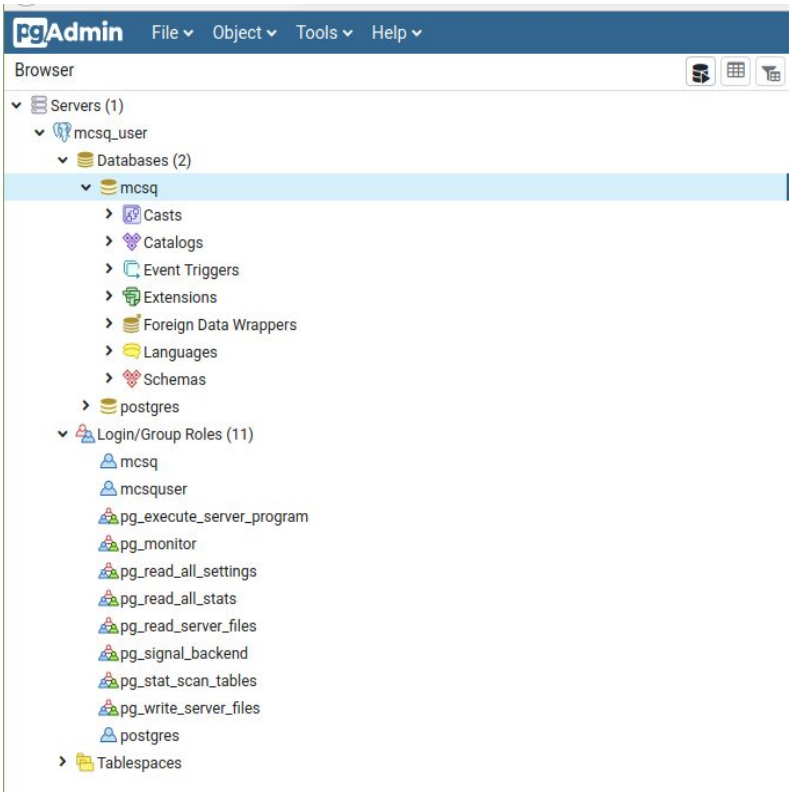


Figure 4: MCSQ database.

Then, click on the 'Query tool' button (right upper corner in Figure 3) and you will open the query editor, as shown below in Figure 4. Here, you write and execute SQL queries to retrieve the desired information from MCSQ. To execute the queries, click on the 'Run' (▷) button.

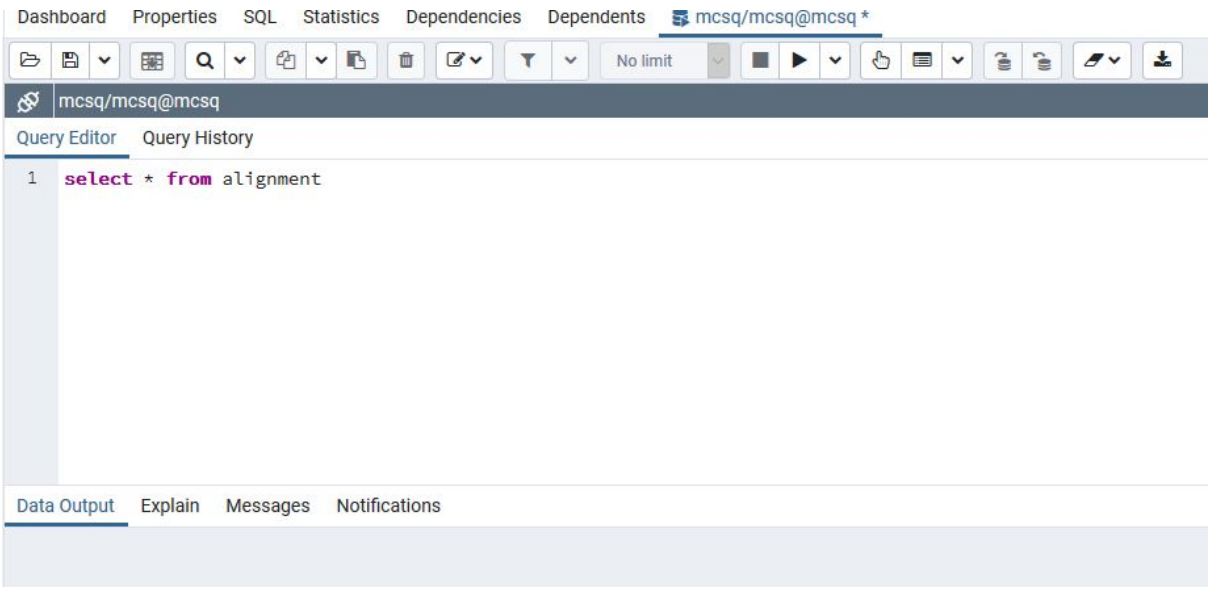


Figure 5: Query editor.

