Bookkeeping Database

Luca Tomassetti [University of Ferrara & INFN]

Oct. 7, 2009 – X SuperB General Meeting, SLAC

Summary

- Database Schema
- Implementation & Queries
- (public) test & (near) future developments
- Towards a Distributed Production Software

Database Schema Validation

Database Schema

- Sep 28th: SBK Meeting
  Validation of the proposed schema

- Main features:
  - Production
  - Full and Fast jobs
  - Merging
  - Software releases

- Open Questions:
  - Generators's Parameters
  - Machine / Generator / Input Files (FullSim)
  - Uniqueness of values

Database Schema

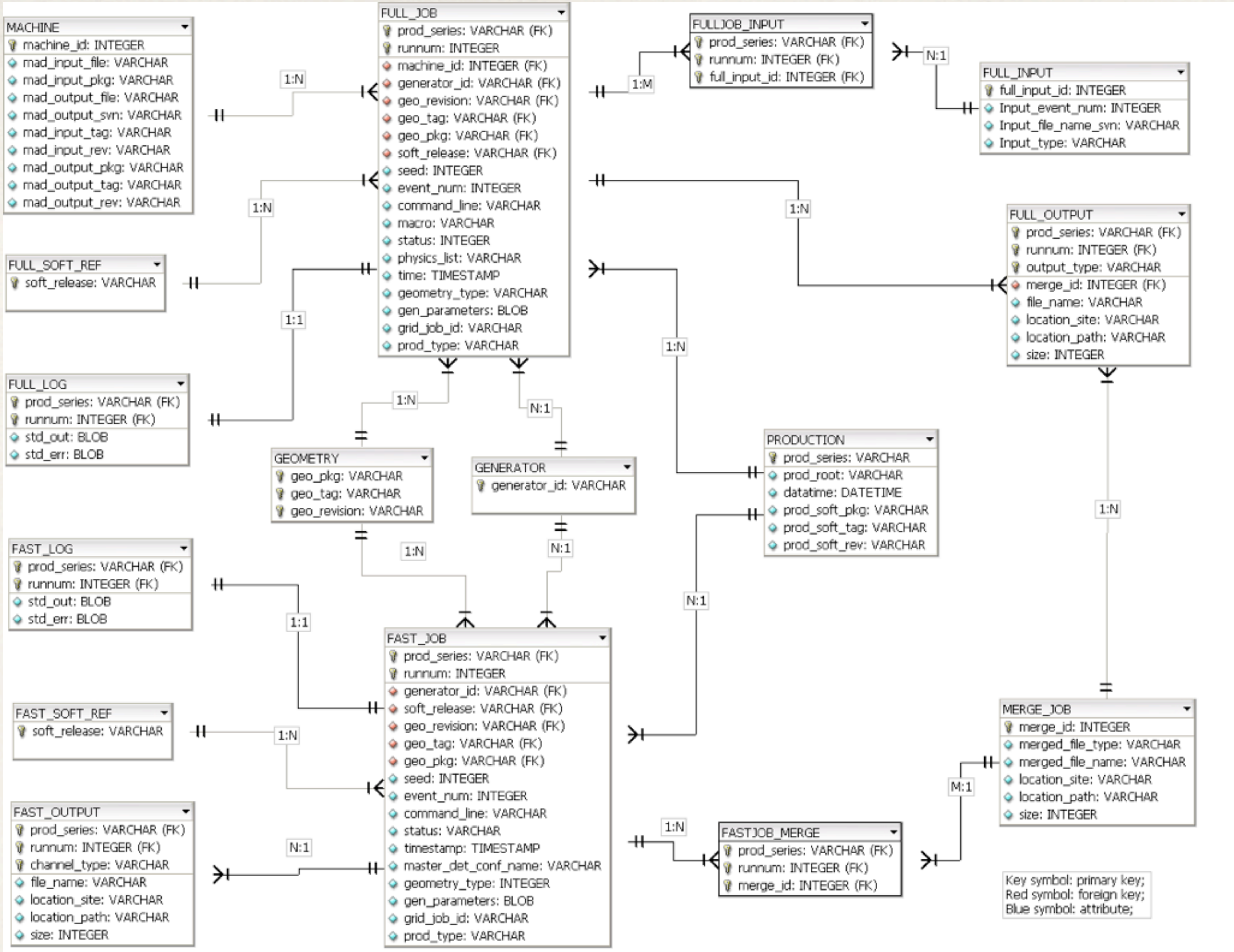
- Sep 28th: SBK Meeting
  Validation of the proposed schema

- Main features:
  - Production
  - Full and Fast jobs
  - Merging
  - Software releases

- Open Questions:
  - NSI Generators's Parameters
  - NSI Machine / Generator / Input Files (FullSim)
  - VI Uniqueness of values

NSI: Not So Important!
VI: Very Important!

Features



MACHINE
machine_id: INTEGER
mad_input_file: VARCHAR
mad_input_pkg: VARCHAR
mad_output_file: VARCHAR
mad_output_svn: VARCHAR
mad_input_tag: VARCHAR
mad_input_rev: VARCHAR
mad_output_pkg: VARCHAR
mad_output_tag: VARCHAR
mad_output_rev: VARCHAR

FULL_SOFT_REF
soft_release: VARCHAR

FULL_LOG
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
std_out: BLOB
std_err: BLOB

FULL_JOB
prod_series: VARCHAR (FK)
runnum: INTEGER
machine_id: INTEGER (FK)
generator_id: VARCHAR (FK)
geo_revision: VARCHAR (FK)
geo_tag: VARCHAR (FK)
geo_pkg: VARCHAR (FK)
soft_release: VARCHAR (FK)
seed: INTEGER
event_num: INTEGER
command_line: VARCHAR
macro: VARCHAR
status: INTEGER
physics_list: VARCHAR
time: TIMESTAMP
geometry_type: VARCHAR
gen_parameters: BLOB
grid_job_id: VARCHAR
prod_type: VARCHAR

FULLJOB_INPUT
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
full_input_id: INTEGER (FK)

FULL_INPUT
full_input_id: INTEGER
Input_event_num: INTEGER
Input_file_name_svn: VARCHAR
Input_type: VARCHAR

FULL_OUTPUT
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
output_type: VARCHAR
merge_id: INTEGER (FK)
file_name: VARCHAR
location_site: VARCHAR
location_path: VARCHAR
size: INTEGER

GEOMETRY
geo_pkg: VARCHAR
geo_tag: VARCHAR
geo_revision: VARCHAR

GENERATOR
generator_id: VARCHAR

PRODUCTION
prod_series: VARCHAR
prod_root: VARCHAR
datatime: DATETIME
prod_soft_pkg: VARCHAR
prod_soft_tag: VARCHAR
prod_soft_rev: VARCHAR

FAST_LOG
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
std_out: BLOB
std_err: BLOB

FAST_SOFT_REF
soft_release: VARCHAR

FAST_OUTPUT
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
channel_type: VARCHAR
file_name: VARCHAR
location_site: VARCHAR
location_path: VARCHAR
size: INTEGER

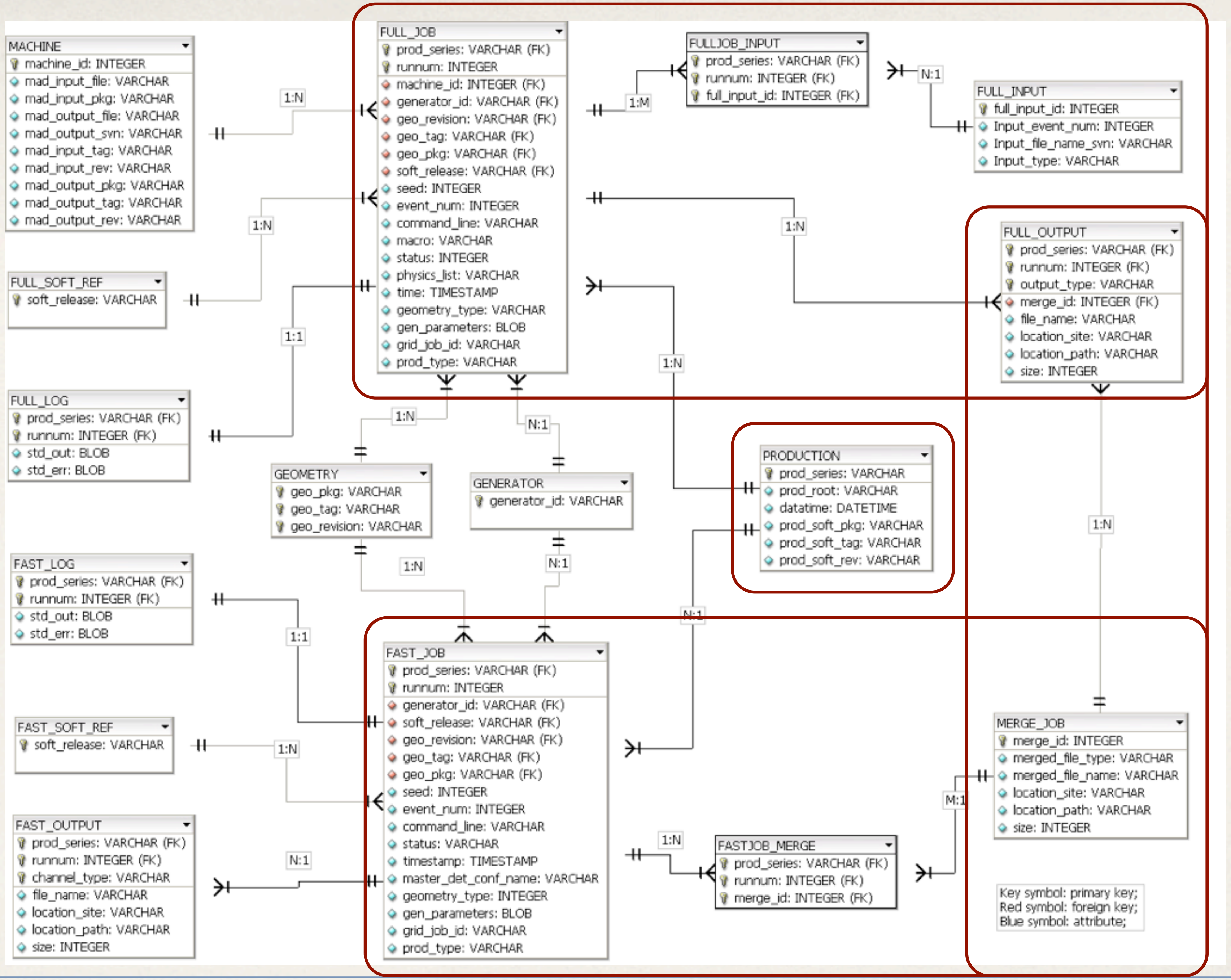
FAST_JOB
prod_series: VARCHAR (FK)
runnum: INTEGER
generator_id: VARCHAR (FK)
soft_release: VARCHAR (FK)
geo_revision: VARCHAR (FK)
geo_tag: VARCHAR (FK)
geo_pkg: VARCHAR (FK)
seed: INTEGER
event_num: INTEGER
command_line: VARCHAR
status: VARCHAR
timestamp: TIMESTAMP
master_det_conf_name: VARCHAR
geometry_type: INTEGER
gen_parameters: BLOB
grid_job_id: VARCHAR
prod_type: VARCHAR

