Database development and usage for prototype and pre-series testing

Ali Al Kadhim

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

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

- A unified HGCAL 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.

Hexaboard — (PCB)
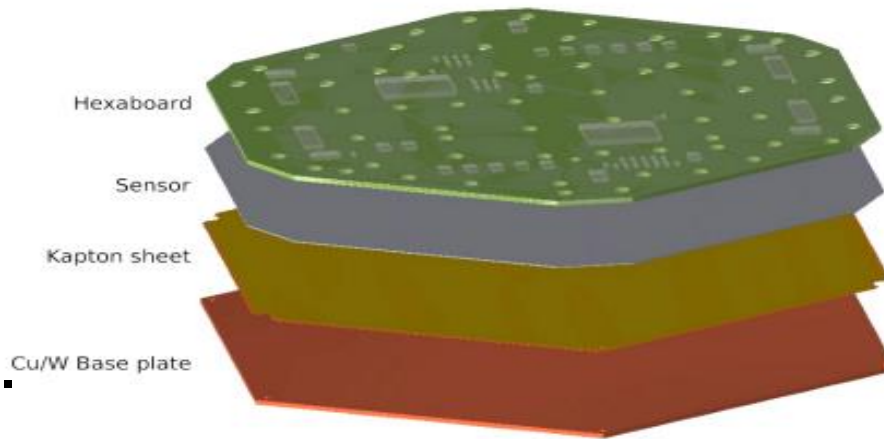
Sensor — Si 8" sensor

Kapton sheet — For electrical insulation from base plate

Cu/W Base plate — For placement onto cassettes

CE-E Si module

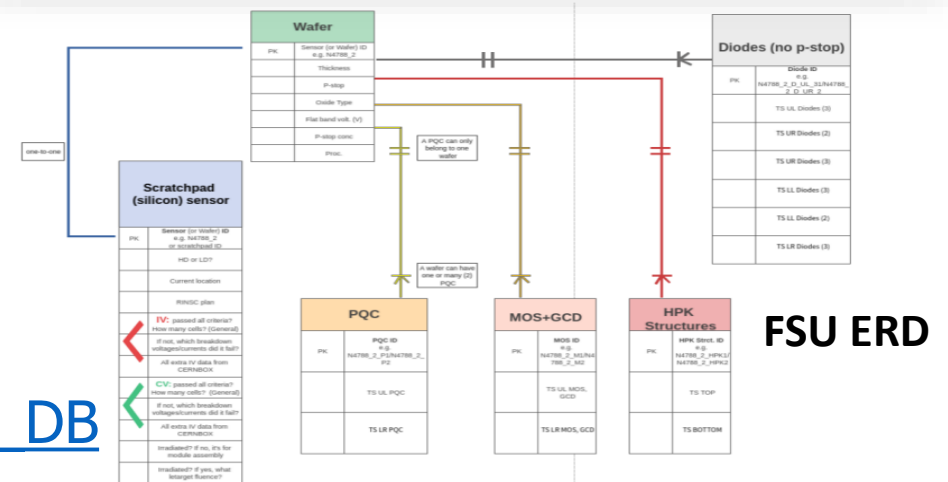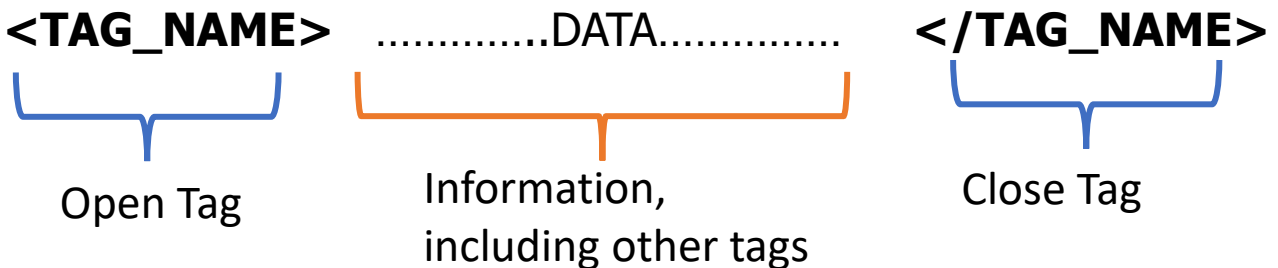# Spreadsheets to Database

- 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	Thick-ness	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

									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)
Sensor ID	Scratch pad ID	Thick-ness	P-Stop	Oxide type	Flat band volt. (V)	P-Stop conc.	Proc.							
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