If you click on the dropdown ‘Schemas’ and then ‘public’, you will have a complete view of the objects that compose the MCSQ database, as depicted in Figure 5:

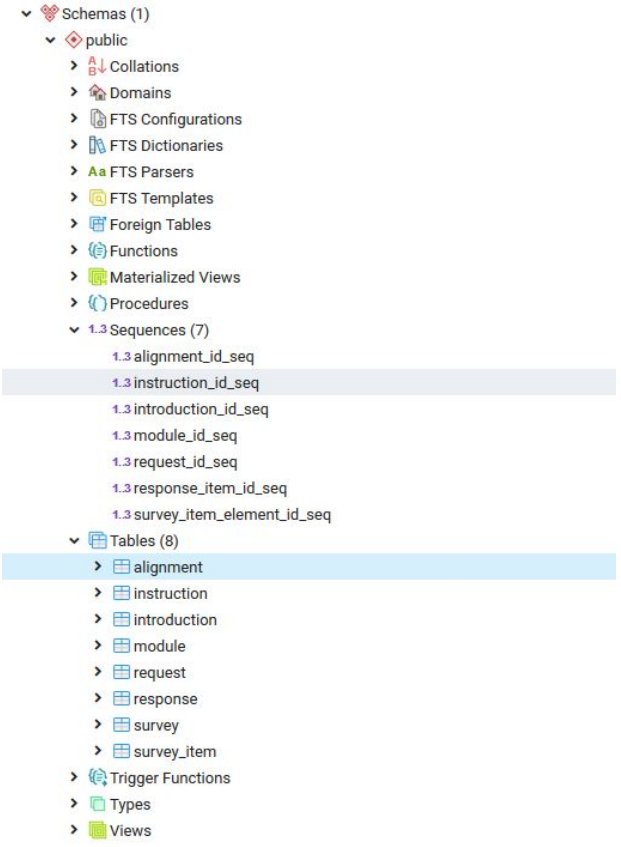


Figure 6: MCSQ database objects.

You can further explore the structure of individual objects by clicking in them. For instance, if you click on the table *alignment*, you can see its columns, as shown in Figure 6:

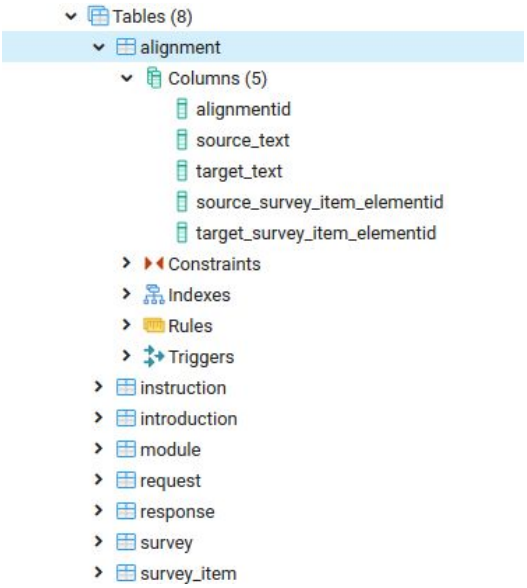


Figure 7: Columns of table *alignment*.

MCSQ Query Examples

MCSQ is available for community usage only for making search queries, i.e. it is not possible to alter the database. In order to retrieve data from the tables, it is necessary to make **SELECT** statements. The basic query syntax for **SELECT** statements works as it follows:

SELECT 'the names of columns you want to retrieve information from' **FROM** 'table name'.

For instance, suppose you want to retrieve the module name and the module description of the studies inserted in the database, you'd need to run the following query:

SELECT module_name, module_description **FROM** module;

	module_name character varying	module_description character varying
1	A	ESS module about Media; social trust
2	B	ESS module about Politics, including: political interest, efficacy, trust, electoral and other forms of participation, party allegiance, s...
3	C	ESS module about Subjective well-being and social exclusion; religion; perceived discrimination; national and ethnic identity
4	D	ESS R03 module about Timing of life; the life course; timing of key life events, attitudes to ideal age, youngest age and oldest age ...
5	E	ESS R03 module about Personal and social well-being, helping others, feelings in the last week, life satisfaction, satisfaction with ...
6	F	ESS module about Socio-demographic profile, including: Household composition, sex, age, type of area, Education & occupation d...
7	SUPP_B	ESS Supplementary questions with module B equivalents (from SQP database)
8	SUPP_E	ESS Supplementary questions with module E equivalents (from SQP database)
9	SUPP_H	ESS Test questions
10	SUPP_D	ESS Supplementary questions with module D equivalents (from SQP database)
11	SUPP_I	ESS R03 module about Interviewer self-completion questions
12	SUPP_G	ESS Human values scale
13	SUPP_A	ESS Supplementary questions with module A equivalents (from SQP database)

Figure 8: Results for the '**SELECT** module_name, module_description **FROM** module' query.