MERGE_JOB
merge_id: INTEGER
merged_file_type: VARCHAR
merged_file_name: VARCHAR
location_site: VARCHAR
location_path: VARCHAR
size: INTEGER

FASTJOB_MERGE
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
merge_id: INTEGER (FK)

Key symbol: primary key;
Red symbol: foreign key;
Blue symbol: attribute;

Database Schema



MACHINE
- machine_id: INTEGER
- mad_input_file: VARCHAR
- mad_input_pkg: VARCHAR
- mad_output_file: VARCHAR
- mad_output_svn: VARCHAR
- mad_input_tag: VARCHAR
- mad_input_rev: VARCHAR
- mad_output_pkg: VARCHAR
- mad_output_tag: VARCHAR
- mad_output_rev: VARCHAR

FULL_SOFT_REF
- soft_release: VARCHAR

FULL_LOG
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- std_out: BLOB
- std_err: BLOB

FAST_LOG
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- std_out: BLOB
- std_err: BLOB

FAST_SOFT_REF
- soft_release: VARCHAR

FAST_OUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- channel_type: VARCHAR
- file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

FULL_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- machine_id: INTEGER (FK)
- generator_id: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- macro: VARCHAR
- status: INTEGER
- physics_list: VARCHAR
- time: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

GEOMETRY
- geo_pkg: VARCHAR
- geo_tag: VARCHAR
- geo_revision: VARCHAR

GENERATOR
- generator_id: VARCHAR

FAST_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- generator_id: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- status: VARCHAR
- timestamp: TIMESTAMP
- master_det_conf_name: VARCHAR
- geometry_type: INTEGER
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

FULLJOB_INPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- full_input_id: INTEGER (FK)

FULL_INPUT
- full_input_id: INTEGER
- Input_event_num: INTEGER
- Input_file_name_svn: VARCHAR
- Input_type: VARCHAR

FULL_OUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- output_type: VARCHAR
- merge_id: INTEGER (FK)
- file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

PRODUCTION
- prod_series: VARCHAR
- prod_root: VARCHAR
- datatime: DATETIME
- prod_soft_pkg: VARCHAR
- prod_soft_tag: VARCHAR
- prod_soft_rev: VARCHAR

MERGE_JOB
- merge_id: INTEGER
- merged_file_type: VARCHAR
- merged_file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

FASTJOB_MERGE
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- merge_id: INTEGER (FK)

Key symbol: primary key;
Red symbol: foreign key;
Blue symbol: attribute;

Database Schema

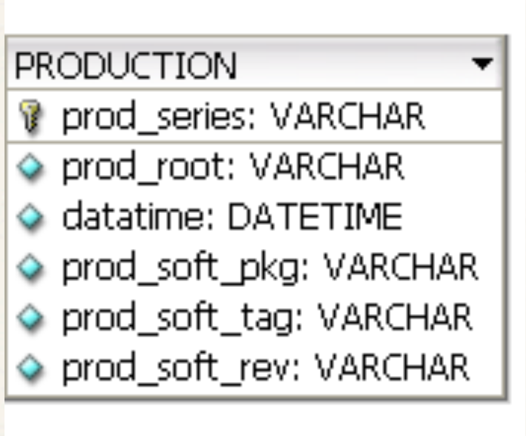
Production

- Each production is a row identified by the prod_series

- All stuff related to a production refers (from other tables) to the corresponding row by means of prod_series value

- prod_root:
  common to Full and Fast jobs

- datatime:
  launch of the production



PRODUCTION

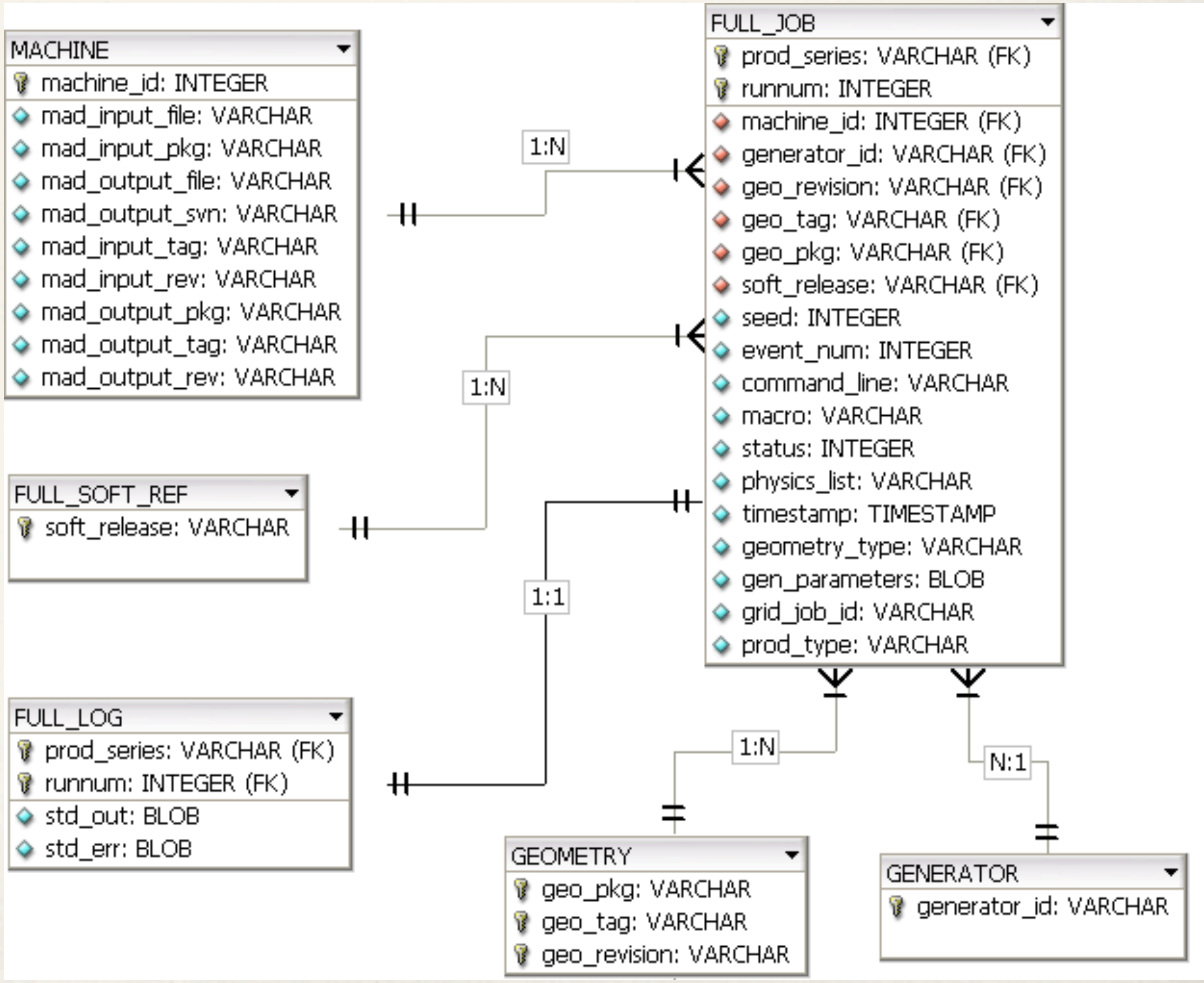
- prod_series: VARCHAR
- prod_root: VARCHAR
- datatime: DATETIME
- prod_soft_pkg: VARCHAR
- prod_soft_tag: VARCHAR
- prod_soft_rev: VARCHAR

Full Job

- prod_series AND runnum identify a job
  - Many job in a production
  - The same runnum in different productions

