

Database development and usage for prototype and pre-series testing

Ali Al Kadhim



Ali Al Kadhim - HGCal DB



Outline

- Motivation
- Personal involvement: spreadsheets to database (DB)
- CMS HGCAL DB overview
- DB Graphical User Interface (GUI) and XML schemas
- Connecting to the DB with SQLDeveloper tutorial
- Proposed changes to DB and HGCAL community involvement

Motivation

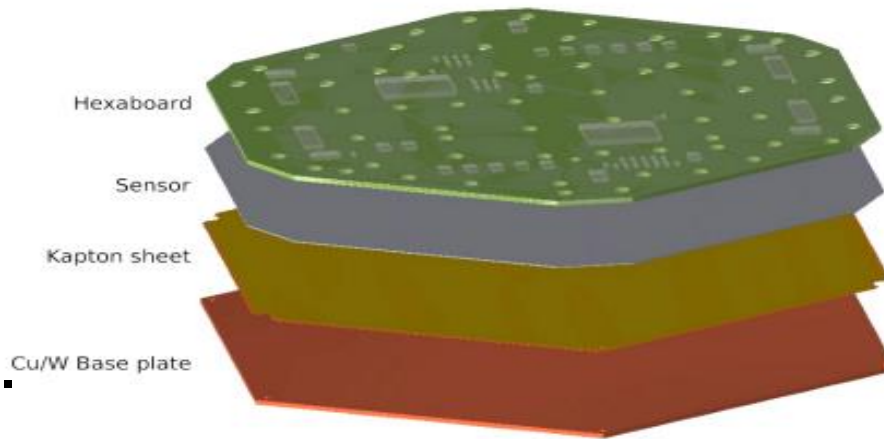
- Phase-2 upgrade of the CMS HGCal will encompass tens of thousands of new sensors.
- Extensive campaigns have been conducted by CMS to determine suitable material and layout parameters for operation in HL-LHC environment.
- Sensor Quality Control (SQC) institutions routinely run tests on these sensors to determine if they pass these requirements.
 - Successful HGCal construction will require collaboration between institutions/users on sensors and their results -> Sensor results must be stored in a central database (DB)!

Motivation

- HGICAL requires $\approx 26,000$ Si detector modules (5% are tested):
 - CE-E module (electromagnetic) or a CE-H module (hadronic).

- A unified HGICAL DB allows users from all institutions access to test results, status, location, etc.

- When assembling these modules, users must have easy access to sensor info/results from a unified database.



CE-E Si module

(PCB)

Si 8" sensor

For electrical insulation
from base plate

For placement onto
cassettes

Spreadsheets to Database

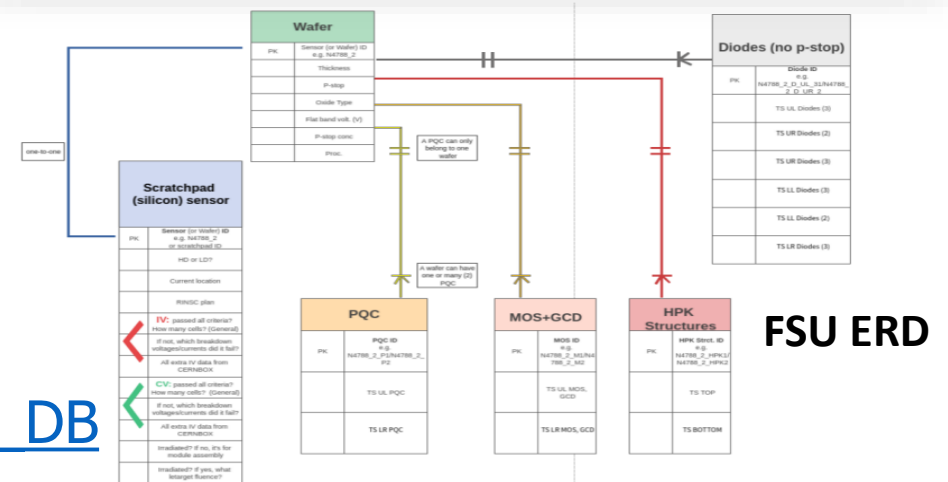
- Summary results of Si sensor analyses.
have been entered manually in spreadsheets
(which grew organically), such as:
 - This is very **error-prone** and **time-consuming!**
 - Not query-able.**
 - Doesn't capture **one-to-many** relationships.

Sensor ID	Scratch pad ID	Thickness	P-Stop	Oxide type	Flat band volt. (V)	P-Stop conc.	Proc.	HPK full probe card					
								I_tot_600V < 100uA	I_tot_800V < 2.5* I_tot_600V	1) Ncell with I600 > 100nA	2) Ncell with I800 > 2.5 * I600 & I600 > 10nA threshold I800 > 25nA & I600 < 10nA	3) More than 8 bad cells: requirem. 1) and 2)	4) More than two neighbour cells bad: requirem. 1) and 2)
N4791_1	100088	300	com	A	.5	STD	FZ	Passed	Failed	1	0	Passed	Passed
N4791_2	100089	300	com	A	.5	STD	FZ	Passed	Passed	0	0	Passed	Passed
N4791_3	100090	300	com	A	.5	STD	FZ	Passed	Passed	0	0	Passed	Passed

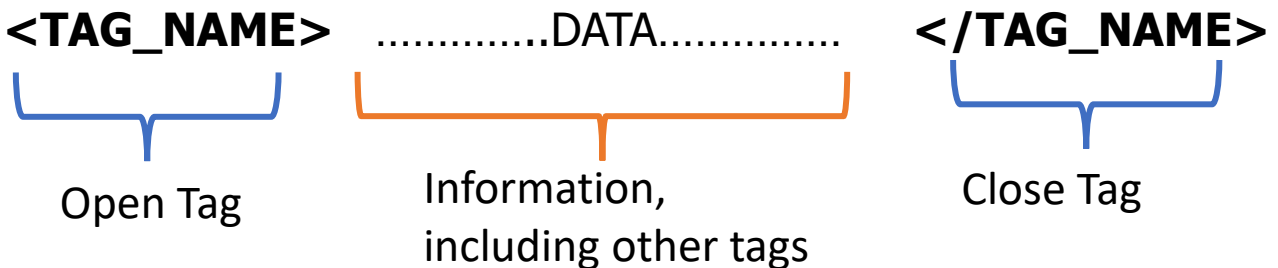
~~Spreadsheets~~ to Database

- Summary results of Si sensor analyses have been entered manually in spreadsheets (which grew organically), such as:
 - This is very **error-prone** and **time-consuming**!
 - Not query-able.**
 - Doesn't capture **one-to-many** relationships.
- We (FSU) constructed a relational (SQL) database that stores and updates sensor attributes, summary results, history, location, etc.
- Code: https://github.com/AliAlkadhim/FSU_HGCAL_DB
- More on FSU HGCAL DB here: https://indico.cern.ch/event/1164937/contributions/4891850/attachments/2451443/4200827/HGCAL_DB_May25_Alkadhim.pdf