Where *module_name* and *module_description* are the names of the columns that represent the module name and the module descriptions in the database, and *module* is the name of the table that contains this information.

You can also retrieve the information of all columns contained in a single table by using the symbol '*':

```
SELECT * FROM module;
```

	moduleid [PK] integer	module_name character varying	module_description character varying
1	1	A	ESS module about Media; social trust
2	2	B	ESS module about Politics, including: political int...
3	3	C	ESS module about Subjective well-being and soc...
4	4	D	ESS R03 module about Timing of life; the life cou...
5	5	E	ESS R03 module about Personal and social well-...
6	6	F	ESS module about Socio-demographic profile, in...
7	7	SUPP_B	ESS Supplementary questions with module B eq...
8	8	SUPP_E	ESS Supplementary questions with module E eq...
9	9	SUPP_H	ESS Test questions
10	10	SUPP_D	ESS Supplementary questions with module D eq...
11	11	SUPP_I	ESS R03 module about Interviewer self-completi...
12	12	SUPP_G	ESS Human values scale
13	13	SUPP_A	ESS Supplementary questions with module A eq...
14	14	D	Immigration and asylum issues, including: attitu...
15	15	E	ESS module about Citizen involvement: including...
16	16	SUPP_I	ESS Interviewer questions
17	17	D	ESS R02 module about Health and care seekin h...

Figure 9: Results for the '**SELECT** * **FROM** module' query.

You can also include conditions to filter your results by adding the **WHERE** word in the query. We can see all studies that were inserted in the database that happened in the year 2006 by running the query:

SELECT * **FROM** survey **WHERE** year = 2006

	surveyid [PK] character varying	study character varying	wave_round character varying	year integer	country_language character varying
1	ESS_R03_2006_CAT_ES	ESS	R03	2006	CAT_ES
2	ESS_R03_2006_POR_PT	ESS	R03	2006	POR_PT
3	ESS_R03_2006_SPA_ES	ESS	R03	2006	SPA_ES
4	ESS_R03_2006_GER_AT	ESS	R03	2006	GER_AT
5	ESS_R03_2006_GER_CH	ESS	R03	2006	GER_CH
6	ESS_R03_2006_GER_DE	ESS	R03	2006	GER_DE
7	ESS_R03_2006_RUS_LV	ESS	R03	2006	RUS_LV
8	ESS_R03_2006_RUS_UA	ESS	R03	2006	RUS_UA
9	ESS_R03_2006_RUS_EE	ESS	R03	2006	RUS_EE
10	ESS_R03_2006_RUS_RU	ESS	R03	2006	RUS_RU
11	ESS_R03_2006_ENG_IE	ESS	R03	2006	ENG_IE
12	ESS_R03_2006_ENG_SOURCE	ESS	R03	2006	ENG_SOURCE
13	ESS_R03_2006_ENG_GB	ESS	R03	2006	ENG_GB

Figure 10: Results for the '**SELECT** * **FROM** survey **WHERE** year = 2006' query.

It is also possible to make filters using only parts of strings instead of the whole string[6] by adding the words **LIKE**/**ILIKE** (**ILIKE** is case insensitive) to the statement. For example, let's say you want to see for which rounds and studies there are survey items regarding the RUS_UA (Russian from Ukraine) language. This information is contained in the surveyid column, so we can make a query in the survey table, filtering the results in the surveyid column as it follows:

```
SELECT * FROM survey WHERE surveyid LIKE'%RUS_UA'
```

	surveyid [PK] character varying	study character varying	wave_round character varying	year integer	country_language character varying
1	ESS_R02_2004_RUS_UA	ESS	R02	2004	RUS_UA
2	ESS_R03_2006_RUS_UA	ESS	R03	2006	RUS_UA
3	ESS_R04_2008_RUS_UA	ESS	R04	2008	RUS_UA
4	ESS_R05_2010_RUS_UA	ESS	R05	2010	RUS_UA
5	ESS_R06_2012_RUS_UA	ESS	R06	2012	RUS_UA
6	EVS_R04_2008_RUS_UA	EVS	R04	2008	RUS_UA
7	EVS_R03_1999_RUS_UA	EVS	R03	1999	RUS_UA

Figure 11: Results for the '**SELECT** * **FROM** survey **WHERE** surveyid **LIKE** '%RUS_UA'' query.

Notice that it is necessary to always use a simple quote ('') when dealing with string values. Since surveyid is a type string column there are simple quotes around the values you want to retrieve for this column, like in '%RUS_UA'.