- machine_id
- generator_id
- geo_...
- soft_release

FK



MACHINE
- machine_id: INTEGER
- mad_input_file: VARCHAR
- mad_input_pkg: VARCHAR
- mad_output_file: VARCHAR
- mad_output_svn: VARCHAR
- mad_input_tag: VARCHAR
- mad_input_rev: VARCHAR
- mad_output_pkg: VARCHAR
- mad_output_tag: VARCHAR
- mad_output_rev: VARCHAR

FULL_SOFT_REF
- soft_release: VARCHAR

FULL_LOG
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- std_out: BLOB
- std_err: BLOB

FULL_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- machine_id: INTEGER (FK)
- generator_id: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- macro: VARCHAR
- status: INTEGER
- physics_list: VARCHAR
- timestamp: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

GEOMETRY
- geo_pkg: VARCHAR
- geo_tag: VARCHAR
- geo_revision: VARCHAR

GENERATOR
- generator_id: VARCHAR

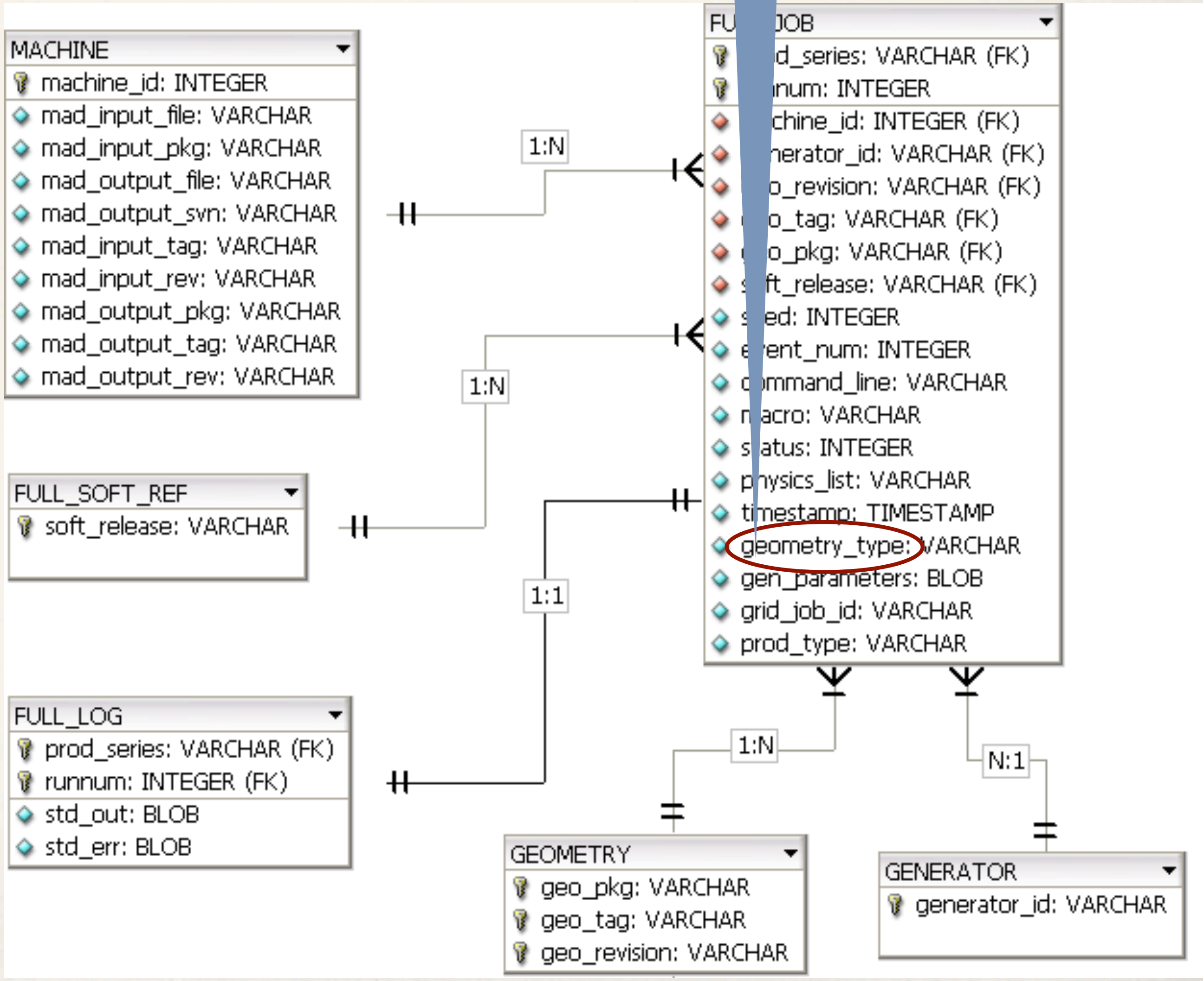
Full Job

geometry_type: DG0, DG1, ... (as defined by the DGWG)

The "real" geometry is defined in the Geometry package

- prod_series AND runnum identify a job
  - Many job in a production
  - The same runnum in different productions
- machine_id
- generator_id
- geo_...
- soft_release

} FK



MACHINE
- machine_id: INTEGER
- mad_input_file: VARCHAR
- mad_input_pkg: VARCHAR
- mad_output_file: VARCHAR
- mad_output_svn: VARCHAR
- mad_input_tag: VARCHAR
- mad_input_rev: VARCHAR
- mad_output_pkg: VARCHAR
- mad_output_tag: VARCHAR
- mad_output_rev: VARCHAR

FULL_SOFT_REF
- soft_release: VARCHAR

FULL_LOG
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- std_out: BLOB
- std_err: BLOB

FU JOB
- d_series: VARCHAR (FK)
- num: INTEGER
- chine_id: INTEGER (FK)
- nerator_id: VARCHAR (FK)
- o_revision: VARCHAR (FK)
- o_tag: VARCHAR (FK)
- o_pkg: VARCHAR (FK)
- ft_release: VARCHAR (FK)
- ed: INTEGER
- ent_num: INTEGER
- ommand_line: VARCHAR
- acro: VARCHAR
- atus: INTEGER
- physics_list: VARCHAR
- timestamp: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

GEOMETRY
- geo_pkg: VARCHAR
- geo_tag: VARCHAR
- geo_revision: VARCHAR

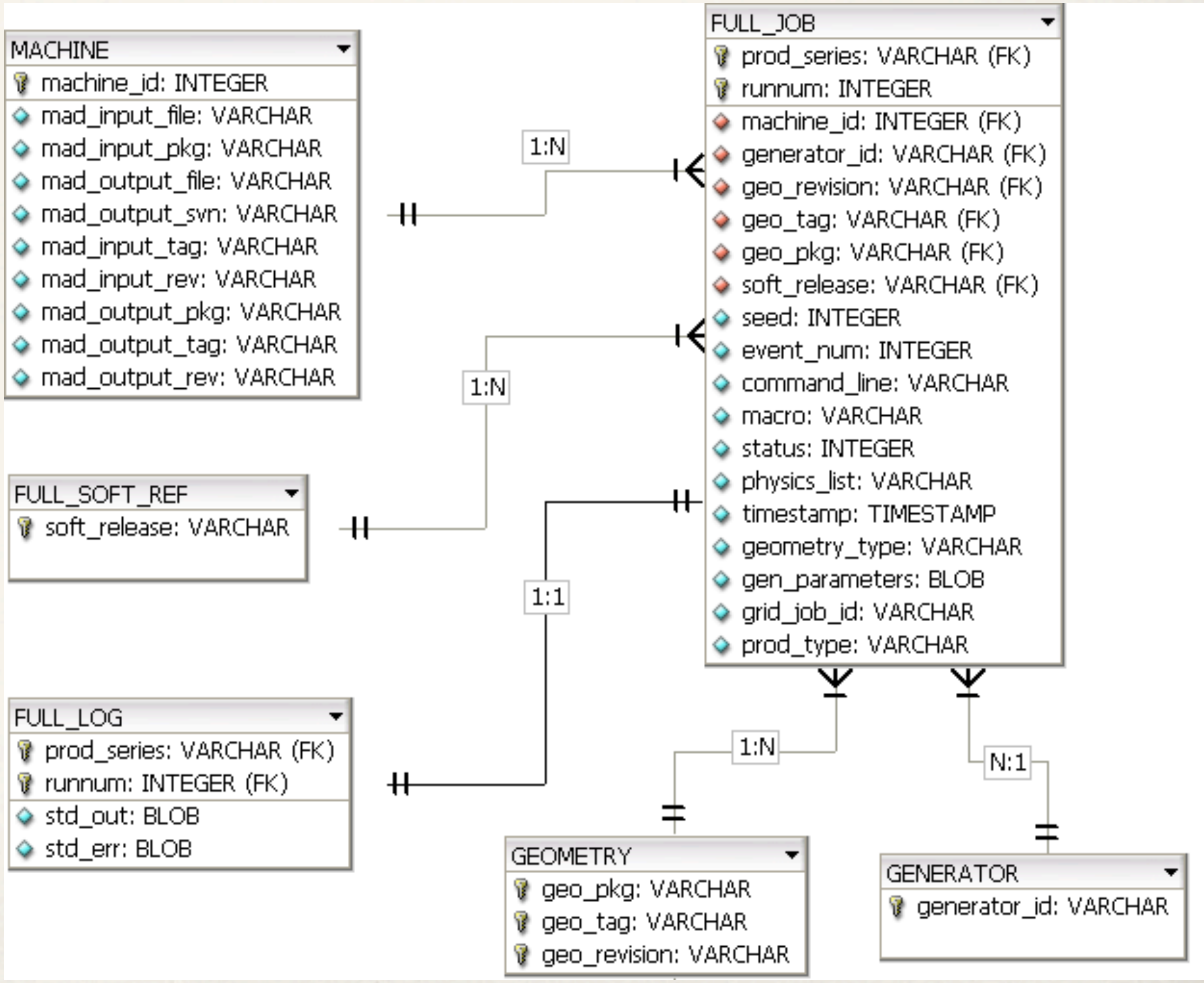
GENERATOR
- generator_id: VARCHAR

Full Job

- prod_series AND runnum identify a job
  - Many job in a production
  - The same runnum in different productions