Sensor ID	Scratch pad ID	Thickness	P. Stop	Oxide type	Flat band volt. (V)	P. Stop conc.	Proc.	HPK full probe card					
								I_tot_600V <100uA	I_tot_800V <2.5* I_tot_600V	1) Ncell with 1600 > 100nA	2) Ncell with 1800 > 2.5 * 1600 & 1600>10nA threshold 1800>25nA & 1600<10nA	3) More than 8 bad cells: requirem. 1) and 2)	4) More than two neighbour cells bad: requirem. 1) and 2)
N4791_1	100088	300	com	A	5	STD	FZ	Passed	Failed	1	0	Passed	Passed
N4791_2	100089	300	com	A	5	STD	FZ	Passed	Passed	0	0	Passed	Passed
N4791_3	100090	300	com	A	5	STD	FZ	Passed	Passed	0	0	Passed	Passed



CMS HGCAL DB

- Global DB for all HGCAL parts, tests, modules, etc. from all testing, production, etc. institutions.
- There are 2 instances of HGCAL Database (DB)
 - HGCAL Development DB – **INT2R** Database
 - HGCAL Production DB – **CMSR** Database
- To learn more about the different schemas in the HGCAL DB, see <https://indico.cern.ch/event/1164937/contributions/4891851/attachments/2451351/4200663/XML%20Templates%20HGCAL%20Si%20Sensors.pdf>
- The DB (I will discuss INT2R, but the two are nearly identical), is based on XML schemas.
- XM has tag-based syntax (like HTML), all it is: 

HGCAL DB

HGCAL DB consists of the following inter-connected schemas (accounts), each used to store different data types:

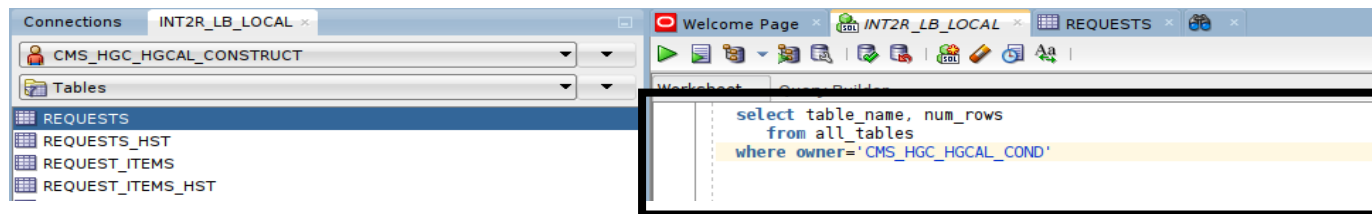
Tables in these schemas are related to each other!

- **CMS_HGC_CORE_ATTRIBUTE**
to store attribute information related to parts and data types
- **CMS_HGC_CORE_COND**
to store meta data of user-generated data (**conditions data**).
- **CMS_HGC_HGCAL_COND**
To store **detailed tables** for specific user-generated test data.
- **CMS_HGC_CORE_CONSTRUCT**
to store detectors **parts**.
- **CMS_HGC_CORE_IOV_MGMNT**
to store data IOVs (**interval of validity**) of user generated data.
- **CMS_HGC_CORE_MANAGEMNT**
to store “management” data type, e.g. locations, institutions, etc.

- Every part has a *kind of Part* and a *serial Number*, and is uniquely identified in the DB by this pair of parameters.

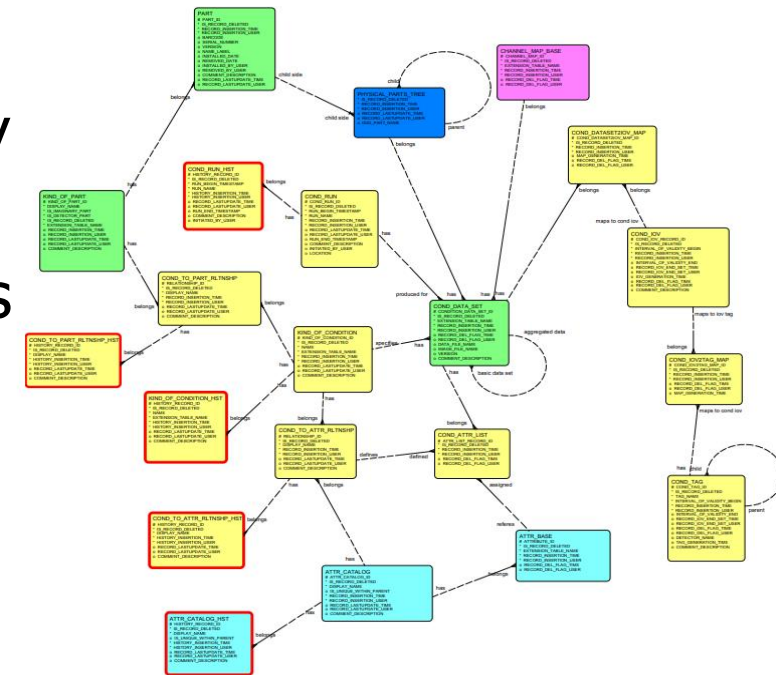
Accessing the Database with SQLDeveloper

- **SQLDeveloper** is currently the preferred way to access and query this XML database.
- For more instructions on how to access, view and query the database, see the backup slides!
- Once connected, you can see the CMS_HGC_* schemas previously mentioned.
- You can also do any SQL queries (also in backup).



Sample SQL query

Sample ERD



Connecting to the DB with SQLDeveloper Tutorial (linux)


- Now put the following text in a file named **tnsnames.ora** and put it in your **/etc/** directory.