The '%' symbols determine to which sides you can have other characters. If you want to allow characters only on the left side of the string you can use '%word', to allow characters only on the right side of the string, you can use 'word%' and for both sides '%word%'. This works for all string values in the database, including text. For example, lets say you want to search for all request segments that have the word *democracy* contained in them. You could run the following query:

```
SELECT * FROM request WHERE text ILIKE '%democracy%'
```

[6] A **string** is a sequence of characters.

requestid [PK] integer	text character varying
17418	Political parties that wish to overthrow democracy should be banned
17411	How democracy works in Ireland
17792	how democracy works in Ireland
17809	And on the whole how satisfied are you with the way democracy works in Ireland?
17834	On the whole how satisfied are you with the way democracy works in Ireland?
17875	And on the whole, how satisfied are you with the way democracy works in country?
18196	And on the whole, how satisfied are you with the way democracy works in Ireland?
18270	Using this card, please tell me how important you think it is for democracy in general...
18276	And still thinking generally rather than about Ireland, how important do you think it is for dem...
18288	Now some questions about the same topics, but this time about how you think democracy is ...
18310	Then I'll ask how important you think your choice is for democracy in general.
18313	There are differing opinions on whether or not everyone should be free to express their politic...
18314	Which one of the statements on this card describes what you think is best for democracy in g...
18315	How important do you think it is for democracy in general that everyone is free to express the...
18317	How important do you think it is for democracy in general that those who hold extreme politic...
18321	How important do you think it is for democracy in general that the government changes its pl...
18323	How important do you think it is for democracy in general that the government sticks to its pl...
18326	Which option on this card describes what you think is best for democracy in general?

Figure 12: Results for the '**SELECT** * **FROM** request **WHERE** text **ILIKE** '%democracy%'' query.

It is also possible to combine words as it follows:

SELECT * **FROM** request **WHERE** text **ILIKE** '%democracy%political%'

requestid [PK] integer	text character varying
18315	How important do you think it is for democracy in general that everyone is free to express their political views openly, even if they are extreme?
18317	How important do you think it is for democracy in general that those who hold extreme political views are prevented from expressing them openly?

Figure 13: Results for the '**SELECT** * **FROM** request **WHERE** text **ILIKE** '%democracy%political%'' query

And to search for only parts of strings, as in:

SELECT * **FROM** request **WHERE** text **ILIKE** '%hap%'

	requestid [PK] integer	text character varying
1	17430	Taking all things together, how happy would you say you are?
2	17466	Feel happier?
3	17510	How often, if ever, have each of these things happened to you in the last five years?
4	17517	Using this card, how worried are you that things like this will happen to you?
5	18226	you were happy?
6	18438	...feel happier?
7	18825	In the next few questions, we would like you to think about what might happen during the next 12 months.
8	19436	How dissatisfied or satisfied were you with the way the police treated you the last time this happened?
9	19950	In the next few questions, we would like you to think about what might happen during the next 12 months.
10	20118	...you were happy?
11	20175	How dissatisfied or satisfied were you with time this happened?
12	20329	How dissatisfied or satisfied were you with the way the Gardaí treated you the last time this happened?
13	33111	Some people feel they have completely free choice and control over their lives, and other people feel that what they do has no real effect on what happen...
14	33219	Happy sexual relationship
15	33224	If someone says a child needs a home with both a father and a mother to grow up happily, would you tend to agree or disagree?
16	33231	marriage or a long-term stable relationship is necessary to be happy
17	33292	Please tell me for each one, if it were to happen whether you think it would be a good thing, a bad thing, or don't you mind?
18	34144	(For any event which has happened more than once, please tell me at what age you first experienced it.)

Figure 14: Results for the '**SELECT** * **FROM** request **WHERE** text **ILIKE** '%hap%'' query.

It is possible to add multiple conditions to filter your results by using the operator **AND**. For example, if you want to select survey items from questionnaires written in French that are instructions, you can run the following query:

