



# International Atomic Energy Agency

## X4Pro Retrieval by SQL sentences

Naohiko Otsuka

IAEA Nuclear Data Section

Note: The actual output of the SQL query may depend on the version of the X4Pro database. The outputs on my slides are based on **X4Pro/SQLite database ver.2011-11-28** distributed from the NDS website.



# Contents

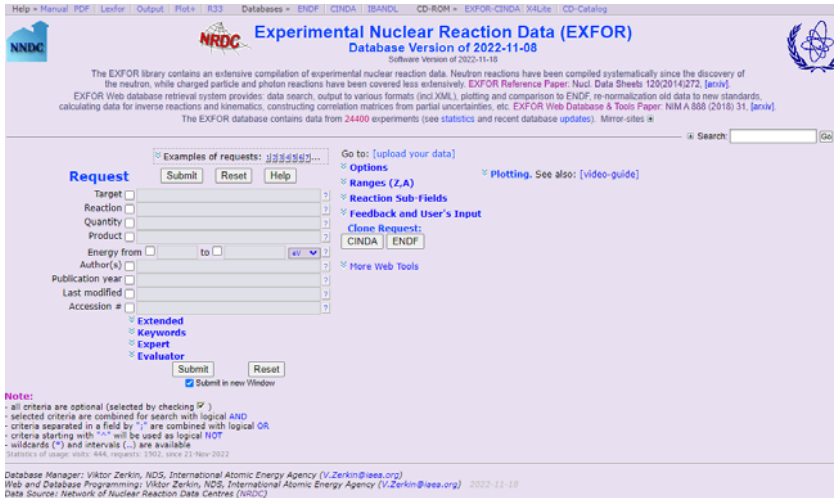
- Web retrieval and X4Pro retrieval
- Download and Browse X4Pro database file
- Introduction to SQL - Simple X4Pro retrievals
- Realistic samples of X4Pro retrieval and tabulation
  1. Generation of reference list
  2. Generation of 0<sup>th</sup> Legendre coefficient table



# Web Retrieval and X4Pro Retrieval

# Web Retrieval and X4Pro Retrieval

## NDS EXFOR web retrieval system



Help Manual PDF Lexfor Output Plot R33 Databases ENDF CINDA ISANDL CD-ROM EXFOR-CINDA X4LIFE CD-Catalog

**Experimental Nuclear Reaction Data (EXFOR)**  
Database Version of 2022-11-08  
Software Version of 2022-11-18

The EXFOR library contains an extensive compilation of experimental nuclear reaction data. Neutron reactions have been compiled systematically since the discovery of the neutron, while charged particle and photon reactions have been covered less extensively. EXFOR Reference Paper: Nucl. Data Sheets 120(2014)272, [enxv]  
EXFOR Web database retrieval system provides: data search, output to various formats (incl XML), plotting and comparison to ENDF, re-normalization old data to new standards, calculating data for inverse reactions and kinematics, constructing correlation matrices from partial uncertainties, etc. EXFOR Web Database & Tools Paper: NIM A 868 (2016) 31, [enxv]  
The EXFOR database contains data from 24400 experiments (see statistics and recent database updates). Mirror-sites: <#>

Examples of requests:  Go to:  [upload your data]

**Request**

Target  Reaction  Quantity  Product

Energy from  to

Author(s)  Publication year  Last modified  Accession #

Extended  Keywords  Expert  Evaluator

Submit in new Window

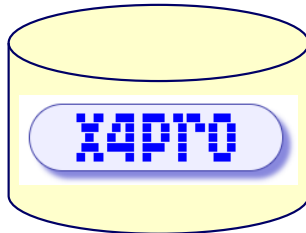
**Note:**  
- all criteria are optional (selected by checking )  
- selected criteria are combined for search with logical AND  
- criteria separated in a field by ";" are combined with logical OR  
- criteria starting with "~" will be used as logical NOT  
- wildcards (\*) and intervals (..) are available  
Statistics of usage: [Web](#), [XML](#), [Inquiries](#), [SQL](#), since 23-Nov-2022

Database Manager: Viktor Zerkin, NDS, International Atomic Energy Agency (V.Zerkin@iaea.org)  
Web and Database Programming: Viktor Zerkin, NDS, International Atomic Energy Agency (V.Zerkin@iaea.org) 2022-11-18  
Data Source: Network of Nuclear Reaction Data Centres (NNDC)

I daily use it for EXFOR management:

- check presence of a dataset
- access to publisher's website
- check duplication
- plot data together with ENDFs etc.

## X4Pro with SQL commands



I start to use this new option when I want to generate a machine readable table of data points, references etc.

# Download and Browse X4Pro Database

# How to Obtain X4Pro Database (DB) File

Download Nuclear Data Packages (<https://nds.iaea.org/cdroms/>)