- Now `ssh` to `lxplus` with port-forwarding:

```
ssh -XY -L 10131:itrac1609-  
v.cern.ch:10121 -L 10132:itrac1601-  
v.cern.ch:10121  
username@lxplus.cern.ch
```

Where [username](#) is your `lxplus` username.

tnsnames.ora

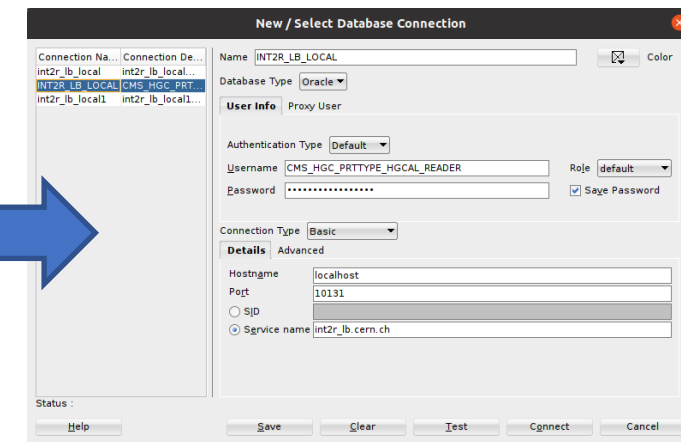
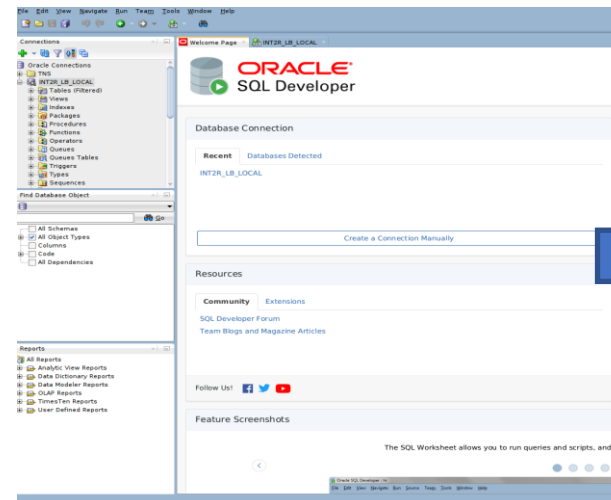


```
int2r_lb_local=(  
  DESCRIPTION=  
    (ADDRESS= (PROTOCOL=TCP) (HOST=localhost) (PORT=10131))  
    (ADDRESS= (PROTOCOL=TCP) (HOST=localhost) (PORT=10132))  
    (LOAD_BALANCE=yes)  
    (CONNECT_DATA=  
      (SERVER=DEDICATED)  
      (SERVICE_NAME=int2r_lb.cern.ch)  
      (FAILOVER_MODE=  
        (TYPE=SELECT)  
        (METHOD=BASIC)  
        (RETRIES=180)  
        (DELAY=5)  
      )  
    )  
  )  
)
```

You must `ssh` every time you want to connect to the DB with SQLDeveloper!

Connecting to the DB with SQLDeveloper Tutorial

- Launch SQLDeveloper (on unix by typing **sqldeveloper**).
- Press "Open a Connection Manually" in the front page.
- On the "New / Select Database Connection" Dialogue box, fill:
 - Name: **INT2R_LB_LOCAL**
 - Username: **CMS_HGC_PRTTYPE_HGCAL_READER**
 - Password: **HGCAL_Reader_2016**
 - Check "save password" if you want.
 - Hostname: **localhost**
 - Port: **10131**
 - Check on "Service name"
 - Service name: **int2r_lb.cern.ch**
 - Press "**Connect**"
 - You're connected!



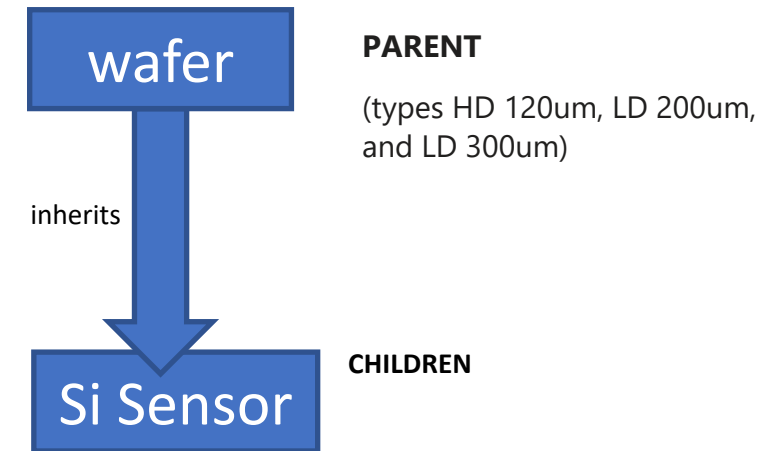
GUIs! OMS and DBA

- Production Detector Construction Application (**DCA**: https://cmsdca.cern.ch/hgc_cmsr/construct/parts/) GUI

High level Information on detector assembly parts: mainly for current status and tracking (location)

Parent							
ID	KindOfPart	Location	Manufacturer	Serial number	Name	Latest status	Record insertion time
2267	HGC Six Inch Silicon Module	SiDet	None	UCSB_HGC_TST_MOD_8		None	Dec. 11, 2017, 8:45 p.m.

Children							
ID	KindOfPart	Location	Manufacturer	Serial number	Name	Latest status	Record insertion time
2780	HPK Six Inch 128 Sensor Cell	Fermilab	Hamamatsu-HPK	HPK_6in_128_0003-1		None	March 25, 2018, 10:20 p.m.
2781	HPK Six Inch 128 Sensor Cell	Fermilab	Hamamatsu-HPK	HPK_6in_128_0003-2		None	March 25, 2018, 10:20 p.m.
2782	HPK Six Inch 128 Sensor Cell	Fermilab	Hamamatsu-HPK	HPK_6in_128_0003-3		None	March 25, 2018, 10:20 p.m.
2783	HPK Six Inch 128 Sensor Cell	Fermilab	Hamamatsu-HPK	HPK_6in_128_0003-4		None	March 25, 2018, 10:20 p.m.
2784	HPK Six Inch 128 Sensor Cell	Fermilab	Hamamatsu-HPK	HPK_6in_128_0003-5		None	March 25, 2018, 10:20 p.m.



- To view OMS: go [here](#)
 - We have plans to make a OMS a complete HGICAL GUI.