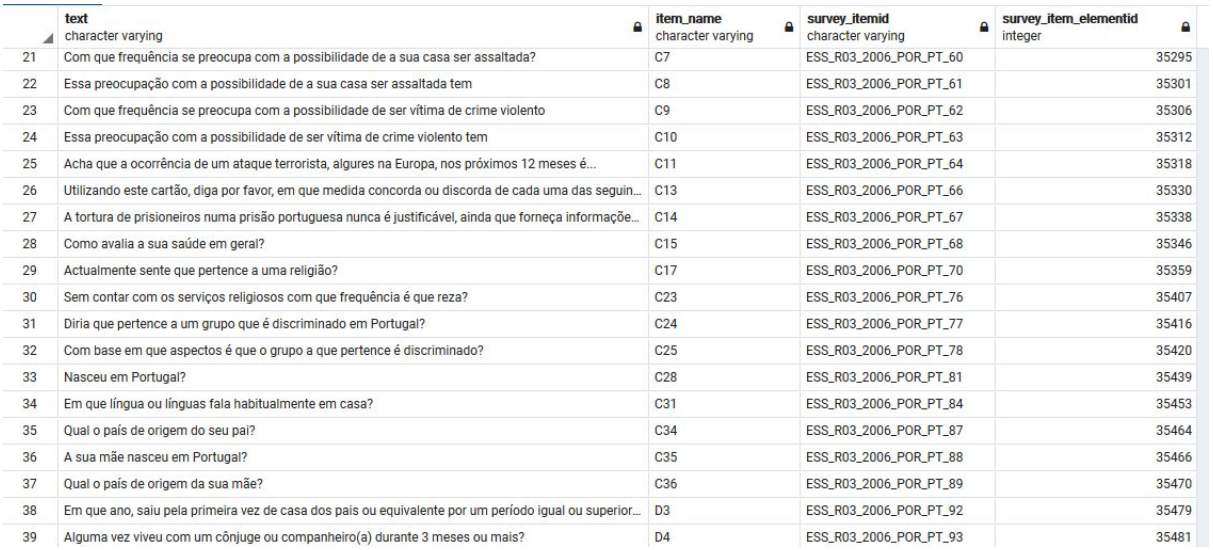
'**SELECT** * **FROM** survey_item **WHERE** surveyid **LIKE** '%FRE_%' **AND**
item_type = 'INSTRUCTION'

	survey_item_elementid [PK] integer	survey_itemid [PK] character varying	surveyid character varying	moduleid integer	requestid integer	responseid integer	response_item_id integer	instructionid integer	introductionid integer	country charact
1	237778	ESS_R06_2012_FRE_BE_0	ESS_R06_2012_FRE_BE	1	[null]	[null]	[null]	3547	[null]	FRE_BE
2	237780	ESS_R06_2012_FRE_BE_0	ESS_R06_2012_FRE_BE	1	[null]	[null]	[null]	3682	[null]	FRE_BE
3	237790	ESS_R06_2012_FRE_BE_1	ESS_R06_2012_FRE_BE	1	[null]	[null]	[null]	3683	[null]	FRE_BE
4	237792	ESS_R06_2012_FRE_BE_1	ESS_R06_2012_FRE_BE	1	[null]	[null]	[null]	3684	[null]	FRE_BE
5	237802	ESS_R06_2012_FRE_BE_2	ESS_R06_2012_FRE_BE	1	[null]	[null]	[null]	3550	[null]	FRE_BE
6	237817	ESS_R06_2012_FRE_BE_3	ESS_R06_2012_FRE_BE	1	[null]	[null]	[null]	3551	[null]	FRE_BE
7	237831	ESS_R06_2012_FRE_BE_4	ESS_R06_2012_FRE_BE	1	[null]	[null]	[null]	3552	[null]	FRE_BE
8	237833	ESS_R06_2012_FRE_BE_4	ESS_R06_2012_FRE_BE	1	[null]	[null]	[null]	3407	[null]	FRE_BE
9	237854	ESS_R06_2012_FRE_BE_7	ESS_R06_2012_FRE_BE	2	[null]	[null]	[null]	3553	[null]	FRE_BE
10	237975	ESS_R06_2012_FRE_BE_16	ESS_R06_2012_FRE_BE	2	[null]	[null]	[null]	3685	[null]	FRE_BE
11	237978	ESS_R06_2012_FRE_BE_16	ESS_R06_2012_FRE_BE	2	[null]	[null]	[null]	3686	[null]	FRE_BE

Figure 15: Results for the '**SELECT** * **FROM** survey_item **WHERE** surveyid **LIKE** '%FRE_%' **AND** item_type = 'INSTRUCTION' query.

One very important functionality of the ER database is to combine (join) tables. Let's say you want to see the texts of ESS requests, written in Portuguese. The table survey_item holds the structural information about items while the texts are in the request table. So in order to retrieve these segments, we combine the two tables as it follows:

'SELECT r.text, s.item_name, s.survey_itemid, s.survey_item_elementid FROM survey_item s, request r WHERE s.surveyid LIKE 'ESS%POR_%' AND r.requestid=s.requestid'