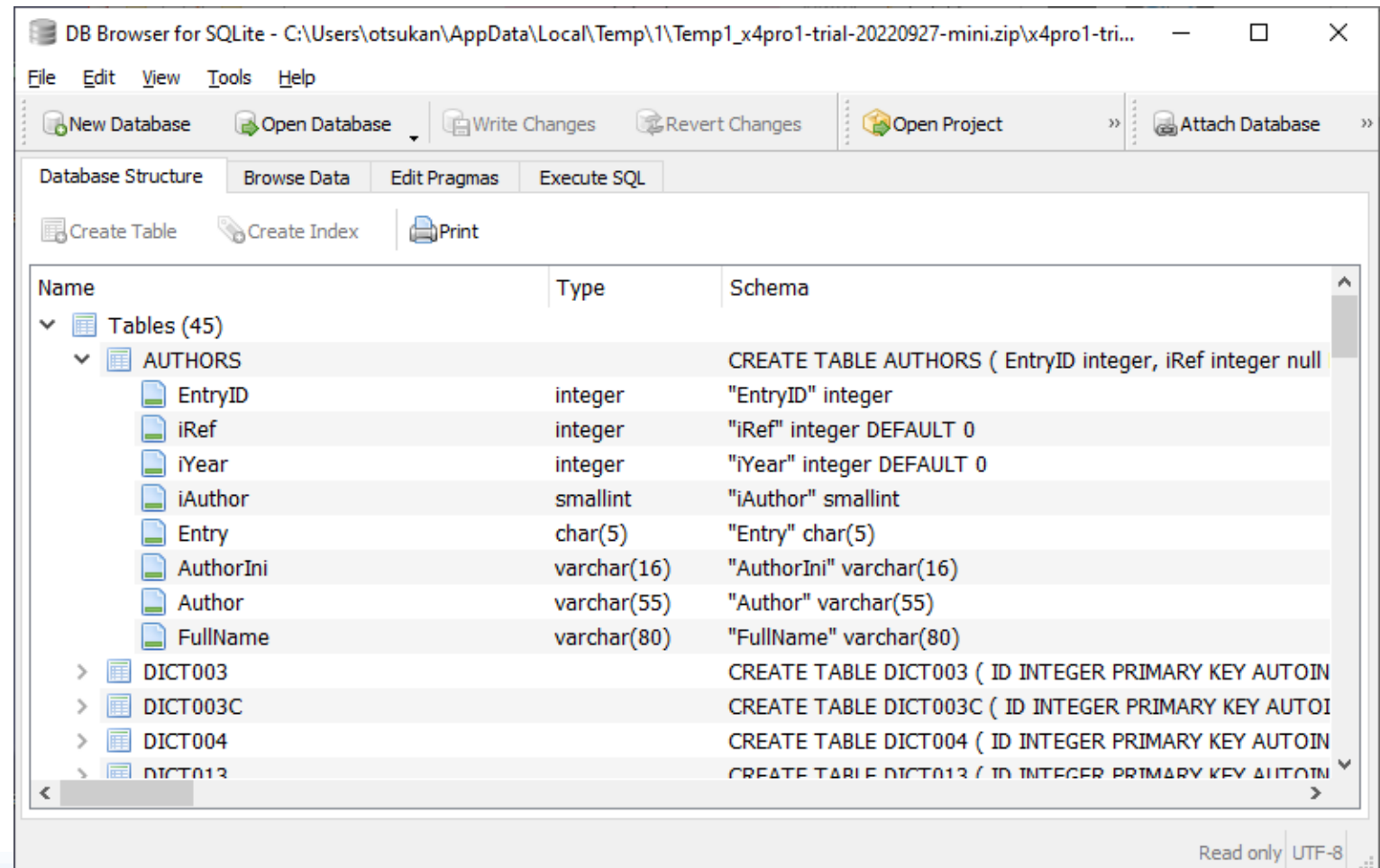
Pilot projects //under development		
29 <input type="checkbox"/>	<b>X4Apps</b> 2021-12-15	<b>Dec-2021</b> EXFOR-CINDA database (SQLite) with GUI (Java) retrieval system for Windows, Linux, MacOSX. Provides scripts and utility codes for EXFOR data search, retrieval and conversion to: Html, XML, JSON, C5. Includes Endver/GUI package integrated with Prepro, EXFOR, ZVView. No installation. <a href="#">[screen-shots]</a> This is united package of: "EXFOR-CINDA for Windows", "EXFOR for Applications", "Endver/GUI" <a href="#">Download</a> (zip, 523Mb)
30 <input type="checkbox"/>	<b>x4pro1 trial</b> 2022-09-27	<b>Sep-2022</b> X4Pro - universal, fully relational EXFOR database (professional edition). //Trial version. The package includes: ✓ X4Pro/SQLite database, ver.2022-08-29, 7.9Gb (incomplete) ✓ Fortran/C EXFOR data retrieval demo programs: SIG, DAE, SIG→C4 • new program recalculating EXFOR data: [DA,LEG/RS]&[SIG] → [DA] → [C4] <i>#new</i> ✓ Python demo programs, ver.2022-09-09: • retrieve and plot local EXFOR and remote ENDF data: SIG, DA, DE, DAE, [FY] • renormalize and correct EXFOR data: automatically, by user's and expert's codes • recalculate EXFOR data: [Ratio]→[SIG], [DA,LEG]→[SIG] • recalculate EXFOR data: [DA,LEG/RS] & [SIG] → [DA] <i>#new</i> • retrieve and plot original EXFOR covariance data En×En:[pdf1] Reac×Reac:[pdf2] • export JSON-X5Z from X4Pro to CouchDB (NoSQL database):[all_docs][doc1] <i>#new</i> Environment: Windows/Linux/MacOS + Python3 with plotly, matplotlib, requests, couchdb See: <a href="#">[page]</a> <a href="#">[readme]</a> <a href="#">[copyright]</a> <a href="#">[license]</a> <a href="#">[version]</a> <a href="#">Presentations:NRDC 2022, ND 2022</a> <a href="#">Download</a> (zip:2.1Gb) <a href="#">Download/mini-version</a> (zip:420Mb)

The zipped file include a database (db) file (e.g., x4sqlite1.db).  
N.B. This is a trial version and covers a part of EXFOR Master File.

# Browsing X4Pro DB File

We can open and browse the X4Pro db file by “DB Browser for SQLite” - Freely available from <https://sqlitebrowser.org/> for Windows, Mac, Linux, ...)

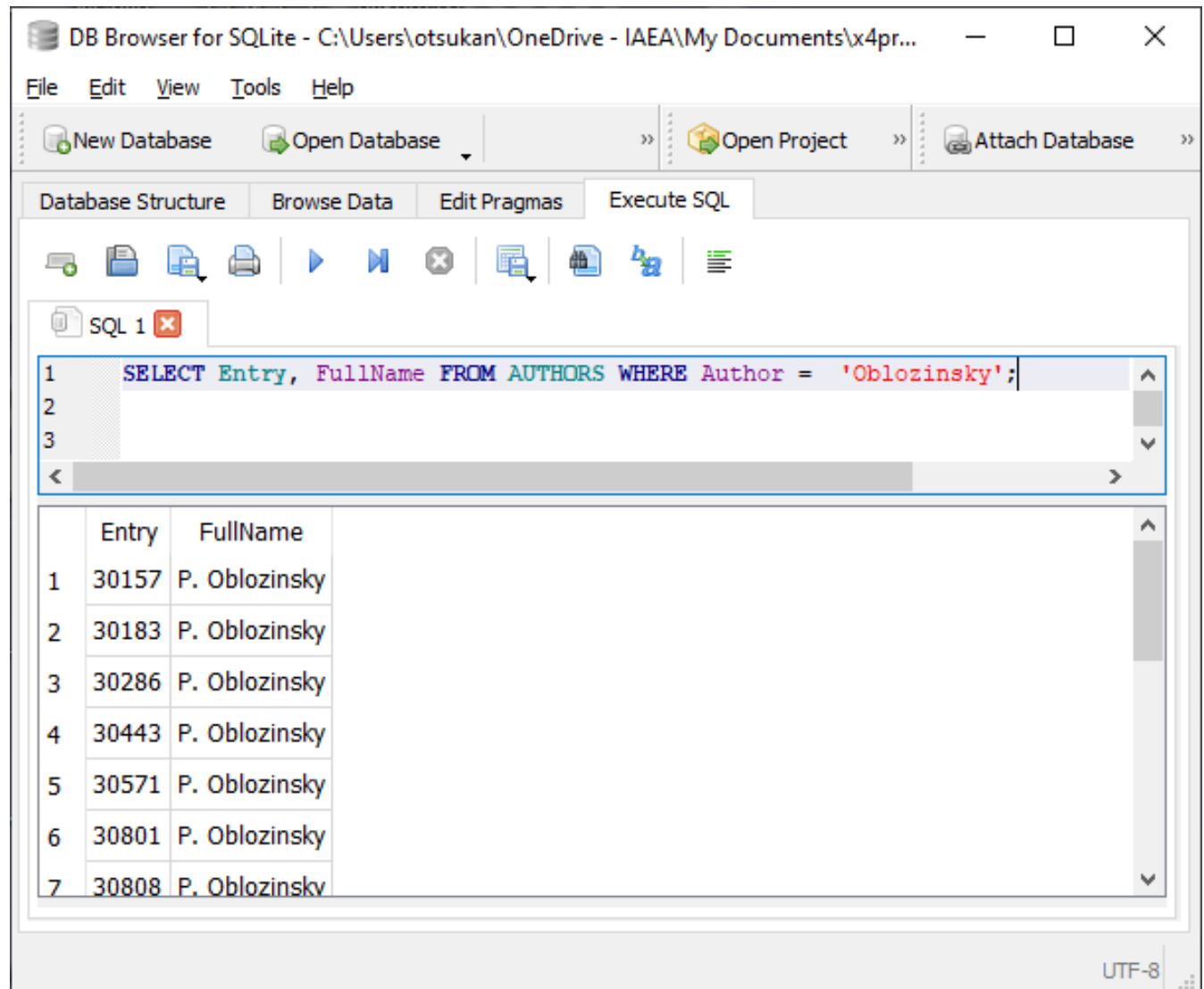
Database structure  
(List of tables and  
table columns)  
visualized on “DB  
Browser for SQLite”