The DB is XML schema

XML Template for HGC_CERN_SENSOR_IV table

```
<ROOT>
<HEADER>
  <TYPE>
    <!-- Comment: Table Name -->
    <EXTENSION_TABLE_NAME>HGC_CERN_SENSOR_IV</EXTENSION_TABLE_NAME>
    <!-- Kind of condition name -->
    <NAME> HGC CERN Sensor IV</NAME>
  </TYPE>
  <RUN>
    <RUN_NAME>Your Run Name</RUN_NAME>
    <!-- Enter your timestamp -->
    <RUN_BEGIN_TIMESTAMP>2018-05-14 00:00:00</RUN_BEGIN_TIMESTAMP>
    <RUN_END_TIMESTAMP>2018-05-14 00:00:00</RUN_END_TIMESTAMP>
    <INITIATED_BY_USER>Your Name</INITIATED_BY_USER>
    <LOCATION>CERN</LOCATION>
    <COMMENT_DESCRIPTION>Your Comments</COMMENT_DESCRIPTION>
  </RUN>
</HEADER>
<DATA_SET>
  <PART>
    <!-- Kind of Part name -->
    <KIND_OF_PART>HGC Sensor</KIND_OF_PART>
    <!-- Sensor serial number -->
    <SERIAL_NUMBER>XXXXXXXXXXXXXXXXXXXX</SERIAL_NUMBER>
  </PART>
  <DATA>
    <!-- table columns with data -->
    <VOLTS>-25</VOLTS>
    <CURNT_NANOAMP>-7.609905</CURNT_NANOAMP>
    <ERR_CURNT_NANOAMP>0.01653122</ERR_CURNT_NANOAMP>
    <TOT_CURNT_NANOAMP>-2000</TOT_CURNT_NANOAMP>
    <ACTUAL_VOLTS>-25</ACTUAL_VOLTS>
    <TIME_SECS>10</TIME_SECS>
    <TEMP_DEGC>23</TEMP_DEGC>
    <HUMIDITY_PRCNT>46.6</HUMIDITY_PRCNT>
    <CELL_NR>YYYY</CELL_NR>
  </DATA>
  .
  .
  .
</DATA_SET>
</ROOT>
```



viewed in SQLDeveloper

RECORD_ID	CONDITION_DATA_SET_ID	VOLTS	CURNT_NANOAMP	ERR_CURNT_NANOAMP	TOT_CURNT_NANOAMP	ACTUAL_VOLTS	TIME_SECS	TEMP_DEGC	HUMIDITY_PRCNT	CELL_NR
1	44950	6215	-900	-25.18362	0.02615799	-5890	-900.05	5203.35	23.3	45.7 (null)
2	44951	6215	-1000	-27.55292	0.001586537	-6540	-1000.02	5547.73	23.3	45.5 (null)
3	44952	6216	-25	-8.354687	0.005196689	-1770	-25	43.42	23	46.6 (null)
4	44953	6216	-50	-10.0235	0.052135	-2100	-49.98	382.88	23	46.6 (null)
5	44954	6216	-75	-10.41493	0.05343381	-2310	-74.98	720.04	23	46.5 (null)
6	44955	6216	-100	-11.07294	0.05636589	-2470	-100	1058.32	23	46.4 (null)
7	44956	6216	-125	-11.57239	0.06113021	-2590	-125.01	1396.93	23	46.4 (null)
8	44957	6216	-150	-11.98722	0.003079648	-2670	-150.03	1736.18	23	46.4 (null)
9	44958	6216	-175	-12.28808	0.002297368	-2740	-175.02	2077.31	23	46.3 (null)
10	44959	6216	-200	-12.53266	0.0003752777	-2800	-200	2417.74	23.1	46.2 (null)
11	44960	6216	-250	-13.03023	0.01779207	-2900	-250.01	2760.22	23.1	46.1 (null)
12	44961	6216	-300	-13.45169	0.001566727	-3010	-300.04	3102.42	23.1	46 (null)
13	44962	6216	-400	-14.68026	0.003727604	-3310	-400.02	3446.69	23.1	45.9 (null)
14	44963	6216	-500	-16.27897	0.004527924	-3690	-500.03	3790.62	23.2	45.9 (null)
15	44964	6216	-600	-18.2191	0.0007176582	-4160	-600.07	4143.67	23.2	45.9 (null)
16	44965	6216	-700	-20.36152	0.02306462	-4700	-700.06	4501.7	23.2	45.8 (null)
17	44966	6216	-800	-22.65578	0.006589089	-5280	-800.06	4863.22	23.2	45.8 (null)
18	44967	6216	-900	-24.98369	0.002124908	-5890	-900.05	5204.63	23.3	45.7 (null)
19	44968	6216	-1000	-27.3154	0.01333515	-6540	-1000.02	5549.02	23.3	45.5 (null)
20	44969	6217	-25	-7.495169	0.006570783	-1770	-25	44.7	23	46.6 (null)
21	44970	6217	-50	-8.798325	0.006254623	-2100	-49.98	384.58	23	46.5 (null)
22	44971	6217	-75	-9.66792	0.01745253	-2310	-74.98	721.73	23	46.5 (null)
23	44972	6217	-100	-10.40346	0.003713897	-2470	-100	1060.02	23	46.4 (null)
24	44973	6217	-125	-11.67291	0.00768125	-2590	-125.01	1398.62	23	46.4 (null)
25	44974	6217	-150	-12.3211	0.006001911	-2670	-150.03	1737.46	23	46.4 (null)
26	44975	6217	-175	-12.5811	0.001207145	-2740	-175.02	2078.59	23	46.3 (null)
27	44976	6217	-200	-13.65851	0.01223054	-2800	-200	2419.02	23.1	46.2 (null)
28	44977	6217	-250	-13.9874	0.0009528379	-2900	-250.01	2761.5	23.1	46.1 (null)
29	44978	6217	-300	-14.49301	0.005028681	-3010	-300.04	3103.71	23.1	46 (null)
30	44979	6217	-400	-15.67001	0.01586948	-3310	-400.02	3447.96	23.1	45.9 (null)
31	44980	6217	-500	-17.34125	0.002627553	-3690	-500.03	3791.9	23.2	45.9 (null)
32	44981	6217	-600	-19.48549	0.004614965	-4160	-600.07	4144.96	23.2	45.9 (null)
33	44982	6217	-700	-21.71133	0.01221068	-4700	-700.06	4502.99	23.2	45.8 (null)
34	44983	6217	-800	-23.07205	0.01468643	-5280	-800.06	4864.51	23.2	45.8 (null)
35	44984	6217	-900	-24.64296	0.01375111	-5890	-900.05	5205.02	23.3	45.7 (null)

Converting .txt results to .xml DB files (using the XML layout)

- **HGCAL_sensor_db_interface:**

- https://gitlab.cern.ch/CLICdp/HGCAL/hgcal_sensor_db_interface

- Example Usage

- Convert HPK data to .txt:

- `python db_convert_hpk_to_txt.py -i '/home/data/cms/results/hamamatsu/HPK_8in_432ch_120um/200805_S15591-03(ES1) inspection sheet.xlsx'`

- Convert .txt data to .xml:

- `python db_register_data_iv.py -i '/home/data/cms/results/hamamatsu/8-432-120E-00-N1308-0000178-432-120E-00-N1308-00001 IV.txt' --serial '8-432-120E-00-N1308-00001-SE-1' --run 'IV Test 01' --location 'CERN' --operator 'Florian Pitters' --comment 'Nothing special.'`

- Similar operations can also be found here

- https://github.com/AliAlkadhimi/FSU_HGCAL_DB

Uploading to the DB (CMSR or INT2R)

- Once the XML files for the data that you want to store in the DB have been generated (based on the templates mentioned previously [here](#) and [here](#)), and you've been given upload permissions by Umesh (joshi@fnal.gov), you can upload to the DB, which uses DBLoader
 - DBloader code: <https://github.com/valdasraps/cmsdbldr>
- More instructions to upload are in backup slides.
- To upload to the INT2R (development) database, simply do
 - `scp <file.xml> USERNAME@dbloader-hgcal.cern.ch:/home/dbspool/spool/hgc/int2r`
- To upload to the CMSR (production) database, do
 - `scp <file.xml> USERNAME@dbloader-hgcal.cern.ch:/home/dbspool/spool/hgc/int2r`
- Where [USERNAME](#) is your lxplus username

We Need More Involvement & Input from HGICAL!