FSU ERD

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: **<TAG_NAME>** …………..DATA…………… **</TAG_NAME>**

Open Tag

Information, including other tags

Close Tag

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).
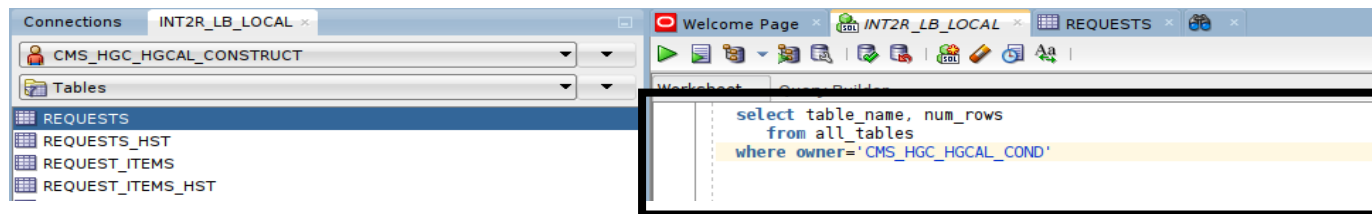
Sample ERD

Sample SQL query
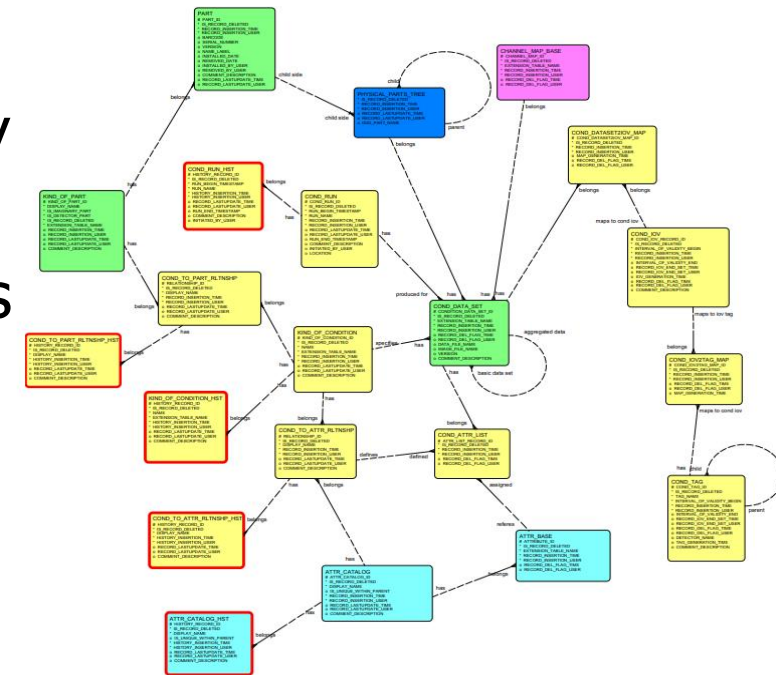
Connecting to the DB with SQLDeveloper Tutorial (linux)