# Execution of SQL Sentence - (1) DB Browser

SQL sentence (input) →

Output →



The screenshot shows the 'Execute SQL' tab in the DB Browser for SQLite application. The SQL query entered in the input field is: `SELECT Entry, FullName FROM AUTHORS WHERE Author = 'Oblozinsky';`. The results are displayed in a table below the query editor.

	Entry	FullName
1	30157	P. Oblozinsky
2	30183	P. Oblozinsky
3	30286	P. Oblozinsky
4	30443	P. Oblozinsky
5	30571	P. Oblozinsky
6	30801	P. Oblozinsky
7	30808	P. Oblozinsky

***Suggestion!*** Copy & paste the SQL sentences of my slides in the input field, then click ►. You can easily get the output from X4Pro db!



# Introduction to SQL

## - SQL Commands & Tables -

# Execution of SQL Sentence – (2) Command Mode

```
[ohtsuka@nds121] sqlite3 x4pro.db
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
```

```
sqlite> SELECT Entry, FullName FROM AUTHORS WHERE Author =
'Oblozinsky';
```

```
30157|P. Oblozinsky
30183|P. Oblozinsky
30286|P. Oblozinsky
30443|P. Oblozinsky
30571|P. Oblozinsky
...
```

A good solution for **batch processing** (e.g., Retrieve EXFOR entries having all former centre heads: Oblozinsky, Nichols, Hasegawa, Herman, Forrest, ...)

# Example of Tables Generated from X4Pro

## 1. Listing of full name under AUTHOR where the family name is *Schwerer*

### Request

```
SELECT Entry, FullName FROM AUTHORS WHERE Author = 'Schwerer';
```

### Output

```
20811|O.Schwerer  
20670|O.Schwerer
```

## 2. Listing of REACTION string where SF3 is *N+P* and SF6 is *SIG*

### Request

```
SELECT ReactrID, Code FROM REACSTR WHERE SF3 = 'N+P' AND SF6='SIG';
```

### Output

```
101450221|44-RU-96 (N,N+P) 43-TC-95-M, ,SIG  
101450231|44-RU-96 (N,N+P) 43-TC-95-G, ,SIG  
102140262|44-RU-102 (N,N+P) 43-TC-101, ,SIG
```

...

# Example of Tables Generated from X4Pro (Cont)

## 3. Listing of $^{16}\text{O}(n,\alpha)^{13}\text{C}$ cross section (SF1=8-O-16,SF2=N,SF3=A,SF6=SIG) data points

### Request (a long but still single SQL sentence)

```
SELECT
  x4pro_ds.reacode
  ,hdr_x.hdr, hdr_x.units, json_extract(x4.xdat,'$.'||hdr_x.hdr)
  ,hdr_y.hdr, hdr_y.units, json_extract(x4.xdat,'$.'||hdr_y.hdr)
  ,x4pro_ds.DatasetID
FROM
  x4pro_ds
  INNER JOIN x4pro_x4data AS x4      ON x4pro_ds.DatasetID = x4.DatasetID
  INNER JOIN x4pro_hdr      AS hdr_x ON x4pro_ds.DatasetID = hdr_x.DatasetID AND hdr_x.hdr = 'EN'
  INNER JOIN x4pro_hdr      AS hdr_y ON x4pro_ds.DatasetID = hdr_y.DatasetID AND hdr_y.hdr = 'DATA'
  INNER JOIN REACSTR        ON x4pro_ds.DatasetID = REACSTR.ReacodeID
WHERE
  REACSTR.SF1      = '8-O-16'
  AND REACSTR.SF2  = 'N'
  AND REACSTR.SF3  = 'A'
  AND REACSTR.SF5  = ''
  AND REACSTR.SF6  = 'SIG'
  AND REACSTR.SF7  = ''
  AND REACSTR.SF8  = ''
  AND REACSTR.SF9  = '';
```



# Example of Table Generated from X4Pro (Cont)

## 3. Listing of $^{16}\text{O}(n,\alpha)^{13}\text{C}$ cross section (SF1=8-O-16,SF2=N,SF3=A,SF6=SIG) data points

### Output

...

```
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV| 4.17|DATA|MB|108.0|21072002
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV| 4.22|DATA|MB|137.0|21072002
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV|14.8|DATA|MB|328.0|21343010
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV|12.3|DATA|MB|180.0|21343012
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV|13.0|DATA|MB|150.0|21343012
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV|14.1|DATA|MB|295.0|21343012
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV|16.0|DATA|MB|360.0|21343012
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV|16.9|DATA|MB|345.0|21343012
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV|18.0|DATA|MB|240.0|21343012
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV|19.5|DATA|MB|215.0|21343012
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV|14.9|DATA|MB|250.0|21461002
8-O-16 (N,A) 6-C-13 , , SIG|EN|MEV| 7.136|DATA|MB| 86.1|21474003
```

...

Convenient for further processing by computer programs

# Tables in X4Pro DB

X4Pro DB is a collection of tables (and indices).

## Example of tables

Table	Columns
AUTHORS	Entry, Author, FullName, ...
ENTRY	Entry, Author1Ini, Author1, Reference1, YearRef1, ...
KEYWORD	SubentID, Pointer, Code, FreeText, ...
REACSTR	SubentID, Pointer, ReacodeID, SF1, SF2, ...
REFERS	Entry, Reference, DOI, AuthorList, Title, DateRef, ...
x4pro_hdr	DatasetID, hdr, units, ...
X4pro_x4data	DatasetID, idat, xdata

**Note:** DatasetID (=ReacodeID) is the combination of SubentID and Pointer. e.g., “10001004” (=10001.004), “100010051” (=10001.005.1)



# Examples of Table Structure

**Table: AUTHORS**

...	Entry	...	Author	FullName
...	10002	...	Block	R.C.Block
...	10001	...	Hockenbury	R.W.Hockenbury
...	...	...	...	...

**Table: REACSTR**

...	ReacodeID	...	SF1	SF2	SF3	SF4	SF5	SF6	...
...	10001002	...	13-AL-27	N	G	13-AL-28		SIG	...
...	10001003	...	26-FE-0	N	G			SIG	...
...	...	...	...	...					



# Introduction to SQL

- **Simple X4Pro retrievals** -



# SQL Sentence

## Basic structure of a SQL sentence

```
SELECT X FROM Y WHERE Z;
```

X= Names of table columns for printing

Y= Names of tables

Z= Condition

A SQL sentence is terminated by a semicolon (;). Line break is allowed within a sentence.