- machine_id
- generator_id
- geo_...
- soft_release

FK



MACHINE
- machine_id: INTEGER
- mad_input_file: VARCHAR
- mad_input_pkg: VARCHAR
- mad_output_file: VARCHAR
- mad_output_svn: VARCHAR
- mad_input_tag: VARCHAR
- mad_input_rev: VARCHAR
- mad_output_pkg: VARCHAR
- mad_output_tag: VARCHAR
- mad_output_rev: VARCHAR

FULL_SOFT_REF
- soft_release: VARCHAR

FULL_LOG
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- std_out: BLOB
- std_err: BLOB

FULL_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- machine_id: INTEGER (FK)
- generator_id: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- macro: VARCHAR
- status: INTEGER
- physics_list: VARCHAR
- timestamp: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

GEOMETRY
- geo_pkg: VARCHAR
- geo_tag: VARCHAR
- geo_revision: VARCHAR

GENERATOR
- generator_id: VARCHAR

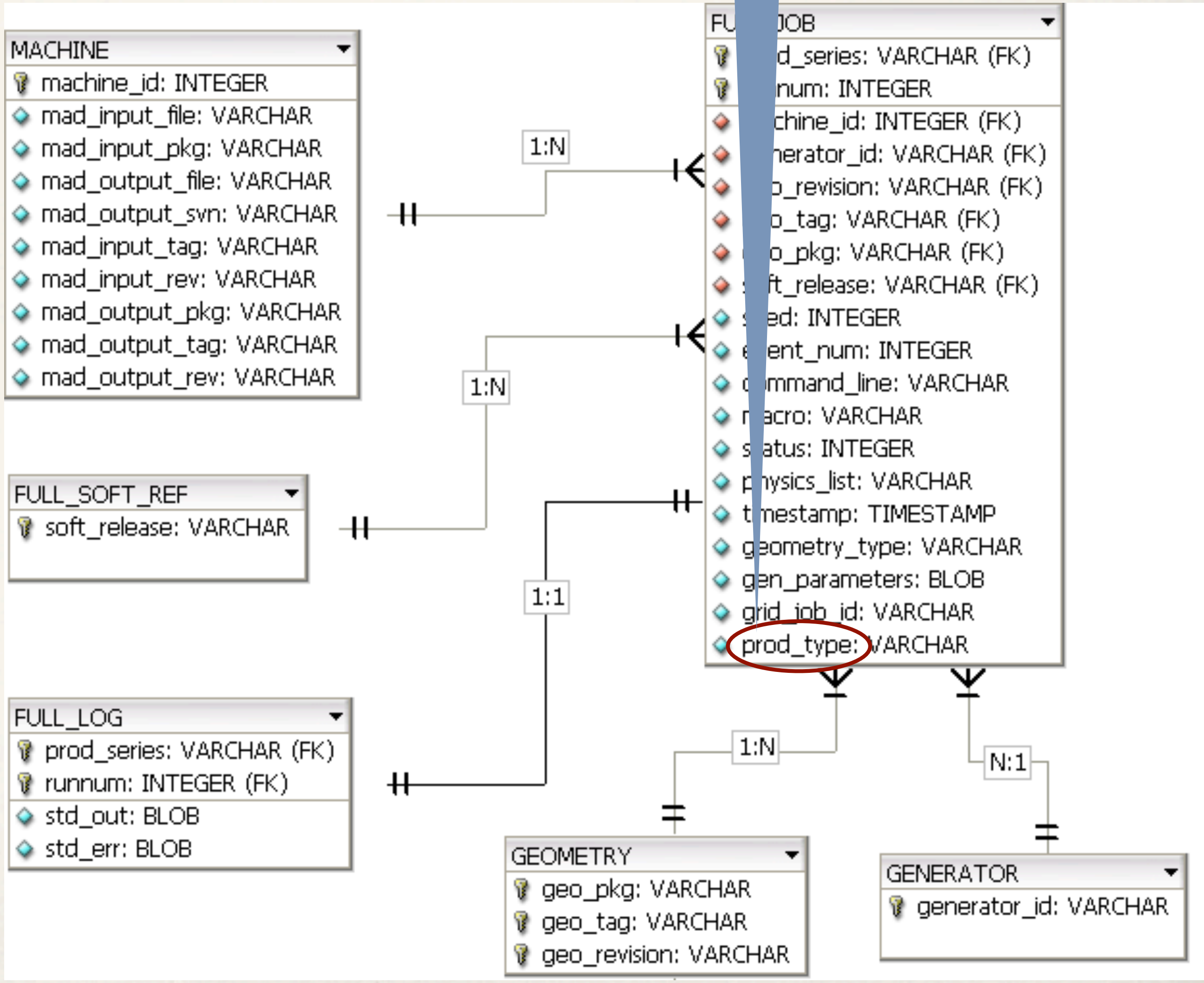
Full Job

prod_type: A (Machine BG), B (Physics BG), C (BG Physics), D (Physics)

Sub-classification? e.g. A1: Touschek, A2: Beam, A3: Track... B1: Bha-bha to neutrons,...

- prod_series AND runnum identify a job
  - Many job in a production
  - The same runnum in different productions
- machine_id
- generator_id
- geo_...
- soft_release

} FK



MACHINE
- machine_id: INTEGER
- mad_input_file: VARCHAR
- mad_input_pkg: VARCHAR
- mad_output_file: VARCHAR
- mad_output_svn: VARCHAR
- mad_input_tag: VARCHAR
- mad_input_rev: VARCHAR
- mad_output_pkg: VARCHAR
- mad_output_tag: VARCHAR
- mad_output_rev: VARCHAR

FULL_SOFT_REF
- soft_release: VARCHAR

FULL_LOG
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- std_out: BLOB
- std_err: BLOB

FU JOB
- d_series: VARCHAR (FK)
- num: INTEGER
- chine_id: INTEGER (FK)
- nerator_id: VARCHAR (FK)
- o_revision: VARCHAR (FK)
- o_tag: VARCHAR (FK)
- o_pkg: VARCHAR (FK)
- ft_release: VARCHAR (FK)
- ed: INTEGER
- ent_num: INTEGER
- mmand_line: VARCHAR
- acro: VARCHAR
- atus: INTEGER
- hysics_list: VARCHAR
- mestamp: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

GEOMETRY
- geo_pkg: VARCHAR
- geo_tag: VARCHAR
- geo_revision: VARCHAR

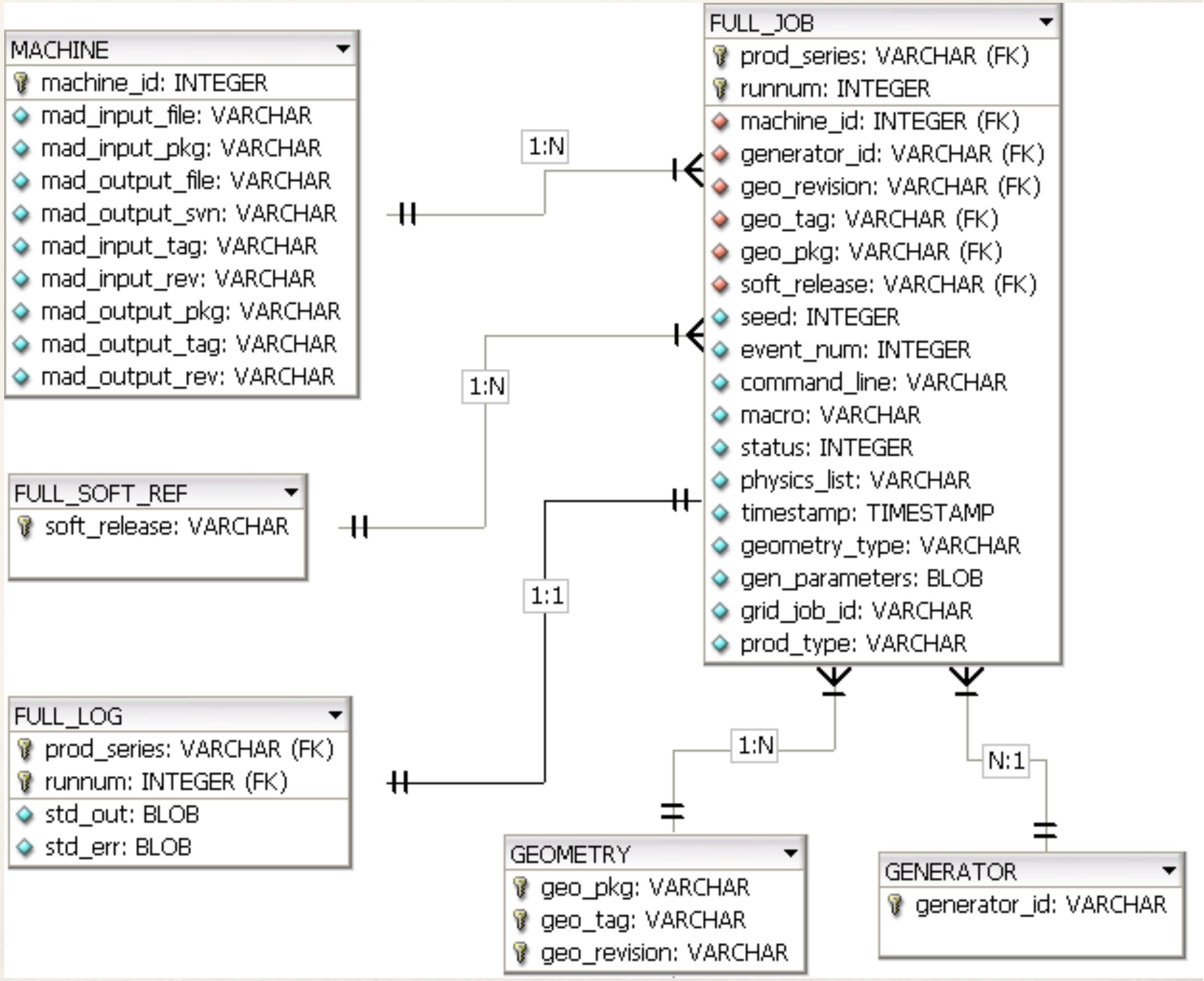
GENERATOR
- generator_id: VARCHAR

Full Job

- prod_series AND runnum identify a job
  - Many job in a production
  - The same runnum in different productions