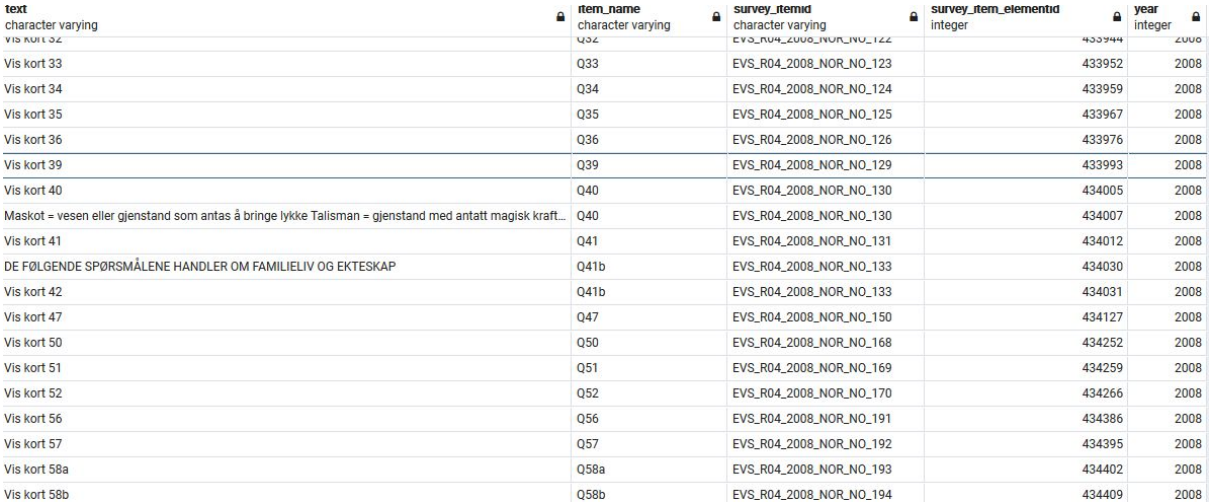


	text character varying	item_name character varying	survey_itemid character varying	survey_item_elementid integer
21	Com que frequência se preocupa com a possibilidade de a sua casa ser assaltada?	C7	ESS_R03_2006_POR_PT_60	35295
22	Essa preocupação com a possibilidade de a sua casa ser assaltada tem	C8	ESS_R03_2006_POR_PT_61	35301
23	Com que frequência se preocupa com a possibilidade de ser vítima de crime violento	C9	ESS_R03_2006_POR_PT_62	35306
24	Essa preocupação com a possibilidade de ser vítima de crime violento tem	C10	ESS_R03_2006_POR_PT_63	35312
25	Acha que a ocorrência de um ataque terrorista, algures na Europa, nos próximos 12 meses é...	C11	ESS_R03_2006_POR_PT_64	35318
26	Utilizando este cartão, diga por favor, em que medida concorda ou discorda de cada uma das seguin...	C13	ESS_R03_2006_POR_PT_66	35330
27	A tortura de prisioneiros numa prisão portuguesa nunca é justificável, ainda que forneça informaçõe...	C14	ESS_R03_2006_POR_PT_67	35338
28	Como avalia a sua saúde em geral?	C15	ESS_R03_2006_POR_PT_68	35346
29	Actualmente sente que pertence a uma religião?	C17	ESS_R03_2006_POR_PT_70	35359
30	Sem contar com os serviços religiosos com que frequência é que reza?	C23	ESS_R03_2006_POR_PT_76	35407
31	Diria que pertence a um grupo que é discriminado em Portugal?	C24	ESS_R03_2006_POR_PT_77	35416
32	Com base em que aspectos é que o grupo a que pertence é discriminado?	C25	ESS_R03_2006_POR_PT_78	35420
33	Nasceu em Portugal?	C28	ESS_R03_2006_POR_PT_81	35439
34	Em que língua ou línguas fala habitualmente em casa?	C31	ESS_R03_2006_POR_PT_84	35453
35	Qual o país de origem do seu pai?	C34	ESS_R03_2006_POR_PT_87	35464
36	A sua mãe nasceu em Portugal?	C35	ESS_R03_2006_POR_PT_88	35466
37	Qual o país de origem da sua mãe?	C36	ESS_R03_2006_POR_PT_89	35470
38	Em que ano, saiu pela primeira vez de casa dos pais ou equivalente por um período igual ou superior...	D3	ESS_R03_2006_POR_PT_92	35479
39	Alguma vez viveu com um cônjuge ou companheiro(a) durante 3 meses ou mais?	D4	ESS_R03_2006_POR_PT_93	35481

Figure 16: Results for the 'SELECT r.text, s.item_name, s.survey_itemid, s.survey_item_elementid FROM survey_item s, request r WHERE s.surveyid LIKE 'ESS%POR_%' AND r.requestid=s.requestid' query.