## Example – Extraction from the table AUTHORS

```
SELECT Entry, FullName FROM AUTHORS WHERE Author = 'Schwerer';
```

*Print “Entry” and “FullName” columns of the table AUTHORS for the lines having “Schwerer” as the Author.*

# Retrieval with SQL Sentence

*Example 1: Extraction from the table AUTHORS (Schwerer is an author.)*

**SELECT Entry, FullName FROM AUTHORS WHERE Author = 'Schwerer';**

...	Entry	...	Author	FullName
...	...	...	...	...
...	20811	...	Winker-Rohatsch	M.Winkler-Rohatsch
...	20811	...	Schwerer	O.Schwerer
...	...	...	...	...
...	20670	...	Warhanek	H.Warhanek
...	20670	...	Schwerer	O.Schwerer
...	20670	...	Winkler	G.Winkler
...	...	...	...	...

This SQL sentence prints the following Entry and FullName:

20811 O.Schwerer

20670 O.Schwerer



# Retrieval with SQL Sentence (Cont)

*Example 2: Extraction from the table KEYWORD (A reactor at KAERI was used.)*

```
SELECT Entry, FreeText FROM KEYWORD
```

```
WHERE KeyWord = 'FACILITY' and Code = 'REAC,3KORKAE' ;
```

...	Entry	...	KeyWord	Code	...	FreeText
...	...	...	...	...	...	...
...	30829	...	REFERENCE	J,KPS,59,1932,2011	...	...
...	30829	...	FACILITY	REAC,3KORKAE	...	BNCT Facility of ...
...	30829	...	INC-SOURCE	THCOL	...	Thermal neutron ...
...	...	...	...	...	...	...

This SQL sentence prints the following Entry and FreeText:

```
30829  BNCT facility of HANARO research reactor
31570  HANARO
31596  HANARO facility
30839  BNCT port of HANARO reactor
```

# Retrieval with SQL Sentence (Cont)

*Example 3: Extraction from the table x4pro\_hdr (Quantity measured in femtbarn)*

```
SELECT DatasetID FROM x4pro_hdr WHERE hdr = 'DATA' and units = 'FB';
```

DatasetID	...	hdr	...	units	...
...	...		...	...	...
C2133002	...	EN		EV	...
C2133003	...	DATA		FB	...
C2133003	...	DATA-ERR	...	FB	...
...	...	...	...	...	...

This SQL sentence prints the following DatasetID:

C2133003  
E1920002  
E2055002  
E2388002

# Search with Two Tables

*Example: Datasets coded with REACTION SF3=ABS and heading=DATA-CM*

Table: REACSTR

ReacodeID	...	SF3	...
...	...	...	...
C0212007		P+A	...
C0212008	...	2A	...
C0212009	...	ABS	...
...	...	...	...

Table: x4pro\_hdr

DatasetID	...	hdr	...
...	...	...	...
C0212008	...	EN	...
C0212009	...	DATA-CM	...
C0212009	...	DATA-ERR	...
C0212009	...	EN-CM	...
...	...	...	...



This search condition can be expressed by

...

```
WHERE REACSTR.SF3 = 'ABS'  
and x4pro_hdr.hdr = 'DATA-CM'  
and REACSTR.ReacodeID = x4pro_hdr.DatasetID
```



# Search with Two Tables (Cont)

*Example: Datasets coded with REACTION SF3=ABS and heading=DATA-CM (Cont)*

```
SELECT REACSTR.ReacodeID, REACSTR.Code
FROM REACSTR, x4pro_hdr
WHERE
    REACSTR.SF3          = 'ABS'
    AND x4pro_hdr.hdr    = 'DATA-CM'
    AND REACSTR.ReacodeID = x4pro_hdr.DatasetID
```

This SQL sentence prints the following ReacodeID (of table “REACSTR”) and Code (of table “REACSTR”):

```
C0212009  8-0-16 (8-0-16, ABS) , , SIG
C0212010  8-0-16 (8-0-16, ABS) , , SIG
```

(Use of DATA-CM looks questionable. Probably DATA-CM must be DATA...)

# Search with Two Codes in Same Table Column

**Example: Entries with both “Semkova” and “Koning” as authors**

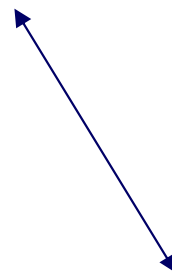
*(Cannot do it on typical web retrieval systems!)*

**Table: AUTHORS**

...	Entry	...	Author	...
...	...	...		
...	22877	...	Koning	...
...	22877	...	Tornin	...
...	22877	...	Capote	...
...	22877	...	Semkova	...
...	22878	...	Zmeskal	...
...	...	...	...	...

**Table: AUTHORS1 (copied from AUTHORS)**

...	Entry	...	Author	...
...	...	...		
...	22877	...	Koning	...
...	22877	...	Tornin	...
...	22877	...	Capote	...
...	22877	...	Semkova	...
...	22878	...	Zmeskal	...
...	...	...	...	...



# Search with Two Codes in Same Table Column (Cont)

*Example: Entries coded with both “Semkova” and “Koning” under AUTHOR (Cont)*

**Table: AUTHORS (original)**

...	Entry	...	Author	...
...	...	...		
...	22877	...	Koning	...
...	22877	...	Tornin	...
...	...	...	...	...



**Table: AUTHORS1 (an alias of AUTHOR)**

...	Entry	...	Author	...
...	...	...		
...	22877	...	Semkova	...
...	22878	...	Zmeskal	...
...	...	...	...	...

Original table (AUTHORS) can be joined with itself (AUTHORS1) based on the Entry by

...

FROM AUTHORS

**INNER JOIN AUTHORS as AUTHORS1 on AUTHORS.Entry = AUTHORS1.Entry**





# Search with Two Codes in Same Table Column (Cont)

*Example: Entries coded with both “Semkova” and “Koning” under AUTHOR (Cont)*

```
SELECT AUTHORS.Entry, AUTHORS.iYear
FROM AUTHORS
INNER JOIN AUTHORS as AUTHORS1 on AUTHORS.Entry = AUTHORS1.Entry
WHERE
    AUTHORS.Author = 'Semkova'
AND AUTHORS1.Author = 'Koning';
```

This SQL sentence prints the following Entry (of table AUTHORS) and iYear (of table AUTHORS):

```
22877 2007
22820 2004
22822 2010
22680 2009
22822 2004
```



# x4pro\_x4data: Data points in JSON

EN                    DATA  
EV                    NO-DIM  
0.1426E+06        25.  
0.1403E+06        30.  
...



Table: x4pro\_x4data

DatasetID	idat	xdat
10001002	1	{"DATA":25.0,"EN":1.426e+5}
10001002	2	{"DATA":25.0,"EN":1.426e+5}
...	...	...