- Everyone from HGICAL (testing, module assembly, etc.) is encouraged to upload to the DB!
- If you need **permissions to upload to HGICAL DB**, please contact Imran (Muhammad.Imran@cern.ch), Rao (muhammad.atif.shad.rao@cern.ch), or **Umesh** (joshi@fnal.gov).
- Attend an HGICAL DB meeting <https://indico.cern.ch/category/11190/> .
- Do we want to add extra information (tables or attributes) to the DB? We are working on changes, but we need your valued opinions!
 - E.g. `HGC_CERN_SENSOR_IV_SUMRY` has summary of grading criteria for sensors. Do we want to add more information to this summary?
 - Should the sensor tables include extra information, e.g. current location?
 - Perhaps sensors should also have a tag, whether they have been tested or not (only 5% of sensors undergo testing).

Backup

More Useful Links

- **HGCALDB Documentation:** <https://hgcaldb.docs.cern.ch/>
- **Jira** (for creating issues, tickets, etc.)
<https://its.cern.ch/jira/projects/CMSOTOMSDB>
- **RESThub API:** <https://github.com/valdasraps/resthub/> (the database is [RESTful!](#))
- **XML Tables Up-to-date Templates Repository:**
<https://cernbox.cern.ch/index.php/s/nyoizOOunNo2pgLpassword:HGcdatabase>
 - Since the XML templates are continuously being changed with addition/removal of attributes, we are collecting the most recent templates

SQL Developer: installation


- Follow the instructions on https://docs.oracle.com/cd/E39885_01/doc.40/e38928/install.htm#RPTIG122 for installation instructions on any system.
- Important: SQL Developer requires JDK 7 or above. Download from <https://www.oracle.com/java/technologies/downloads/#java8> and make sure your default Java in your system points to this **Oracle JKD**.
 - This means setting the **JAVA_HOME** and **JKD_HOME** environment variables are pointing to the Oracle Java.
 - Doing `java -version` should result in "java version ..." not "openjdk version ..."
- SQLDeveloper could now be launched.
- On Unix, launch with typing `sqldeveloper` in a terminal to see that it works.

Connecting to the Database with SQLDeveloper (1)

[tnsnames.ora](#)

- Now put the following text in a file named **tnsnames.ora** and put it in your **/etc/** directory.
- Now ssh to lxplus

```
ssh -XY -L 10131:itrac1609-  
v.cern.ch:10121 -L 10132:itrac1601-  
v.cern.ch:10121  
username@lxplus.cern.ch
```



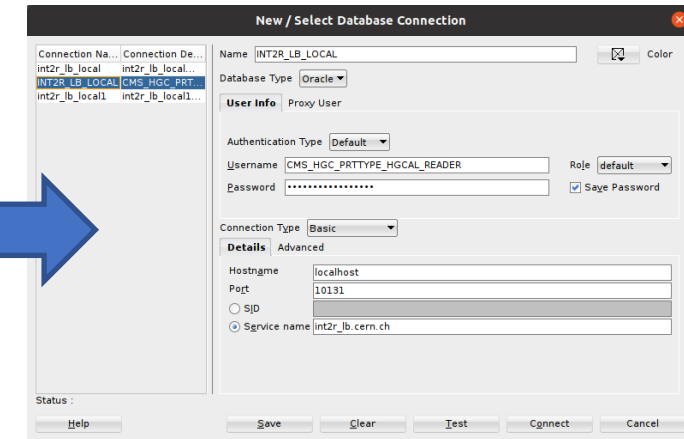
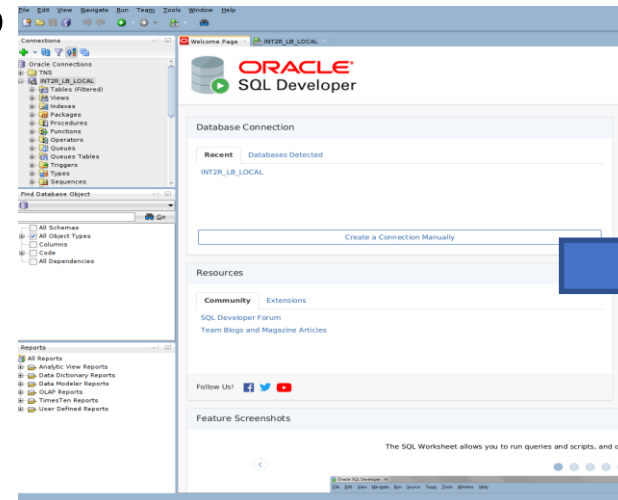
```
int2r_lb_local=(  
  DESCRIPTION=  
    (ADDRESS= (PROTOCOL=TCP) (HOST=localhost) (PORT=10131))  
    (ADDRESS= (PROTOCOL=TCP) (HOST=localhost) (PORT=10132))  
    (LOAD_BALANCE=yes)  
    (CONNECT_DATA=  
      (SERVER=DEDICATED)  
      (SERVICE_NAME=int2r_lb.cern.ch)  
      (FAILOVER_MODE=  
        (TYPE=SELECT)  
        (METHOD=BASIC)  
        (RETRIES=180)  
        (DELAY=5)  
      )  
    )  
  )  
)
```

Where username is your lxplus username.

- You'd need to ssh with the command above every time you want to connect to the DB!