- machine_id
- generator_id
- geo_...
- soft_release

FK



MACHINE
- machine_id: INTEGER
- mad_input_file: VARCHAR
- mad_input_pkg: VARCHAR
- mad_output_file: VARCHAR
- mad_output_svn: VARCHAR
- mad_input_tag: VARCHAR
- mad_input_rev: VARCHAR
- mad_output_pkg: VARCHAR
- mad_output_tag: VARCHAR
- mad_output_rev: VARCHAR

FULL_SOFT_REF
- soft_release: VARCHAR

FULL_LOG
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- std_out: BLOB
- std_err: BLOB

FULL_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- machine_id: INTEGER (FK)
- generator_id: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- macro: VARCHAR
- status: INTEGER
- physics_list: VARCHAR
- timestamp: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

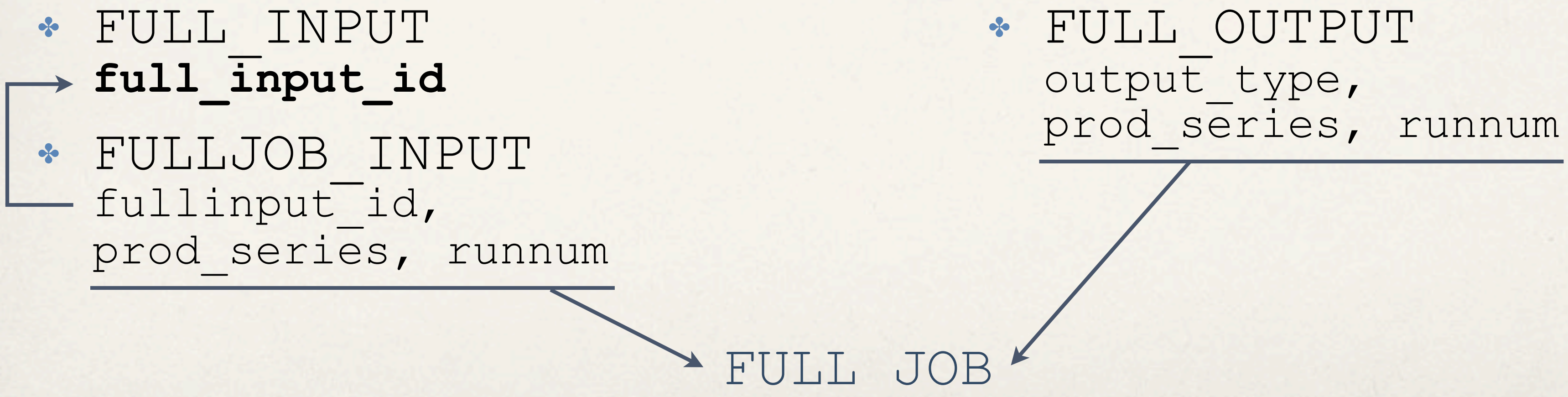
GEOMETRY
- geo_pkg: VARCHAR
- geo_tag: VARCHAR
- geo_revision: VARCHAR

GENERATOR
- generator_id: VARCHAR

Full Job I/O

- Input files:
  - Many per job
    (0,N)
  - The same input file for many jobs
    (1, M)

- Output files:
  - Many per job
    (1,N), typically 2
  - A given output file is produced by one and only one job



- Two tables:
  - FULL_INPUT
    full_input_id
  - FULLJOB_INPUT
    fullinput_id,
    prod_series, runnum

- One table:
  - FULL_OUTPUT
    output_type,
    prod_series, runnum

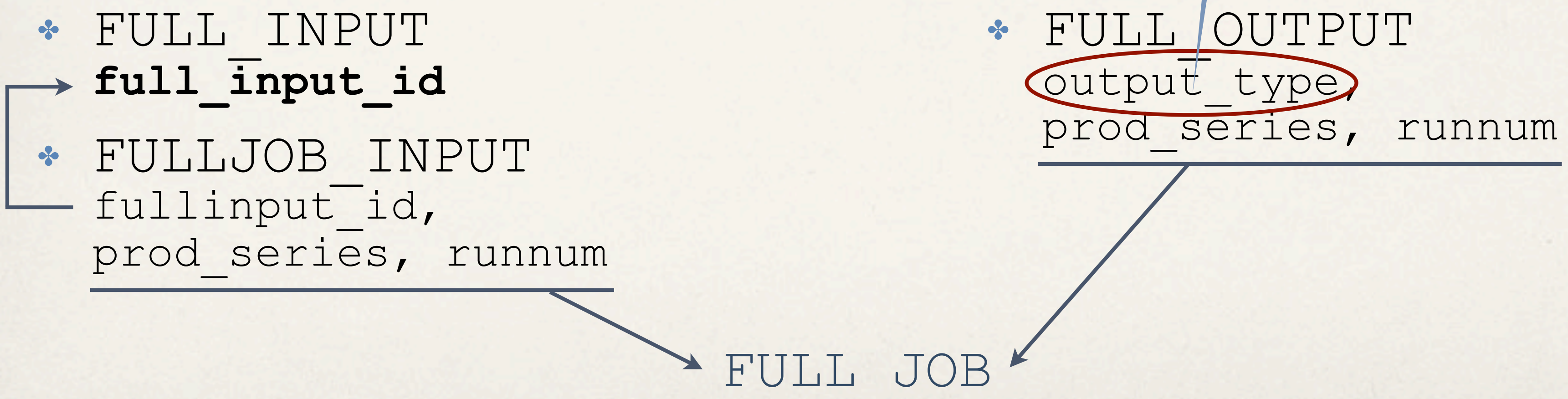
FULL_JOB

Full Job I/O

NOW: “input for fast”, “hits”
exaustive list?

- Input files:
  - Many per job (0,N)
  - The same input file for many jobs (1, M)

- Output files:
  - Many per job (1,N), typically 2
  - A given output file is produced by one and only one job



- Two tables:
  - FULL_INPUT
    full_input_id
  - FULLJOB_INPUT
    fullinput_id,
    prod_series, runnum

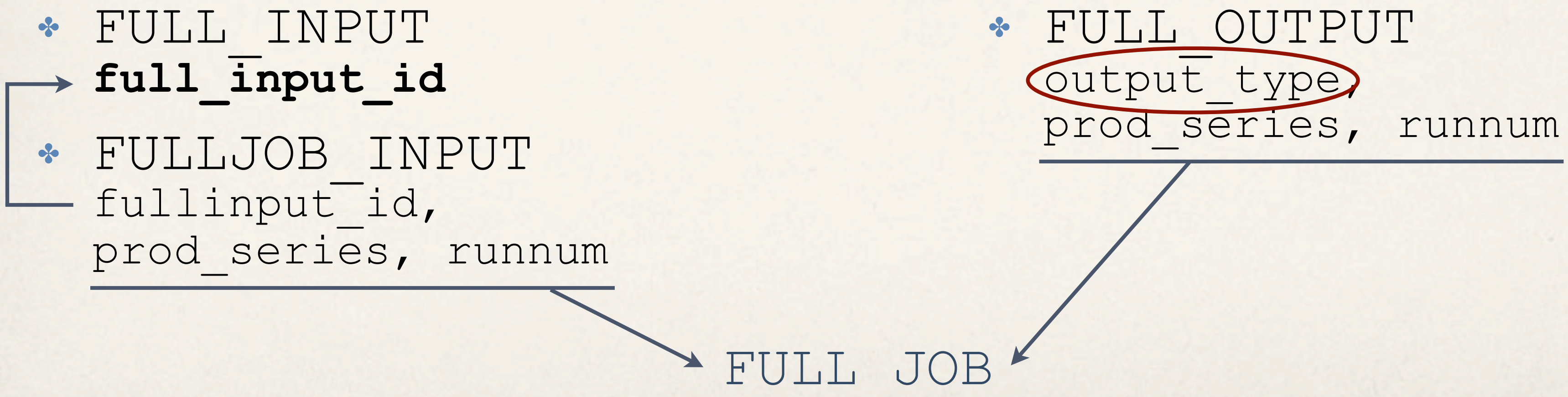
- One table:
  - FULL_OUTPUT
    output_type,
    prod_series, runnum