JSON

*Example: Data points of 10001.002 where Einc > 130 keV*

```
SELECT  json_extract(xdat, '$.EN') as EN,  
        json_extract(xdat, '$.DATA')  
FROM    x4pro_x4data  
WHERE  
        DatasetID=10001002  
        AND EN >1.3E+05;
```

# x4pro\_x4data: Data points in JSON (Cont)

*Example: Data points of 10001.002 where Einc > 130 keV (Cont)*

```
SELECT  json_extract(xdat, '$.EN') as EN,  
        json_extract(xdat, '$.DATA') as DATA  
FROM    x4pro_x4data  
WHERE  
        DatasetID=10001002  
        AND EN >1.3E+05;
```

This SQL sentence prints the following EN and DATA values in the column xdat in the table x4pro\_x4data:

142600.0	25.0
140300.0	30.0
138100.0	26.0
135800.0	28.0
133700.0	29.0
131600.0	21.0

We can use values under any heading as a search condition (Cannot be done on the current web retrieval system!)

# Realistic Samples of X4Pro Retrieval and Tabulation

# Generation of Reference List

**Task:** Create a reference list for citation of data taken from 898 EXFOR entries for inclusion in a LaTeX manuscript.

/1/ Execute the following SQL sentence for 898 EXFOR entries (XXXX=10088, ..., T0297)

```
SELECT DISTINCT  
  Entry, Reference, AuthorsList, Title  
FROM REFERS  
WHERE  
  Entry=10088 /* 10888 is a variable of the SQL loop */  
AND  
  iReference=1 /* First code string under REFERENCE*/
```

(and do it for other Entry numbers. We need an external script writing and executing this SQL sentence in a language.)

# Generation of Reference List (Cont)

```
SELECT DISTINCT
  Entry, Reference, AuthorsList, Title
FROM REFERS
WHERE
  Entry=10888 /* 10888 is a variable of the SQL loop */
AND
  iReference=1 /* First code string under REFERENCE*/
```

This SQL sentence prints the following values extracted from table REFERS:

```
10088|J,PR/C,1,1233,1970|L.Husain,A.Bari,P.K.Kuroda|Neutron Activation ...
10145|J,PR/C,1,358,1970|W.D.Lu, N.Ranakumar, R.W.Fink|Activation Cross ...
10157|J,PR/C,3,629,197102|P.Venugopala Rao,R.E.Wood,J.M.Palms,R.W.Fink|Neutron.
10214|R,BRL-R-1491,1970|J.K.Temperley,D.E.Barnes|Neutron Activation ...
10244|J,NP/A,180,157,1972|A.K.Hankla,R.W.Fink,J.H.Hamilton|Neutron ...
...•
```

# Generation of Reference List (Cont)

/2/ Conversion of SQL print out to BibTeX by an additional script (in any language)

```
10088|J,PR/C,1,1233,1970|L.Husain,A.Bari,P.K.Kuroda|Neutron Activation ...
```

...



```
@article{10088,  
  author      = {Husain, L. and Bari, A. and Kuroda, P.K.},  
  title       = {Neutron Activation Cross Sections at 14.8 MeV for ...  
  journal     = {Phys.Rev. C},  
  year        = {1970},  
  volume      = {1},  
  number      = {},  
  pages       = {1233}  
}
```

**Note:** Dictionary 5 (Journal) is used for conversion of a code (e.g., PR/C → Phys.Rev.C).



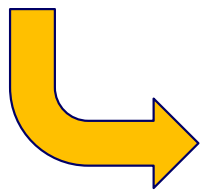
# Generation of Reference List (Cont)

/3/ Run BibTeX and LaTeX for typesetting (The title is removed by the style file chosen.)

→

```
@article{10088,  
  author      = {Husain, L. and Bari, A. and Kuroda, P.K.},  
  title       = {Neutron Activation Cross Sections at 14.8 MeV for ...  
  journal     = {Phys.Rev. C},  
  year        = {1970},  
  volume      = {1},  
  number      = {},  
  pages       = {1233}  
}
```

[146] K. Pepelnik, B. Anders, B. Bahai, M. Farooq, Tech. Rep. GKSS-86-E-29, GKSS, Geesthacht (1986).  
[147] L. Yingjun, Y. Weifan, High En.Nucl.Phys.Ch. 8 ((2)) (1984) 241.  
[148] L. Husain, A. Bari, P. Kuroda, Phys.Rev. C 1 (1970) 1233.  
[149] B. Erlandsson, A. Marcinkowski, K. Nilson, Phys.Scripta 19 ((3)) (1979) 251.  
[150] S. Ghorai, R. Vos, J. Cooper, W. Alford, Nucl.Phys.A 223 (1974) 118.  
[151] V. Levkovskii, G. Kovel'skaya, G. Vinitskaya, V. Stepanov, V. Sokol'skii, Sov.J.Nucl.Phys. 8 (1969) 4.  
[152] T. Thiep, N. Do, T. An, N. Son, Nucl.Phys.A 722 (2003) 568.



Attention! This procedure is correct when

1. the primary reference is appropriate for citation of the data used.
2. TITLE and AUTHOR are the title and author of the primary reference.





# Generation of 0<sup>th</sup> Legendre Coefficient Table

**Task:** Extraction of 0<sup>th</sup> Legendre coefficients for neutron elastic scattering

/1/ Execute the following SQL sentence (x4pro\_ds: a table for datasets)

## 1<sup>st</sup> part of the SQL sentence (SELECT)

```
SELECT
x4pro_ds.reacode
,hdr_x.hdr, hdr_x.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_x.hdr)
,hdr_y.hdr, hdr_y.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_y.hdr)
,x4pro_ds.DatasetID
```

This means to print

- REACTION code (reacode of table x4pro\_ds),
- Heading and unit (hdr and units of table hdr\_x which is an alias of x4pro\_hdr,
- Values of table xdat given with heading hdr\_x.hdr
- [Two similar lines for y]
- subentry #+pointer (DatasetID of table x4pro\_ds)

# Generation of 0<sup>th</sup> Legendre Coefficient Table (Cont)

/1/ Execute the following SQL sentence (cont)

## 2<sup>nd</sup> part of the SQL sentence (FROM)

```
FROM x4pro_ds
INNER JOIN x4pro_x4data      ON x4pro_ds.DatasetID = x4pro_x4data.DatasetID
INNER JOIN x4pro_hdr        AS hdr_x    ON x4pro_ds.DatasetID = hdr_x.DatasetID
                                   AND hdr_x.hdr = 'EN'
INNER JOIN x4pro_hdr        AS hdr_y    ON x4pro_ds.DatasetID = hdr_y.DatasetID
                                   AND hdr_y.hdr = 'DATA'
INNER JOIN REACSTR          ON x4pro_ds.DatasetID = REACSTR.ReacodeID
```

This means extraction lines from tables

- x4pro\_ds
- x4pro\_x4data, linked to the table x4pro\_ds by DatasetID
- hdr\_x (an alias of x4pro\_hdr), linked to x4pro\_ds by DatasetID if hdr = 'EN'
- [similar line for y]
- REACSTR, linked to the table x4pro\_ds by DatasetID/ReacodeID.