In another example, suppose you want to select Norwegian instructions from EVS questionnaires published from the year 2000 onwards:

'SELECT i.text, s.item_name, s.survey_itemid, s.survey_item_elementid, u.year FROM survey_item s, instruction i, survey u WHERE s.instructionid=i.instructionid AND s.surveyid = u.surveyid AND s.surveyid LIKE 'EVS%NOR_%' AND u.year >= 2000'



text character varying	item_name character varying	survey_itemid character varying	survey_item_elementid integer	year integer
Vis kort 32	Q32	EVS_R04_2008_NOR_NO_122	433944	2008
Vis kort 33	Q33	EVS_R04_2008_NOR_NO_123	433952	2008
Vis kort 34	Q34	EVS_R04_2008_NOR_NO_124	433959	2008
Vis kort 35	Q35	EVS_R04_2008_NOR_NO_125	433967	2008
Vis kort 36	Q36	EVS_R04_2008_NOR_NO_126	433976	2008
Vis kort 39	Q39	EVS_R04_2008_NOR_NO_129	433993	2008
Vis kort 40	Q40	EVS_R04_2008_NOR_NO_130	434005	2008
Maskot = vesen eller gjenstand som antas å bringe lykke Talisman = gjenstand med antatt magisk kraft...	Q40	EVS_R04_2008_NOR_NO_130	434007	2008
Vis kort 41	Q41	EVS_R04_2008_NOR_NO_131	434012	2008
DE FØLGENDE SPØRSMÅLENE HANDLER OM FAMILIELIV OG EKTESKAP	Q41b	EVS_R04_2008_NOR_NO_133	434030	2008
Vis kort 42	Q41b	EVS_R04_2008_NOR_NO_133	434031	2008
Vis kort 47	Q47	EVS_R04_2008_NOR_NO_150	434127	2008
Vis kort 50	Q50	EVS_R04_2008_NOR_NO_168	434252	2008
Vis kort 51	Q51	EVS_R04_2008_NOR_NO_169	434259	2008
Vis kort 52	Q52	EVS_R04_2008_NOR_NO_170	434266	2008
Vis kort 56	Q56	EVS_R04_2008_NOR_NO_191	434386	2008
Vis kort 57	Q57	EVS_R04_2008_NOR_NO_192	434395	2008
Vis kort 58a	Q58a	EVS_R04_2008_NOR_NO_193	434402	2008
Vis kort 58b	Q58b	EVS_R04_2008_NOR_NO_194	434409	2008

Figure 17: Results for the 'SELECT i.text, s.item_name, s.survey_itemid, s.survey_item_elementid, u.year FROM survey_item s, instruction i, survey u WHERE s.instructionid=i.instructionid AND s.surveyid = u.surveyid AND s.surveyid LIKE 'EVS%NOR_%' AND u.year >= 2000' query.

In order to combine tables, it is of utter importance to understand the columns of the tables, so we know in which piece of information is possible to combine tables in a way that makes sense. If you have doubts of the columns that compose a given table, please check the ER diagram in Figure 1, or investigate individual database objects as depicted in Figure 6.

There are many other functions useful for analysis, such as getting maximum/minimum values, data aggregation, order, etc. It is unfeasible to cover them all in this document, however, the user can find a comprehensive explanation of them in the PostgreSQL manuals [7]. In addition, there are many tutorial pages available online dedicated to show how to use these functions. In the next examples we show two of them that we consider very useful for MCSQ users.

It is possible to only count the results instead of listing them by adding the word **COUNT** to the query:

```
'SELECT COUNT(*) FROM survey_item WHERE surveyid LIKE '%FRE_%' AND
item_type = 'INSTRUCTION'
```

Data Output | Explain | Messages | Notifications

	count bigint
1	2997

Figure 18: Results for the 'SELECT COUNT(*) FROM survey_item WHERE surveyid LIKE '%FRE_%' AND item_type = 'INSTRUCTION' query.

You can also retrieve only distinct values by using the word **DISTINCT**. For example, if you want to see the all the target language questionnaires that have alignments with the source questionnaires in the database for ESS round 6, you can run the following query:

```
'SELECT DISTINCT s.surveyid FROM alignment a, survey_item s WHERE
a.target_survey_item_elementid = s.survey_item_elementid AND s.survey_itemid LIKE
'ESS_R06%'
```

[7] https://www.postgresql.org/docs/manuals/

	surveyid character varying
1	ESS_R06_2012_RUS_RU
2	ESS_R06_2012_FRE_BE
3	ESS_R06_2012_FRE_CH
4	ESS_R06_2012_POR_PT
5	ESS_R06_2012_CAT_ES
6	ESS_R06_2012_RUS_UA
7	ESS_R06_2012_RUS_IL
8	ESS_R06_2012_NOR_NO
9	ESS_R06_2012_GER_CH
10	ESS_R06_2012_ENG_IE
11	ESS_R06_2012_RUS_EE
12	ESS_R06_2012_ENG_GB
13	ESS_R06_2012_SPA_ES
14	ESS_R06_2012_FRE_FR
15	ESS_R06_2012_GER_DE
16	ESS_R06_2012_CZE_CZ

Figure 19: Results for the 'SELECT DISTINCT s.surveyid FROM alignment a, survey_item s WHERE a.target_survey_item_elementid = s.survey_item_elementid AND s.survey_itemid LIKE 'ESS_R06%' query.