FULL_JOB

Full Job I/O

- Input files:
  - Many per job
    (0,N)
  - The same input file for many jobs
    (1, M)
- Two tables:

- Output files:
  - Many per job
    (1,N), typically 2
  - A given output file is produced by one and only one job
- One table:



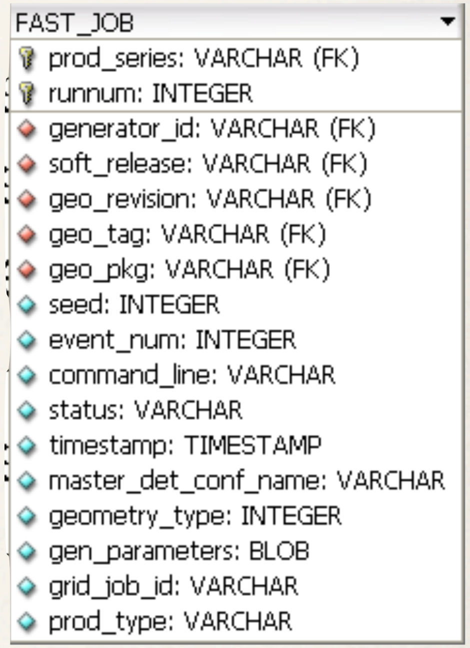
- `FULL_INPUT`
  `full_input_id`
- `FULLJOB_INPUT`
  `fullinput_id,`
  `prod_series, runnum`

- `FULL_OUTPUT`
  `output_type,`
  `prod_series, runnum`

`FULL_JOB`

Fast Job

- The same philosophy as Full Job
- Differences in input/output files
- Explicit reference to the master detector configuration file
  (which is anyway in the Geometry package)
- `prod_type`: FastSim; do we need deeper classification ?

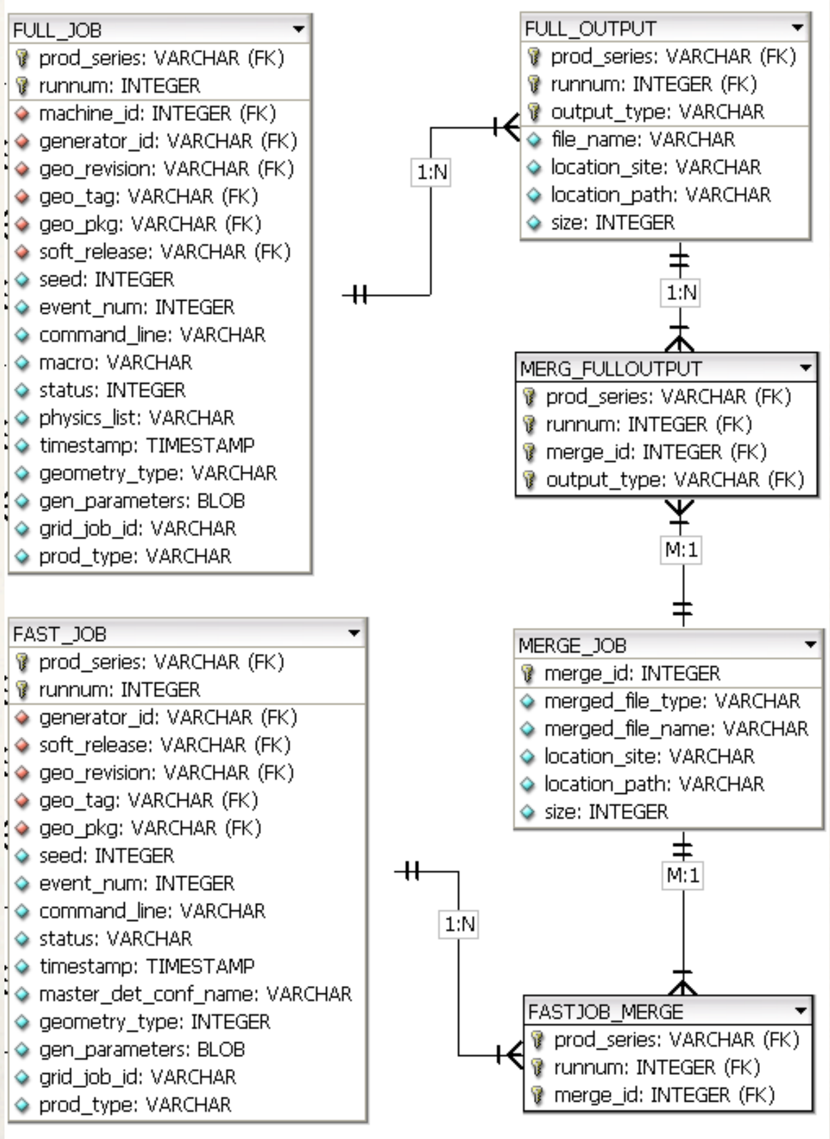


FAST_JOB

prod_series: VARCHAR (FK)
runnum: INTEGER
generator_id: VARCHAR (FK)
soft_release: VARCHAR (FK)
geo_revision: VARCHAR (FK)
geo_tag: VARCHAR (FK)
geo_pkg: VARCHAR (FK)
seed: INTEGER
event_num: INTEGER
command_line: VARCHAR
status: VARCHAR
timestamp: TIMESTAMP
master_det_conf_name: VARCHAR
geometry_type: INTEGER
gen_parameters: BLOB
grid_job_id: VARCHAR
prod_type: VARCHAR

Fast Job I/O

- Fastsim uses (one or more) **merge** files as input
- The same merge file can be used by many Fast jobs
- Two tables:
  - `MERGE_JOB` **`merge_id`**
  - `FASTJOB_MERGE` `merge_id, prod_series, runnum`



FULL_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- machine_id: INTEGER (FK)
- generator_id: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- macro: VARCHAR
- status: INTEGER
- physics_list: VARCHAR
- timestamp: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

FULL_OUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- output_type: VARCHAR
- file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

MERG_FULLOUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- merge_id: INTEGER (FK)
- output_type: VARCHAR (FK)

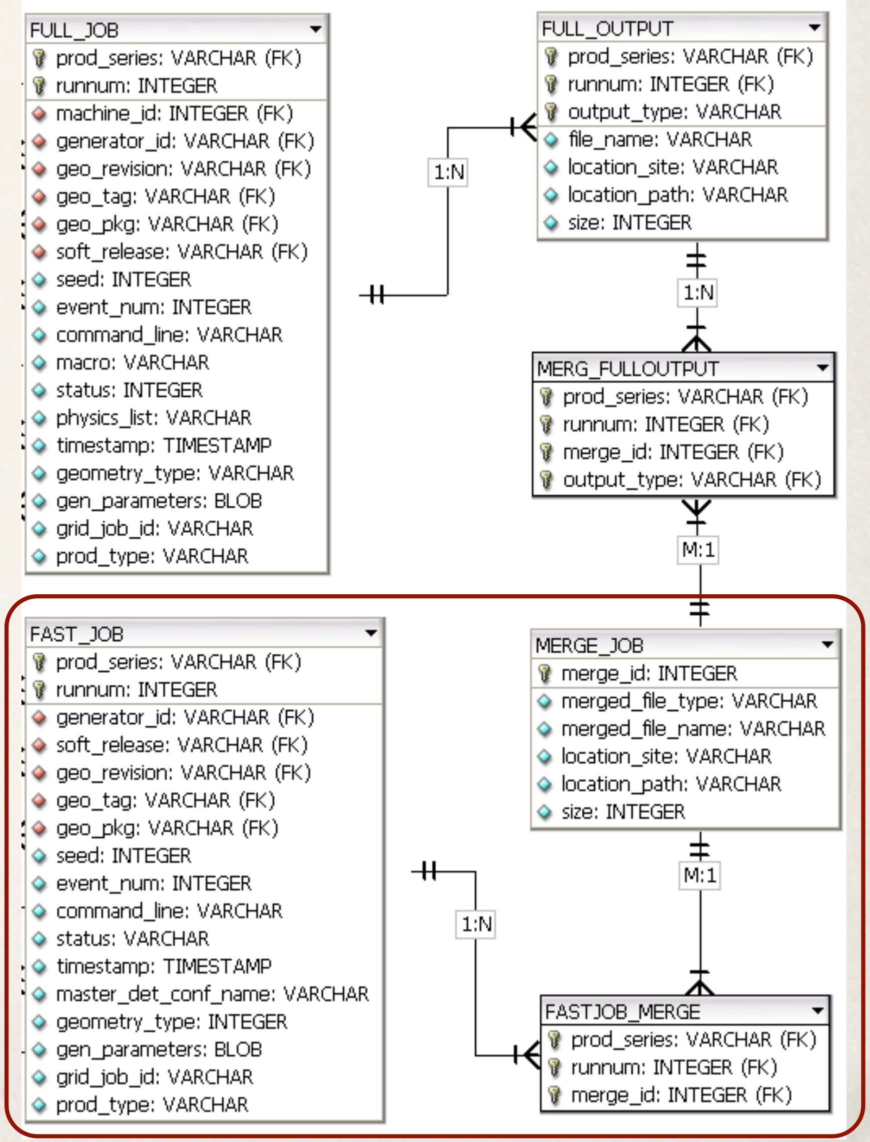
FAST_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- generator_id: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- status: VARCHAR
- timestamp: TIMESTAMP
- master_det_conf_name: VARCHAR
- geometry_type: INTEGER
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

MERGE_JOB
- merge_id: INTEGER
- merged_file_type: VARCHAR
- merged_file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

FASTJOB_MERGE
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- merge_id: INTEGER (FK)