# Generation of 0<sup>th</sup> Legendre Coefficient Table (Cont)

/1/ Execute the following SQL sentence (cont)

## 3<sup>rd</sup> part of the SQL sentence (WHERE)

```
WHERE REACSTR.SF2 = 'N'  
      AND REACSTR.SF3 = 'EL'  
      AND REACSTR.SF5 = ''  
      AND REACSTR.SF6 = 'DA'  
      AND REACSTR.SF7 = ''  
      AND (REACSTR.SF8 = 'LEG' OR  
           REACSTR.SF8 = 'LEG/AV')  
      AND (REACSTR.SF9 = '' OR  
           REACSTR.SF9 = 'EXP')  
      AND (json_extract(x4pro_x4data.xdat, '$.NUMBER')=0 OR  
           json_extract(x4pro_x4data.xdat, '$.NUMBER-CM')=0);
```

This sets the following conditions:

- SF2 of table REACSTR must be N in table REACSTR
- [Similar lines for SF3 to SF9]
- NUMBER=0 or NUMBER-CM=0 in table x4pro\_x4data.

# Generation of 0<sup>th</sup> Legendre Coefficient Table (Cont)

/1/ Execute the following SQL sentence (cont)

## Full SQL sentence (SELECT+FROM+WHERE)

```
SELECT
x4pro_ds.reacode
,hdr_x.hdr, hdr_x.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_x.hdr)
,hdr_y.hdr, hdr_y.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_y.hdr)
,x4pro_ds.DatasetID
FROM x4pro_ds
INNER JOIN x4pro_x4data          ON x4pro_ds.DatasetID = x4pro_x4data.DatasetID
INNER JOIN x4pro_hdr      AS hdr_x  ON x4pro_ds.DatasetID = hdr_x.DatasetID
                                AND hdr_x.hdr  ='EN'
INNER JOIN x4pro_hdr      AS hdr_y  ON x4pro_ds.DatasetID = hdr_y.DatasetID
                                AND hdr_y.hdr  ='DATA'
INNER JOIN REACSTR        ON x4pro_ds.DatasetID = REACSTR.ReacodeID
WHERE REACSTR.SF2      = 'N'
    AND REACSTR.SF3      = 'EL'
    AND REACSTR.SF5      = ''
    AND REACSTR.SF6      = 'DA'
    AND REACSTR.SF7      = ''
    AND (REACSTR.SF8      = 'LEG' OR
         REACSTR.SF8 = 'LEG/AV')
    AND (REACSTR.SF9      = '' OR
         REACSTR.SF9 = 'EXP')
    AND (json_extract(x4pro_x4data.xdat,'$.NUMBER')=0 OR
         json_extract(x4pro_x4data.xdat,'$.NUMBER-CM')=0);
```

# Generation of 0<sup>th</sup> Legendre Coefficient Table (Cont)

/1/ Execute the following SQL sentence (cont)

## Full SQL sentence (SELECT+FROM+WHERE)

```
SELECT
x4pro_ds.reacode
,hdr_x.hdr, hdr_x.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_x.hdr)
,hdr_y.hdr, hdr_y.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_y.hdr)
,x4pro_ds.DatasetID
...
AND (json_extract(x4pro_x4data.xdat,'$.NUMBER')=0 OR
      json_extract(x4pro_x4data.xdat,'$.NUMBER-CM')=0);
```

This SQL sentence prints the following lines extracted from tables x4pro\_ds, x4pro\_hdr and x4pro\_x4dat:

```
5-B-11 (N,EL) 5-B-11, ,DA, ,LEG      EN MEV 0.075      DATA B/SR 0.354      10093139
5-B-11 (N,EL) 5-B-11, ,DA, ,LEG      EN MEV 0.1        DATA B/SR 0.352      10093139
5-B-11 (N,EL) 5-B-11, ,DA, ,LEG      EN MEV 0.125     DATA B/SR 0.347      10093139
5-B-11 (N,EL) 5-B-11, ,DA, ,LEG      EN MEV 0.15      DATA B/SR 0.337      10093139
5-B-11 (N,EL) 5-B-11, ,DA, ,LEG      EN MEV 0.175     DATA B/SR 0.326      10093139
5-B-11 (N,EL) 5-B-11, ,DA, ,LEG      EN MEV 0.2       DATA B/SR 0.312      10093139
```

...

# Generation of 0<sup>th</sup> Legendre Coefficient Table (Cont)

/2/ Conversion of SQL print out to more convenient format by a script (in any language)

```
5-B-11 (N,EL) 5-B-11 , ,DA , ,LEG      EN MEV 0.075          DATA B/SR 0.354          10093139
5-B-11 (N,EL) 5-B-11 , ,DA , ,LEG      EN MEV 0.1           DATA B/SR 0.352          10093139
5-B-11 (N,EL) 5-B-11 , ,DA , ,LEG      EN MEV 0.125         DATA B/SR 0.347          10093139
5-B-11 (N,EL) 5-B-11 , ,DA , ,LEG      EN MEV 0.15          DATA B/SR 0.337          10093139
5-B-11 (N,EL) 5-B-11 , ,DA , ,LEG      EN MEV 0.175         DATA B/SR 0.326          10093139
5-B-11 (N,EL) 5-B-11 , ,DA , ,LEG      EN MEV 0.2           DATA B/SR 0.312          10093139
```

...

```
→
#DATASET      10093139
#REACTION     5-B-11 (N,EL) 5-B-11 , ,DA , ,LEG
#             !! 0th order Legendre coefficients multiplied by 4pi during conversion to C4!!
# Prj Targ M MF MT PXC  Energy  Data      EntrySubP
#----><----->o<-><-->ooo<-----><-----> <----><->o
  1  5011   3   2    7.5000+4  4.4485+0  10093139
  1  5011   3   2    1.0000+5  4.4234+0  10093139
  1  5011   3   2    1.2500+5  4.3605+0  10093139
  1  5011   3   2    1.5000+5  4.2349+0  10093139
  1  5011   3   2    1.7500+5  4.0966+0  10093139
  1  5011   3   2    2.0000+5  3.9207+0  10093139
```

...

Example of conversion to C4 style.



# LEFT JOIN

The following part of the SQL sentence for 0<sup>th</sup> Legendre coefficients is for extraction of EXFOR data lines with EN as a heading and set the value as `hdr_x`:

```
FROM x4pro_ds
...
INNER JOIN x4pro_hdr AS hdr_x ON x4pro_ds.DatasetID = hdr_x.DatasetID
AND hdr_x.hdr = 'EN'
```

A extension of this sentence with **LEFT JOIN** allows extraction of EXFOR data lines with EN, EN-MIN or EN-MAX and set their values as `hdr_x1`, `hdr_x2` or `hdr_x3`:

```
INNER JOIN x4pro_hdr AS hdr_x ON x4pro_ds.DatasetID = hdr_x.DatasetID
AND (hdr_x.hdr='EN' or hdr_x.hdr='EN-MIN' or hdr_x.hdr='EN-MAX')
LEFT JOIN x4pro_hdr AS hdr_x1 ON x4pro_ds.DatasetID = hdr_x1.DatasetID
AND hdr_x1.hdr = 'EN'
LEFT JOIN x4pro_hdr AS hdr_x2 ON x4pro_ds.DatasetID = hdr_x2.DatasetID
AND hdr_x2.hdr = 'EN-MIN'
LEFT JOIN x4pro_hdr AS hdr_x3 ON x4pro_ds.DatasetID = hdr_x3.DatasetID
AND hdr_x3.hdr = 'EN-MAX'
```