- 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.

```
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!
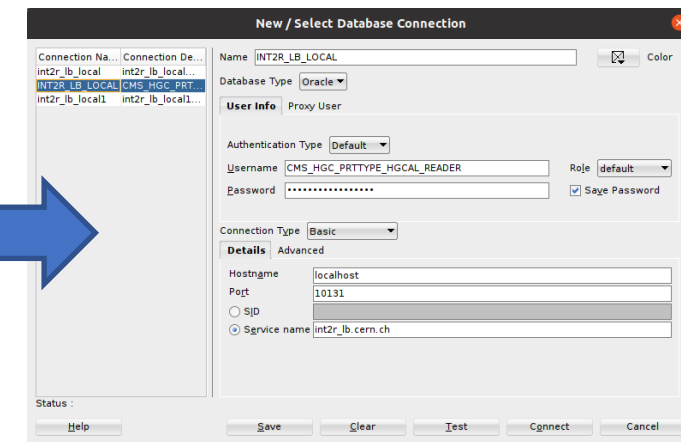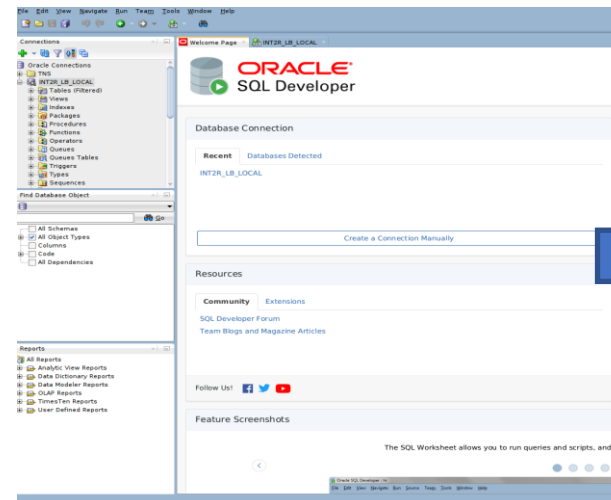
GUIs! OMS and DBA

- 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

(types HD 120um, LD 200um, and LD 300um)

inherits

CHILDREN

- To view OMS: go here
  - We have plans to make a OMS a complete HGCAL GUI.

The DB is XML schema

XML Template for HGC_CERN_SENSOR_IV table

Viewed in `SQLDeveloper`

Ali Al Kadhim - HGCAL DB

13

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-N1308-000017 8-432-120E-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/AliAlkadhim/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) databse, 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 HGCAL!

- Everyone from HGCAL (testing, module assembly, etc.) is encouraged to upload to the DB!

- If you need **permissions to upload to HGCAL DB**, please contact Imran (Muhammad.Imran@cern.ch), Rao (muhammad.atif.shad.rao@cern.ch), or **Umesh** (joshi@fnal.gov).

- Attend an HGCAL 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 "openjkd 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)

- 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
```

Where username is your lxplus username.

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

```
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)
        )
    )
)
```

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**"
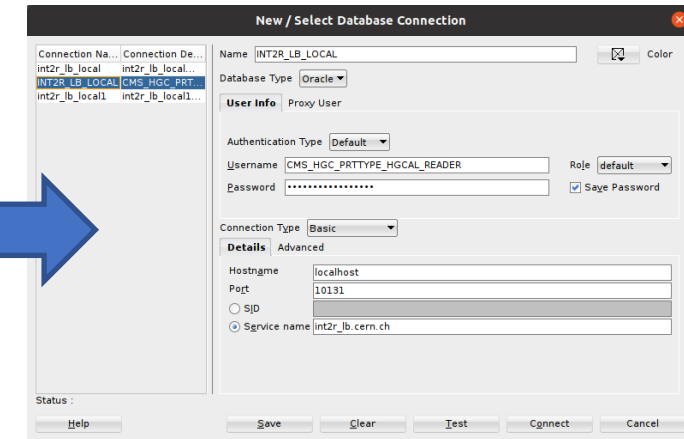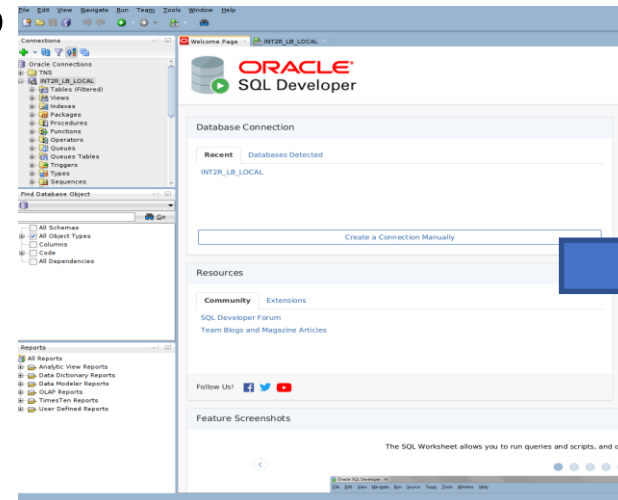
Connecting to the Database with SQLDeveloper (3)

- 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 HGCAL 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%20HGCAL%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 HGCAL 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) databse, 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/dbspool/logs/hgc/int2r/filename.xml` for log information.

Uploading to HGCALDB: 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.)

Queries in SQLDeveloper Example 2

- See everything in the "HGC_CERN_SEN SOR_IV" table.

You can make SQL Queries infinitely complex!

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_workflows**.
- PQC (Process Quality Control) is performed on test structures. Analysis codes is **analysis-pqc**.

sensor cells (diodes)

p^+ implant
n^+ implant
p-stop implant
polysilicon
contact
aluminum
passivation opening
wafer border
dicing street

PQC (halfmoon) sets (for quality assurance and irradiation tests)

Diodes   MOS capacitors   Gate-controlled diodes

(+ MOSFET, Van-der-pauw structures, etc., not yet documented)