Connecting to the Database with SQLDeveloper (2)

- Launch SQLDeveloper (on unix by typing `sqldeveloper` in a terminal).
- Press "Open a Connection Manually" in the front page.
- On the "New / Select Database Connection" Dialogue box, fill:
 - Name: **INT2R_LB_LOCAL**
 - Username: **CMS_HGC_PRTTYPE_HGCAL_READER**
 - Password: **HGCAL_Reader_2016**
 - Check "save password" if you want.
 - Hostname: **localhost**
 - Port: **10131**
 - Check on "**Service name**"
 - Service name: **int2r_lb.cern.ch**
 - Press "**Connect**"



Connecting to the Database with SQLDeveloper (3)

- Click **View** at the top of the screen.
- Click on "**Find DB Object**"
- Click on this database icon and select **INT2R_LB_LOCAL**
- Click on the **+** sign on the left of "**All Schemas**"
- Scroll down to entries that start with "**CMS_HGC_***" and check one (or more) that you want to view.
 - For us the relevant tables are actually ones like "HGC_CERN_SENSOR_IV"
- Then click on "**Go**" (which has binoculars next to it).

Uploading to the HGICAL DB (1/3)

- Generate XML files for your data based on the templates
 - Templates from Umesh:
<https://indico.cern.ch/event/1164937/contributions/4891851/attachments/2451351/4200663/XML%20Templates%20HGICAL%20Si%20Sensors.pdf>
 - Find the relevant templates from the documentation:
<https://twiki.cern.ch/twiki/bin/view/CMS/ConstructionDB>
 - We are also collecting up-to-date templates [here](#)
 - Scripts that can help in conversion to xml can be found [here](#) or [here](#) (or write your own scripts).
- You can upload to the DB using DBLoader (see next slides).
 - DBloader code: <https://github.com/valdasraps/cmsdbldr>

Uploading to the HGICAL DB (2/3)

- Before anything, make sure you have permission to upload from Umesh (joshi@fnal.gov) or other admins in the group.
- Before you can upload, you need to setup an ssh tunnel with

```
ssh -XY -L 10131:itrac1609-v.cern.ch:10121 -L 10132:itrac1601-v.cern.ch:10121 username@lxplus.cern.ch
```

Where [username](#) is your lxplus username.

- If that doesn't work, kill processes using this port with `lsof -ti:10131 | xargs.`
- If it still doesn't work, see later slides (or contact us for help).

Uploading to the HGCal DB (3/3)

- To upload to the INT2R (development) database, do
 - `scp <file.xml> xml username@dbloader-hgcal.cern.ch:/home/dbspool/spool/hgc/int2r`
- To upload to the CMSR (production) database, do
 - `scp <file.xml> xml username@dbloader-hgcal.cern.ch:/home/dbspool/spool/hgcal/int2r`
- Check the status of the job on `/home/dbspool/state/hgc/int2r/filename.xml`
 - E.g. username@dbloader-hgcal.cern.ch:/home/dbspool/state/hgc/int2r/filename.xml
 - **State 0 means success (uploaded), not 0 means error (not uploaded)**
- Check the log file `/home/dbspool1/logs/hgc/int2r/filename.xml` for log information.

Uploading to HGICALDB: Optional

- If the previous instructions didn't work for you (and you have upload permissions), try the following:
 - Copy the XML file(s) you want to upload to **lxplus**, say to a directory DATABASE
 - `scp file.xml`
[username@lxplus.cern.ch:/afs/cern.ch/user/u/username/DATABASE](#)
 - `ssh` to **lxplus** and go to that directory
 - `ssh username@lxplus.cern.ch && cd DATABASE`
 - `Ssh` to your **dbloader** account
 - `ssh username@dbloader-hgcal`
 - Copy file to spool area
 - `cp file.xml /home/dbspool/spool/hgc/int2r/`
 - Check the log for this upload
 - `cat /home/dbspool/logs/hgc/int2r/file.xml`

Queries in SQLDeveloper Example 1

- On "worksheet" tab, write any SQL query
- Press ctrl+enter to execute.

(you don't have to check any tables, since you have access to all of INT2R_LB_LOCAL.)

The screenshot displays the SQL Developer interface with the following components:

- Connections:** A list of database connections, with 'CMS_HGC_HGCAL_CONSTRUCT' selected.
- Tables:** A tree view showing database tables, including REQUESTS, REQUESTS_HST, REQUEST_ITEMS, REQUEST_ITEMS_HST, SHIPMENTS, SHIPMENTS_HST, SHIPMENT_ITEMS, and SHIPMENT_ITEMS_HST.
- Find Database Object:** A search bar with 'INT2R_LB_LOCAL' entered.
- Query Builder:** A tab where the SQL query is written:

```
select table_name, num_rows
from all_tables
where owner='CMS_HGC_HGCAL_COND'
```
- Query Result:** A table showing the results of the query, with 33 rows fetched in 0.942 seconds. The table has two columns: TABLE_NAME and NUM_ROWS.

TABLE_NAME	NUM_ROWS
1 FLATNS_PCB_ROCS_DATA	38
2 HGCROC_TEST	3
3 HGC_BARE_PCB_DATA	38
4 HGC_BARE_PCB_ELEC_DATA	38
5 HGC_CERN_SENSOR_CV	7604
6 HGC_CERN_SENSOR_CV_SUMRY	1
7 HGC_CERN_SENSOR_IV	8003
8 HGC_CERN_SENSOR_IV_SUMRY	1
9 HGC_MOD_ASMBLY	6
10 HGC_MOD_ASMBLY_COND	(null)
11 HGC_MOD_WIREBOND_TEST	(null)
12 HGC_NOISE_MAP_BLOB	2
13 HGC_PQC_DIODE_CV	(null)
14 HGC_PQC_DIODE_IV	(null)
15 HGC_PQC_FET	(null)
16 HGC_PQC_GCD	(null)
17 HGC_PQC_LINWIDTH	(null)
18 HGC_PQC_MOS	(null)
19 HGC_PQC_OXIDE_BREAKDOWN	(null)
20 HGC_PQC_SUMRY	(null)
21 HGC_PQC_VAN_DER_PAUL	(null)
22 HGC_PRT0_MOD_ASMBLY	6
23 HGC_PRT0_MOD_ASMBLY_COND	(null)
24 HGC_SENSOR_CV	618
25 HGC_SENSOR_DEFECT_CHKS	(null)

Queries in SQLDeveloper Example 2

- See everything in the "HGC_CERN_SENSOR_IV" table.