# LEFT JOIN (cont)

Then the previous SQL sentence for extraction of 0<sup>th</sup> Legendre coefficients can be extended to

```
SELECT
x4pro_ds.reacode
,hdr_x1.hdr, hdr_x1.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_x1.hdr)
,hdr_x2.hdr, hdr_x2.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_x2.hdr)
,hdr_x3.hdr, hdr_x3.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_x3.hdr)
,hdr_y.hdr, hdr_y.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_y.hdr)
,x4pro_ds.DatasetID

...
FROM x4pro_ds
INNER JOIN x4pro_x4data      ON x4pro_ds.DatasetID = x4pro_x4data.DatasetID
INNER JOIN x4pro_hdr AS hdr_x  ON x4pro_ds.DatasetID = hdr_x.DatasetID
                                AND (hdr_x.hdr='EN' or hdr_x.hdr='EN-MIN' or hdr_x.hdr='EN-MAX')
LEFT JOIN x4pro_hdr AS hdr_x1  ON x4pro_ds.DatasetID = hdr_x1.DatasetID
                                AND hdr_x1.hdr = 'EN'
LEFT JOIN x4pro_hdr AS hdr_x2  ON x4pro_ds.DatasetID = hdr_x2.DatasetID
                                AND hdr_x2.hdr = 'EN-MIN'
LEFT JOIN x4pro_hdr AS hdr_x3  ON x4pro_ds.DatasetID = hdr_x3.DatasetID
                                AND hdr_x3.hdr = 'EN-MAX'
INNER JOIN x4pro_hdr AS hdr_y   ON x4pro_ds.DatasetID = hdr_y.DatasetID
                                AND hdr_y.hdr = 'DATA'
```





# LEFT JOIN (cont)

The SQL sentence for 0<sup>th</sup> Legendre coefficients extraction extended with LEFT JOIN gives the following SQL output:

```
...
... EN MEV 1.5                                DATA B/SR  0.332      40179022
... EN MEV 1.5                                DATA B/SR  0.362      40179025
...                                     EN-MIN MEV 0.4      EN-MAX MEV 0.85      DATA MB/SR 3.814      40738003
... EN KEV 55.0                                DATA MB/SR 1026.0     40924013
... EN KEV 144.0                              DATA MB/SR 841.0      40924016
...
```

I would convert this red line to the following line similar to C4 for further processing:

```
# Prj Targ M MF MT PXC Energy dEnergy Data dData EntrySubP
#----><----->o<--><--->ooo<-----><-----><-----><-----> <---><-->o
      1 26000 3 2 6.2500+5 2.2500+5 4.7928-2 4.9009-4 40738 3
```

(N.B.  $6.25E+5=(0.85+0.4)/2*1E+06$ ,  $2.25E+05=(0.85-0.4)/2*1E+06$ )



# Trouble due to Repetition of Same Heading

SQL sentence for extraction of EN, DATA and ERR-T from 20338.013:

(Note this dataset is NOT included in the X4Pro database Ver. 2022-11-28 released from the NDS website.)

```
SELECT hdr_x.hdr,  hdr_x.units,  json_extract(x4pro_x4data.xdat,'$.'||hdr_x.hdr)
      ,hdr_y.hdr,  hdr_y.units,  json_extract(x4pro_x4data.xdat,'$.'||hdr_y.hdr)
      ,hdr_dy.hdr, hdr_dy.units, json_extract(x4pro_x4data.xdat,'$.'||hdr_dy.hdr)
      ,x4pro_ds.DatasetID

FROM x4pro_ds
INNER JOIN REACSTR          ON REACSTR.ReacodeID      = x4pro_ds.DatasetID
INNER JOIN x4pro_x4data     ON x4pro_x4data.DatasetID = x4pro_ds.DatasetID
INNER JOIN x4pro_hdr       ON x4pro_hdr.DatasetID    = x4pro_ds.DatasetID
INNER JOIN REACODE         ON REACODE.ReacodeID      = x4pro_ds.DatasetID
INNER JOIN SUBENT          ON REACODE.SubentID       = SUBENT.SubentID
INNER JOIN x4pro_hdr as hdr_x ON REACODE.ReacodeID  = hdr_x.DatasetID AND hdr_x.hdr = 'EN'
INNER JOIN x4pro_hdr as hdr_y ON REACODE.ReacodeID  = hdr_y.DatasetID AND hdr_y.hdr = 'DATA'
INNER JOIN x4pro_hdr as hdr_dy ON REACODE.ReacodeID = hdr_dy.DatasetID AND hdr_dy.hdr='ERR-T'

WHERE REACODE.ReacodeID = '20338013'
```



# Trouble due to Repetition of Same Heading (Cont)

Extraction of EN, DATA and ERR-T from 20338.013 by SQL

EN	MEV	13.41	DATA	B	-0.0017	<u>ERR-T</u>	<u>PER-CENT</u>	25.4	20338013
EN	MEV	13.41	DATA	B	-0.0017	<u>ERR-T</u>	B	25.4	20338013

Namely the first and second data points have the same EN and DATA but differentiated by ERR-T=25.4% and **25.4 b**. How do they appear?

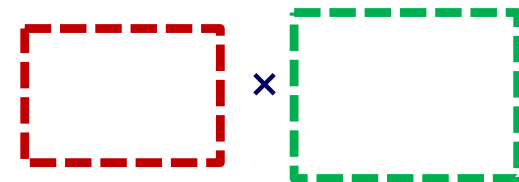
```
SUBENT      20338013   20080829
BIB          7         23
REACTION    (42-MO-92 (N,A) 40-ZR-89-M, , SIG)
```

```
COMMON      8         6
ERR-T
PER-CENT
25.4
ENDCOMMON
```

```
DATA        5         21
EN          +EN-RSL   -EN-RSL   DATA
MEV         MEV      MEV         B
1.3410E+01 1.7000E-01 2.1000E-01 -1.7000E-03 1.6000E-03
1.3510E+01 1.4000E-01 1.8000E-01 4.8000E-03 9.0000E-04
```