Fast Job I/O

- Fastsim uses (one or more) **merge** files as input
- The same merge file can be used by many Fast jobs
- Two tables:
  - MERGE_JOB
merge_id
  - FASTJOB_MERGE
merge_id,
prod_series, runnum



FULL_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- machine_id: INTEGER (FK)
- generator_id: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- macro: VARCHAR
- status: INTEGER
- physics_list: VARCHAR
- timestamp: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

FULL_OUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- output_type: VARCHAR
- file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

MERG_FULLOUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- merge_id: INTEGER (FK)
- output_type: VARCHAR (FK)

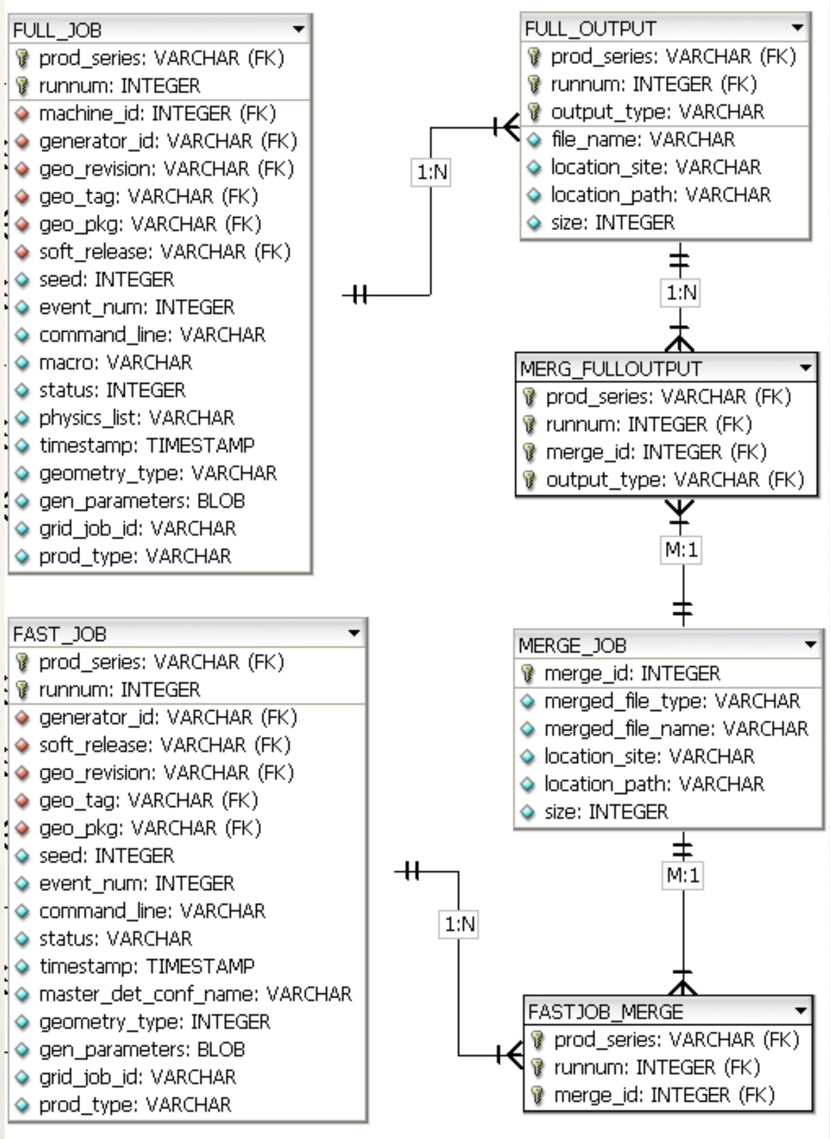
FAST_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- generator_id: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- status: VARCHAR
- timestamp: TIMESTAMP
- master_det_conf_name: VARCHAR
- geometry_type: INTEGER
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

MERGE_JOB
- merge_id: INTEGER
- merged_file_type: VARCHAR
- merged_file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

FASTJOB_MERGE
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- merge_id: INTEGER (FK)

Fast Job I/O – Merge files

- A Merge file consists of many Fullsim Output files (of the same type, i.e. produced by the same generator)
- A Fullsim Ouput file can be used in many Merge files
- A Fullsim Ouput file can be used in one and only one Merge files



FULL_JOB
prod_series: VARCHAR (FK)
runnum: INTEGER
machine_id: INTEGER (FK)
generator_id: VARCHAR (FK)
geo_revision: VARCHAR (FK)
geo_tag: VARCHAR (FK)
geo_pkg: VARCHAR (FK)
soft_release: VARCHAR (FK)
seed: INTEGER
event_num: INTEGER
command_line: VARCHAR
macro: VARCHAR
status: INTEGER
physics_list: VARCHAR
timestamp: TIMESTAMP
geometry_type: VARCHAR
gen_parameters: BLOB
grid_job_id: VARCHAR
prod_type: VARCHAR

FULL_OUTPUT
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
output_type: VARCHAR
file_name: VARCHAR
location_site: VARCHAR
location_path: VARCHAR
size: INTEGER

MERG_FULLOUTPUT
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
merge_id: INTEGER (FK)
output_type: VARCHAR (FK)

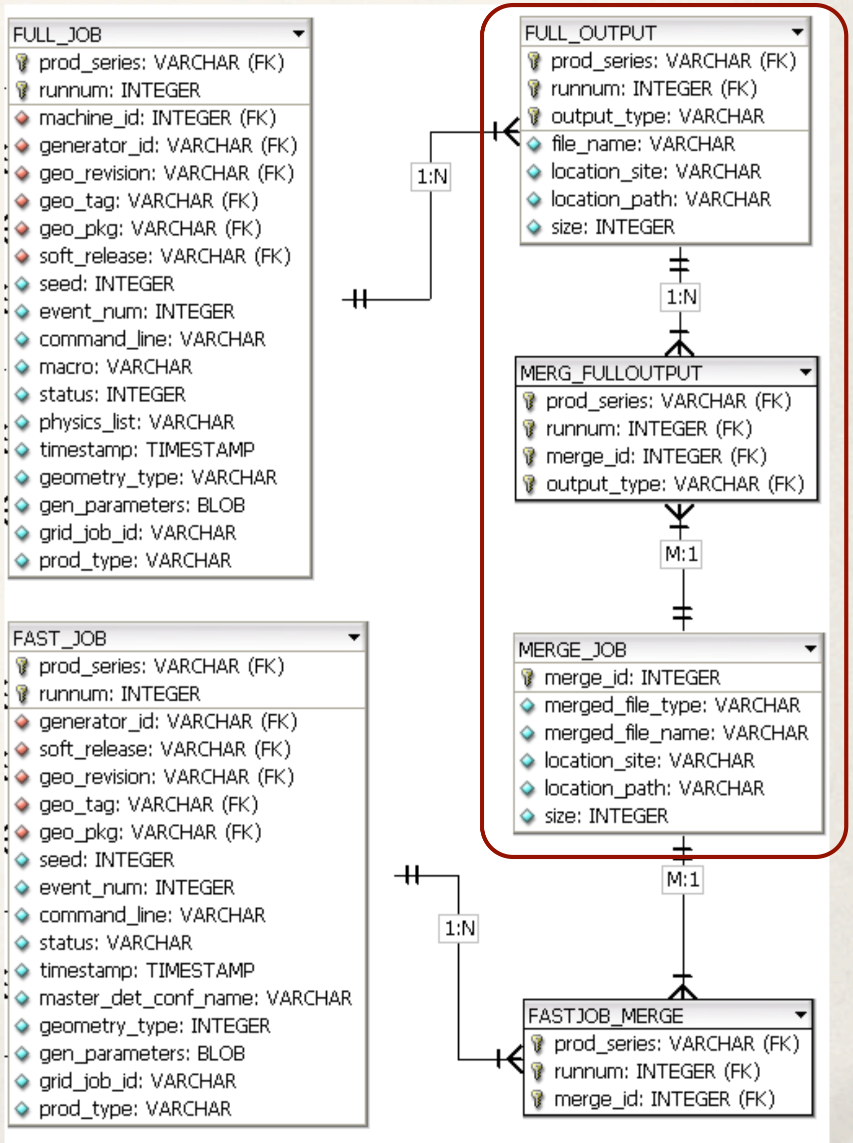
FAST_JOB
prod_series: VARCHAR (FK)
runnum: INTEGER
generator_id: VARCHAR (FK)
soft_release: VARCHAR (FK)
geo_revision: VARCHAR (FK)
geo_tag: VARCHAR (FK)
geo_pkg: VARCHAR (FK)
seed: INTEGER
event_num: INTEGER
command_line: VARCHAR
status: VARCHAR
timestamp: TIMESTAMP
master_det_conf_name: VARCHAR
geometry_type: INTEGER
gen_parameters: BLOB
grid_job_id: VARCHAR
prod_type: VARCHAR

MERGE_JOB
merge_id: INTEGER
merged_file_type: VARCHAR
merged_file_name: VARCHAR
location_site: VARCHAR
location_path: VARCHAR
size: INTEGER

FASTJOB_MERGE
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
merge_id: INTEGER (FK)

Fast Job I/O – Merge files

- A Merge file consists of many Fullsim Output files (of the same type, i.e. produced by the same generator)
- A Fullsim Ouput file can be used in many Merge files
- A Fullsim Ouput file can be used in one and only one Merge files



FULL_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- machine_id: INTEGER (FK)
- generator_id: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- macro: VARCHAR
- status: INTEGER
- physics_list: VARCHAR
- timestamp: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