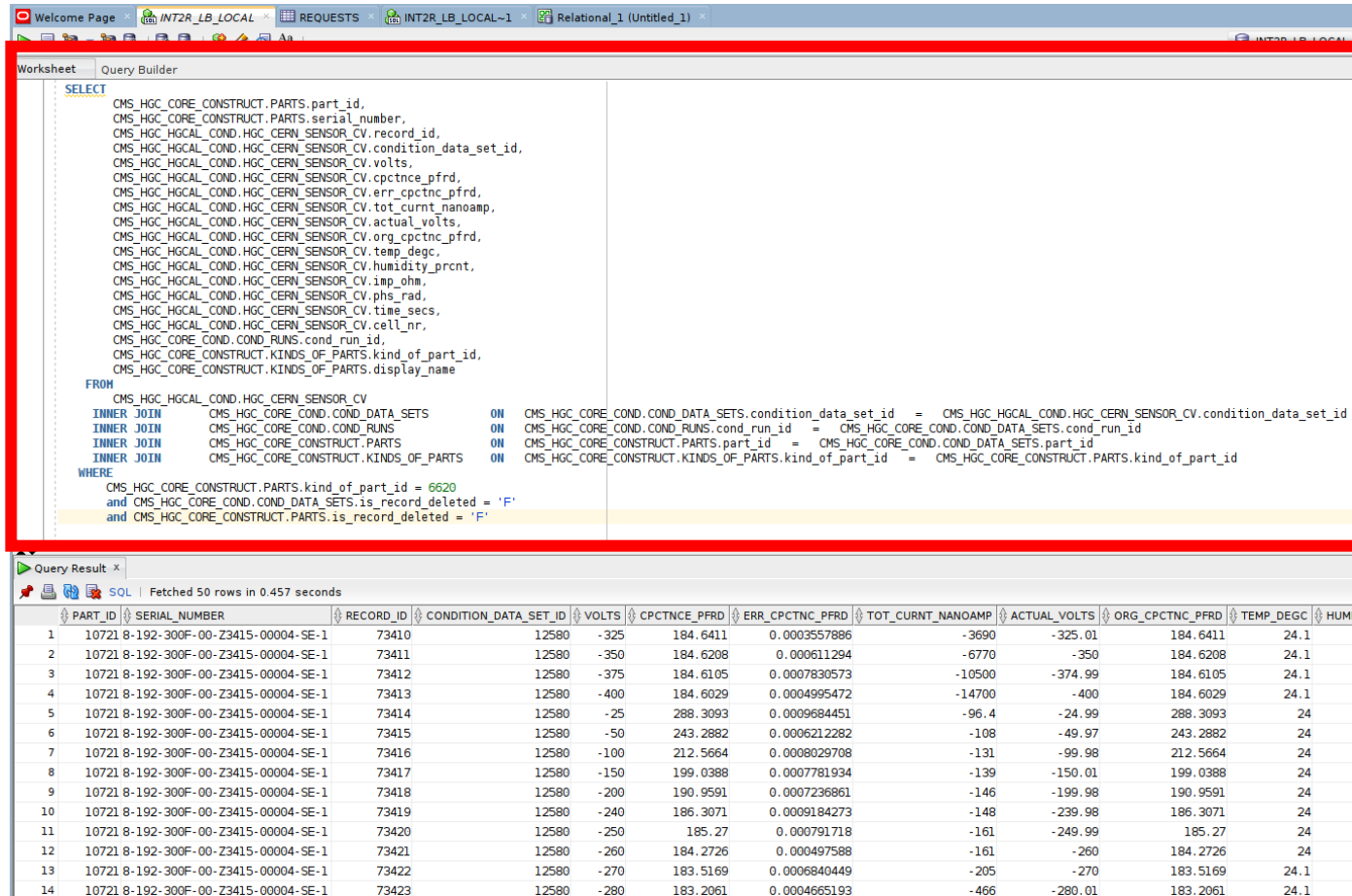
The screenshot displays the Oracle SQL Developer interface for the 'INT2R_LB_LOCAL' database. The 'Connections' pane on the left shows the database structure, with the 'HGC_CERN_SENSOR_IV' table highlighted under the 'HGC_CERN_SENSOR' schema. The 'Query Builder' pane on the right shows the following SQL query:

```
select *
from CMS_HGC_HGCAL_COND.hgc_cern_sensor_iv
```

The 'Query Result' pane at the bottom shows the results of the query, which are 50 rows. The columns are: RECORD_ID, CONDITION_DATA_SET_ID, VOLTS, CURRNT_NANOAMP, ERR_CURRNT_NANOAMP, TOT_CURRNT_NANOAMP, ACTUAL_VOLTS, TIME_SECS, TEMP_DEGC, HUMIDITY_PRCNT, and CELL_NR. The data is as follows:

RECORD_ID	CONDITION_DATA_SET_ID	VOLTS	CURRNT_NANOAMP	ERR_CURRNT_NANOAMP	TOT_CURRNT_NANOAMP	ACTUAL_VOLTS	TIME_SECS	TEMP_DEGC	HUMIDITY_PRCNT	CELL_NR
1	44950	6215	-900	-25.18362	0.02615799	-5890	-900.05	5203.35	23.3	45.7 (null)
2	44951	6215	-1000	-27.55292	0.001586537	-6540	-1000.02	5547.73	23.3	45.5 (null)
3	44952	6216	-25	-8.354687	0.005196689	-1770	-25	43.42	23	46.6 (null)
4	44953	6216	-50	-10.0235	0.052135	-2100	-49.98	382.88	23	46.6 (null)
5	44954	6216	-75	-10.41493	0.05343381	-2310	-74.98	720.04	23	46.5 (null)
6	44955	6216	-100	-11.07294	0.05636589	-2470	-100	1058.32	23	46.4 (null)
7	44956	6216	-125	-11.57239	0.06113021	-2590	-125.01	1396.62	23	46.4 (null)
8	44957	6216	-150	-11.98722	0.003079648	-2670	-150.03	1736.18	23	46.4 (null)
9	44958	6216	-175	-12.28808	0.002297368	-2740	-175.02	2077.31	23	46.3 (null)
10	44959	6216	-200	-12.53266	0.0003752777	-2800	-200	2417.74	23.1	46.2 (null)
11	44960	6216	-250	-13.03023	0.01779207	-2900	-250.01	2760.22	23.1	46.1 (null)
12	44961	6216	-300	-13.45169	0.001566727	-3010	-300.04	3102.42	23.1	46 (null)
13	44962	6216	-400	-14.68026	0.003727604	-3310	-400.02	3446.69	23.1	45.9 (null)
14	44963	6216	-500	-16.27897	0.004527924	-3690	-500.03	3790.62	23.2	45.9 (null)
15	44964	6216	-600	-18.2191	0.0007176582	-4160	-600.07	4143.67	23.2	45.9 (null)
16	44965	6216	-700	-20.36152	0.02306462	-4700	-700.06	4501.7	23.2	45.8 (null)
17	44966	6216	-800	-22.65578	0.006589089	-5280	-800.06	4863.22	23.2	45.8 (null)
18	44967	6216	-900	-24.98369	0.002124908	-5890	-900.05	5204.63	23.3	45.7 (null)
19	44968	6216	-1000	-27.3154	0.0133515	-6540	-1000.02	5549.02	23.3	45.5 (null)
20	44969	6217	-25	-7.495169	0.006570783	-1770	-25	44.7	23	46.6 (null)
21	44970	6217	-50	-8.798325	0.006254623	-2100	-49.98	384.58	23	46.5 (null)
22	44971	6217	-75	-9.66792	0.01745253	-2310	-74.98	721.73	23	46.5 (null)
23	44972	6217	-100	-10.40346	0.003713897	-2470	-100	1060.02	23	46.4 (null)
24	44973	6217	-125	-11.67291	0.00768125	-2590	-125.01	1398.62	23	46.4 (null)
25	44974	6217	-150	-12.3211	0.006001911	-2670	-150.03	1737.46	23	46.4 (null)

You can make SQL Queries infinitely complex!