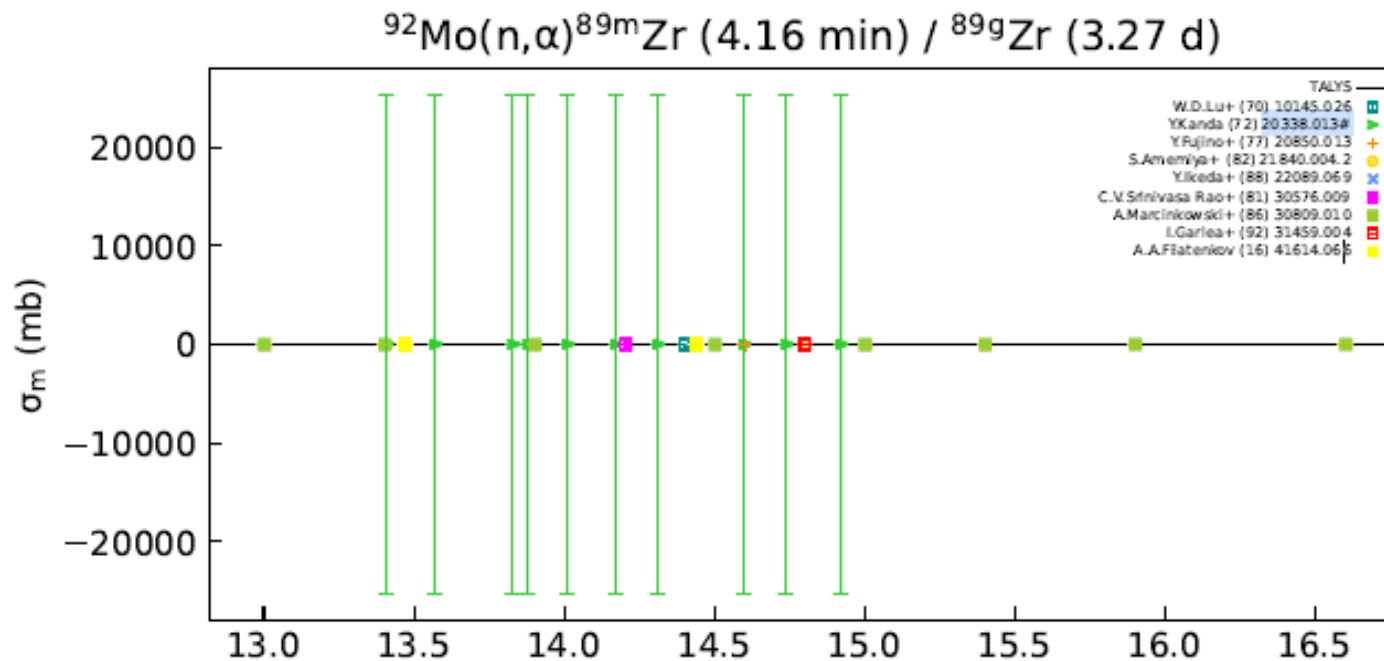
Repetition of ERR-T (Illegal)

SQL prints



**25.4% was wrongly printed as 25.4b for all data points!**

# Trouble due to Repetition of Same Heading (Cont)



The  $\sigma_m$  and  $\sigma_g$  plots with extremely long error bars plotted by Alberto Rodrigo due to coding of ERR-T twice (% in COMMON section and barn in DATA section).



# Trouble due to Repetition of Same Heading (Cont)

- Summary of illegal duplicating headings is generated for every EXFOR database update by Viktor.
- EXFOR Formats Manual say “No field heading (data heading plus, perhaps, a pointer) may be repeated except for the following cases.”

Duplicating HEADER

n	Entry	Subent	Header	TransID	TransDate	File	
1	1	10007	10007003	DATA-ERR	0000	20050926	exfor-2015-05-05.bck
2			10007008	DATA-ERR	0000	20050926	exfor-2015-05-05.bck
3			10007013	DATA-ERR	0000	20050926	exfor-2015-05-05.bck
4	2	10384	10384019	DATA-ERR	0000	20050926	exfor-2015-05-05.bck
5	3	10485	10485004	EN-ERR	1401	20141111	exfor-2015-05-05.bck
6	4	10785	10785002	ANG	1367	20110112	exfor-2015-05-05.bck
7	5	10827	10827017	DATA-ERR	1374	20111113	exfor-2015-05-05.bck
8	6	10875	10875008	SPIN J	1435	20180122	trans.1435
9	7	11319	11319002	EN-RSL	0000	20050926	exfor-2015-05-05.bck
10	8	11432	11432002	DATA-ERR	1396	20140403	exfor-2015-05-05.bck
11			11432003	DATA-ERR	1396	20140403	exfor-2015-05-05.bck
12	9	11635	11635009	STAT-W G	0000	20050926	exfor-2015-05-05.bck
13	10	12220	12220002	ANG-RSL	0000	20050926	exfor-2015-05-05.bck
14	11	12373	12373010	ANG-RSL	0000	20050926	exfor-2015-05-05.bck
15	12	12688	12688002	ANG	1401	20141111	exfor-2015-05-05.bck
16	13	12787	12787002	FLAG	1336	20050926	exfor-2015-05-05.bck
17	14	12989	12989003	POL-BM	0000	20050926	exfor-2015-05-05.bck
18	15	13126	13126002	E-LVL	0000	20050926	exfor-2015-05-05.bck
19	16	13913	13913002	ERR-T	1402	20150107	exfor-2015-05-05.bck
20			13913003	ERR-T	1402	20150107	exfor-2015-05-05.bck
21	17	13923	13923003	E-ERR	0000	20050926	exfor-2015-05-05.bck
22	18	14125	14125019	EN	1401	20141111	exfor-2015-05-05.bck
23			14125019	EN-ERR	1401	20141111	exfor-2015-05-05.bck

<https://nds.iaea.org/exfor-master/x4compil/Errors-main.htm>

# Trouble due to Repetition of Same Heading (Cont)

Repetition is legal for unresolved secondary energies. But ...-MIN and ...-MAX would be better both for database designs and physics.

```

SUBENT          20763004      840207
BIB              2              5
REACTION        (42-MO-94 (N, INL) 42-MO-94, PAR, DA)
...
COMMON          3              3
E-LVL          E-LVL          E-LVL
MEV             MEV             MEV
  2.5330E+00  2.5680E+00  2.6080E+00
ENDCOMMON
    
```

is actually measuring

```

COMMON          3              3
E-LVL-MIN      E-LVL-MAX
MEV             MEV
  2.5330E+00  2.6080E+00
ENDCOMMON
    
```

This problem is observed not only for E-LVL but also for E (e.g., unresolved discrete outgoing gamma lines).

D. Abriola+, J, NDS, 107, 2423, 2006

E(level) <sup>†</sup>	J <sup>π</sup> <sup>‡</sup>	T <sub>1/2</sub>	
2121 5			
2294.79 16	4 <sup>+</sup>	76 <sup>a</sup> fs 11	C
2322 2	(6 <sup>+</sup> ) <sup>#</sup>		
2393.02 6	2 <sup>+</sup>	83 <sup>a</sup> fs +12-10	A C
2423.45 <sup>@</sup> 9	6 <sup>+</sup>		BC
→ 2533.87 12	3 <sup>-</sup>	0.52 <sup>a</sup> ps +9-8	C
→ 2564.98 19	4 <sup>+</sup>	0.16 <sup>a</sup> ps +5-3	
→ 2580 5	(3 <sup>-</sup> ) <sup>#</sup>		
→ 2610.57 & 16	(5) <sup>-</sup>	0.44 <sup>a</sup> ps +11-8	C
2703 5	(3 <sup>-</sup> ) <sup>#</sup>		

We know contribution of 3<sup>-</sup> level is not excluded.



# Conclusions

- X4Pro is very powerful to produce a list of dataset or data points in EXFOR (especially for quantities not supported by C4.)
- One can see the database file contents easily by a browser such as “DB Browser for SQLite”
- Conversion of SQL output to another format (e.g., C4) by an additional script could make the SQL output more useful.
- Better to avoid repetition of the same heading even when it is legal.
- Suggestion to X4Pro: It would be helpful if the column name of the same content is unified (e.g., DatasetID=ReacodeID).





# Thank you!



My Special Thanks to Viktor Zerkin!

K. N  
3. May 2022  
Kaiserwass  
Alte Donau