Combinations between such functions are also possible:
'SELECT COUNT(DISTINCT s.surveyid) FROM alignment a, survey_item s WHERE a.target_survey_item_elementid = s.survey_item_elementid AND s.survey_itemid LIKE 'ESS_R03%'

Data Output | Explain | Messages | Notifications

	count bigint
1	16

Figure 20: Results for the 'SELECT COUNT(DISTINCT s.surveyid) FROM alignment a, survey_item s WHERE a.target_survey_item_elementid = s.survey_item_elementid AND s.survey_itemid LIKE 'ESS_R03%' query.

You can also order the results by ascending or descending values, by using **ORDER BY** column **ASC**/**DESC**. As an example, suppose you want to see the alignments of the ESS round 4 in French from Belgium in ascending order:

'**SELECT** a.alignmentid, a.source_text, a.target_text, s.country_language **FROM** alignment a, survey_item s **WHERE** a.target_survey_item_elementid = s.survey_item_elementid **AND** s.survey_itemid **LIKE** 'ESS%R04%FRE_BE%' **ORDER BY** a.alignmentid **ASC**'

	alignmentid integer	source_text character varying	target_text character varying	country_language character varying
1	1	And now a few questio...	Je vais maintenant vou...	FRE_BE
2	2	Please use this card.	Veuillez utiliser cette c...	FRE_BE
3	3	[null]	CARTE 17	FRE_BE
4	4	Taking all things togeth...	Tout bien considéré, da...	FRE_BE
5	5	Extremely unhappy	Très malheureux	FRE_BE
6	6	1	1	FRE_BE
7	7	2	2	FRE_BE
8	8	3	3	FRE_BE
9	9	4	4	FRE_BE
10	10	5	5	FRE_BE
11	11	6	6	FRE_BE
12	12	7	7	FRE_BE
13	13	8	8	FRE_BE
14	14	9	9	FRE_BE
15	15	Extremely happy	Très heureux	FRE_BE
16	16	(Don't know)	(Ne sait pas)	FRE_BE
17	17	[null]	CARTE 18	FRE_BE
18	18	Using this card, how of...	à l'aide de cette carte, ...	FRE_BE
19	19	Never	Jamais	FRE_BE

Figure 21: Results for the '**SELECT** a.alignmentid, a.source_text, a.target_text, s.country_language **FROM** alignment a, survey_item s **WHERE** a.target_survey_item_elementid = s.survey_item_elementid **AND** s.survey_itemid **LIKE** 'ESS%R04%FRE_BE%' **ORDER BY** a.alignmentid **ASC**' query.

It is also possible to aggregate numerical data by using **GROUP BY**. To exemplify an application, suppose you want to see the total number of segments per country/language from the EVS study:

'SELECT country_language, COUNT(survey_itemid) FROM survey_item WHERE survey_itemid LIKE 'EVS%' GROUP BY country_language

	country_language character varying	count bigint
1	CZE_CZ	5203
2	ENG_GB	5584
3	ENG_IE	4975
4	ENG_LU	2692
5	ENG_MT	2668
6	ENG_NIR	4849
7	ENG_SOURCE	29309
8	FRE_BE	5258
9	FRE_CH	2754
10	FRE_FR	4854
11	FRE_LU	4873
12	GER_AT	6041
13	GER_CH	2796
14	GER_DE	5658
15	GER_LU	2706
16	NOR_NO	2536
17	POR_LU	4832
18	POR_PT	5505
19	RUS_AZ	2696

Figure 22: Results for the 'SELECT country_language, COUNT(survey_itemid) FROM survey_item WHERE survey_itemid LIKE 'EVS%' GROUP BY country_language' query.

How to cite

The [MCSQ]: Multilingual Corpus of Survey Questionnaires is an open-access research resource. If you use part of the code, data, manual and/or findings to inspire your own scientific work, please cite the article:

Zavala-Rojas, D., Sorato, D., Hareide, L., & Hofland, K. (forthcoming 2020). The [MCSQ]: Multilingual Corpus of Survey Questionnaires. *Meta: Journal Des Traducteurs.*

```
@article{Zavala-Rojas,author = {Zavala-Rojas, Diana and Sorato,
Danielly and Hareide, Lidun and Hofland, Knut},journal = {Meta:
Journal des traducteurs},title = {{[MCSQ] Multilingual Corpus of
Survey Questionnaires}}}
```

Contact information

The PI of this project is Diana Zavala-Rojas, and this database was implemented and it is maintained by Danielly Sorato. For general enquiries, please send them to the electronic address diana.zavala@upf.edu. If you have queries about the (technical) usage/structure of MCSQ or the contents of this document, please send them to the electronic address danielly.sorato@upf.edu.