The screenshot displays a database query builder interface. The top section shows a 'Query Builder' tab with a 'SELECT' query. The query is highly complex, involving multiple joins and filters. The bottom section shows the 'Query Result' tab, which displays a table with 14 rows and 12 columns. The table contains data for various sensors and their readings.

```
SELECT
  CMS_HGC_CORE_CONSTRUCT.PARTS.part_id,
  CMS_HGC_CORE_CONSTRUCT.PARTS.serial_number,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.record_id,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.condition_data_set_id,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.volts,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.cpctnc_pfrd,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.err_cpctnc_pfrd,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.tot_curnt_nanoamp,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.actual_volts,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.org_cpctnc_pfrd,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.temp_deg,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.humidity_prcnt,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.imp_ohm,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.phs_rad,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.time_secs,
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.cell_nr,
  CMS_HGC_CORE_COND.COND_RUNS.cond_run_id,
  CMS_HGC_CORE_CONSTRUCT.KINDS_OF_PARTS.kind_of_part_id,
  CMS_HGC_CORE_CONSTRUCT.KINDS_OF_PARTS.display_name
FROM
  CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV
INNER JOIN CMS_HGC_CORE_COND.COND_DATA_SETS ON CMS_HGC_CORE_COND.COND_DATA_SETS.condition_data_set_id = CMS_HGC_HGCAL_COND.HGC_CERN_SENSOR_CV.condition_data_set_id
INNER JOIN CMS_HGC_CORE_COND.COND_RUNS ON CMS_HGC_CORE_COND.COND_RUNS.cond_run_id = CMS_HGC_CORE_COND.COND_DATA_SETS.cond_run_id
INNER JOIN CMS_HGC_CORE_CONSTRUCT.PARTS ON CMS_HGC_CORE_CONSTRUCT.PARTS.part_id = CMS_HGC_CORE_COND.COND_DATA_SETS.part_id
INNER JOIN CMS_HGC_CORE_CONSTRUCT.KINDS_OF_PARTS ON CMS_HGC_CORE_CONSTRUCT.KINDS_OF_PARTS.kind_of_part_id = CMS_HGC_CORE_CONSTRUCT.PARTS.kind_of_part_id
WHERE
  CMS_HGC_CORE_CONSTRUCT.PARTS.kind_of_part_id = 6620
  and CMS_HGC_CORE_COND.COND_DATA_SETS.is_record_deleted = 'F'
  and CMS_HGC_CORE_CONSTRUCT.PARTS.is_record_deleted = 'F'
```

PART_ID	SERIAL_NUMBER	RECORD_ID	CONDITION_DATA_SET_ID	VOLTS	CPCTNC_PFRD	ERR_CPCTNC_PFRD	TOT_CURNT_NANOAMP	ACTUAL_VOLTS	ORG_CPCTNC_PFRD	TEMP_DEGC	HUMID
1	10721 8-192-300F-00-Z3415-00004-SE-1	73410	12580	-325	184.6411	0.0003557886	-3690	-325.01	184.6411	24.1	
2	10721 8-192-300F-00-Z3415-00004-SE-1	73411	12580	-350	184.6208	0.000611294	-6770	-350	184.6208	24.1	
3	10721 8-192-300F-00-Z3415-00004-SE-1	73412	12580	-375	184.6105	0.0007830573	-10500	-374.99	184.6105	24.1	
4	10721 8-192-300F-00-Z3415-00004-SE-1	73413	12580	-400	184.6029	0.0004995472	-14700	-400	184.6029	24.1	
5	10721 8-192-300F-00-Z3415-00004-SE-1	73414	12580	-25	288.3093	0.0009684451	-96.4	-24.99	288.3093	24	
6	10721 8-192-300F-00-Z3415-00004-SE-1	73415	12580	-50	243.2882	0.0006212282	-108	-49.97	243.2882	24	
7	10721 8-192-300F-00-Z3415-00004-SE-1	73416	12580	-100	212.5664	0.0008029708	-131	-99.98	212.5664	24	
8	10721 8-192-300F-00-Z3415-00004-SE-1	73417	12580	-150	199.0388	0.0007781934	-139	-150.01	199.0388	24	
9	10721 8-192-300F-00-Z3415-00004-SE-1	73418	12580	-200	190.9591	0.0007236861	-146	-199.98	190.9591	24	
10	10721 8-192-300F-00-Z3415-00004-SE-1	73419	12580	-240	186.3071	0.0009184273	-148	-239.98	186.3071	24	
11	10721 8-192-300F-00-Z3415-00004-SE-1	73420	12580	-250	185.27	0.000791718	-161	-249.99	185.27	24	
12	10721 8-192-300F-00-Z3415-00004-SE-1	73421	12580	-260	184.2726	0.000497588	-161	-260	184.2726	24	
13	10721 8-192-300F-00-Z3415-00004-SE-1	73422	12580	-270	183.5169	0.0006840449	-205	-270	183.5169	24.1	
14	10721 8-192-300F-00-Z3415-00004-SE-1	73423	12580	-280	183.2061	0.0004665193	-466	-280.01	183.2061	24.1	

Want to Generate ERDs? You can do that too!

- Click **File**, then **Data Modeler**, then **Import**, then **Data Dictionary**
- Select DB Connection (e.g. INT2R_LB_LOCAL), and click Next
- Select a schema (e.g. CMS_HGC_HGCAL_COND), and click Next
- Select all the objects in the schema, and click Next
- Resulting ERD:



SQC, PQC and their Analysis Codes

- [HexDaq](#) is used for the output of raw data test results.
- SQC (sensor Quality Control) tests (such as IV and CV) tests the actual Si sensors. Analysis code for the results is [lcd hgcal analysis workflow S](#).
- PQC (Process Quality Control) is performed on test structures. Analysis codes is [analysis-pqc](#).