FAST_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- generator_id: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- status: VARCHAR
- timestamp: TIMESTAMP
- master_det_conf_name: VARCHAR
- geometry_type: INTEGER
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

FULL_OUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- output_type: VARCHAR
- file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

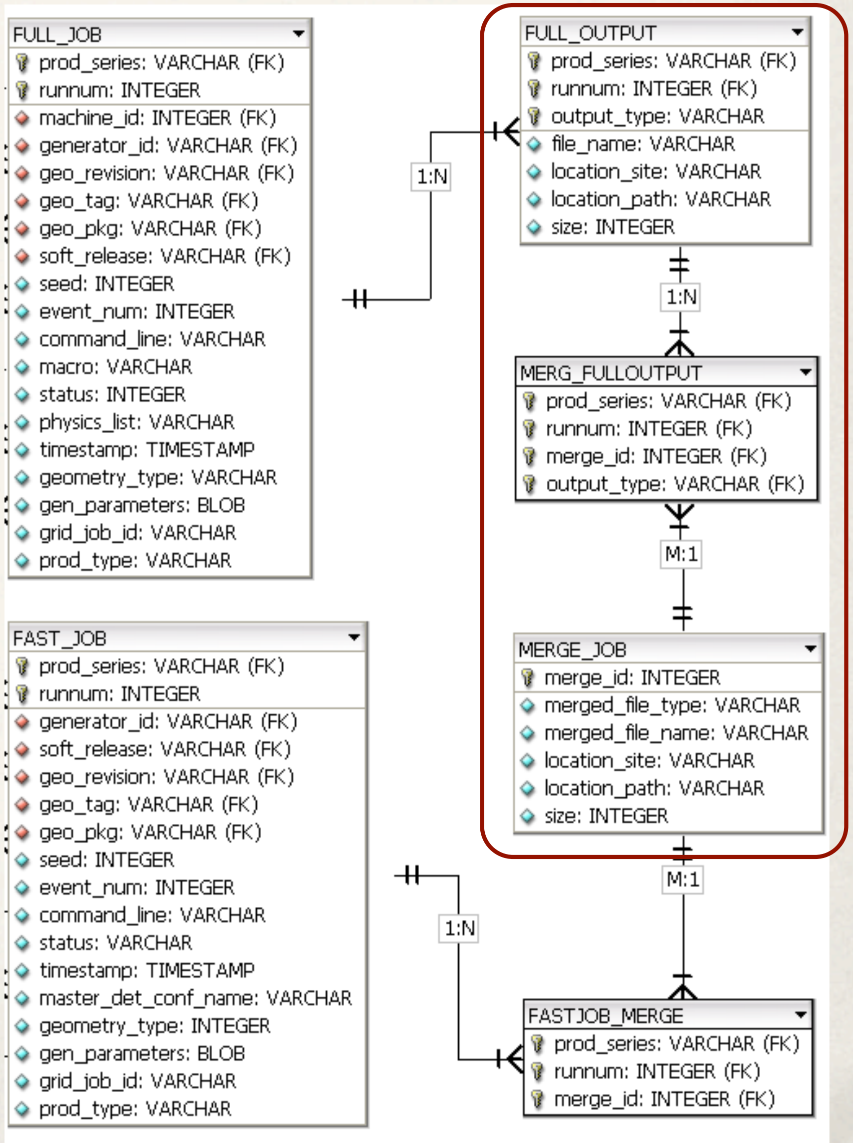
MERG_FULLOUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- merge_id: INTEGER (FK)
- output_type: VARCHAR (FK)

MERGE_JOB
- merge_id: INTEGER
- merged_file_type: VARCHAR
- merged_file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

FASTJOB_MERGE
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- merge_id: INTEGER (FK)

Fast Job I/O – Merge files

- A Merge file consists of many Fullsim Output files (of the same type, i.e. produced by the same generator)
- A Fullsim Ouput file can be used in many Merge files



FULL_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- machine_id: INTEGER (FK)
- generator_id: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- macro: VARCHAR
- status: INTEGER
- physics_list: VARCHAR
- timestamp: TIMESTAMP
- geometry_type: VARCHAR
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

FAST_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- generator_id: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- status: VARCHAR
- timestamp: TIMESTAMP
- master_det_conf_name: VARCHAR
- geometry_type: INTEGER
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

FULL_OUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- output_type: VARCHAR
- file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

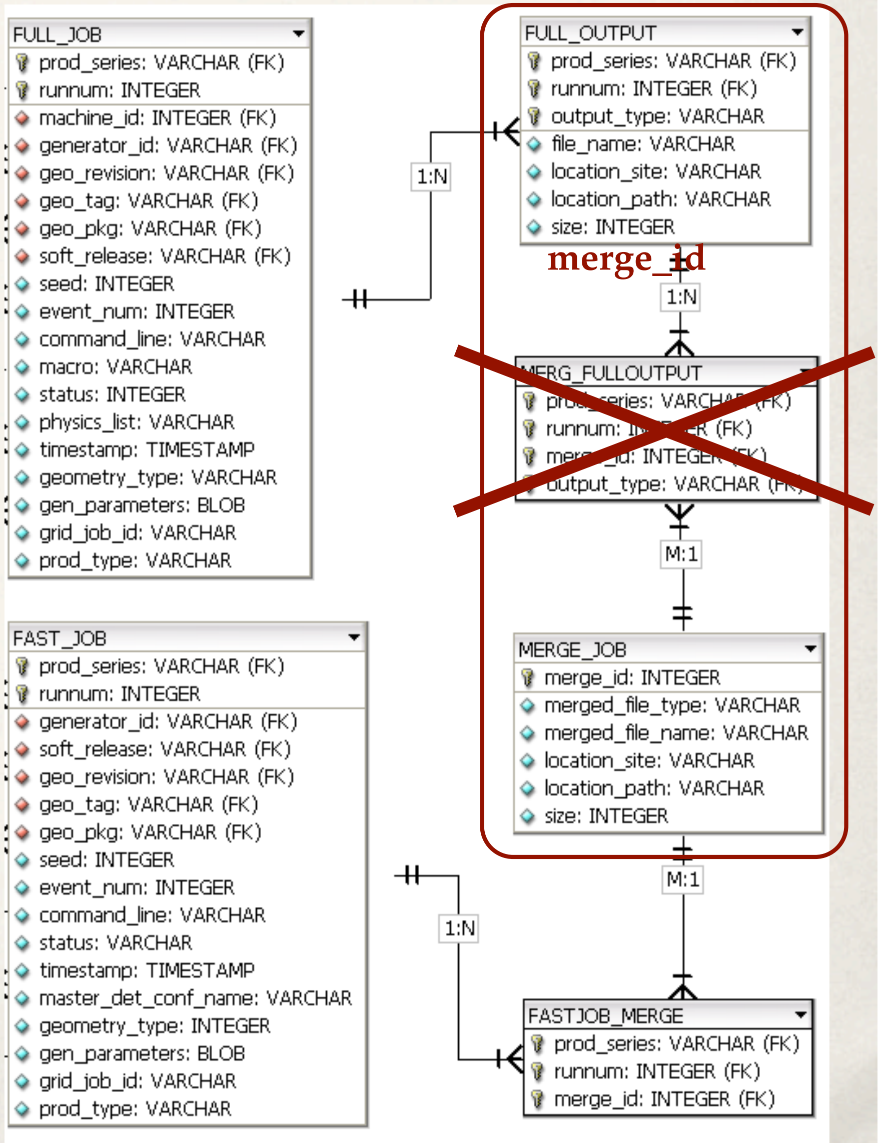
MERG_FULLOUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- merge_id: INTEGER (FK)
- output_type: VARCHAR (FK)

MERGE_JOB
- merge_id: INTEGER
- merged_file_type: VARCHAR
- merged_file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

FASTJOB_MERGE
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- merge_id: INTEGER (FK)

Fast Job I/O – Merge files

- A Merge file consists of many Fullsim Output files (of the same type, i.e. produced by the same generator)
- ~~A Fullsim Ouput file can be used in many Merge files~~
- A Fullsim Ouput file can be used in one and only one Merge files



FULL_JOB
prod_series: VARCHAR (FK)
runnum: INTEGER
machine_id: INTEGER (FK)
generator_id: VARCHAR (FK)
geo_revision: VARCHAR (FK)
geo_tag: VARCHAR (FK)
geo_pkg: VARCHAR (FK)
soft_release: VARCHAR (FK)
seed: INTEGER
event_num: INTEGER
command_line: VARCHAR
macro: VARCHAR
status: INTEGER
physics_list: VARCHAR
timestamp: TIMESTAMP
geometry_type: VARCHAR
gen_parameters: BLOB
grid_job_id: VARCHAR
prod_type: VARCHAR

FULL_OUTPUT
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
output_type: VARCHAR
file_name: VARCHAR
location_site: VARCHAR
location_path: VARCHAR
size: INTEGER

~~MERG_FULLOUTPUT~~
~~prod_series: VARCHAR (FK)~~
~~runnum: INTEGER (FK)~~
~~merge_id: INTEGER (FK)~~
~~output_type: VARCHAR (FK)~~

FAST_JOB
prod_series: VARCHAR (FK)
runnum: INTEGER
generator_id: VARCHAR (FK)
soft_release: VARCHAR (FK)
geo_revision: VARCHAR (FK)
geo_tag: VARCHAR (FK)
geo_pkg: VARCHAR (FK)
seed: INTEGER
event_num: INTEGER
command_line: VARCHAR
status: VARCHAR
timestamp: TIMESTAMP
master_det_conf_name: VARCHAR
geometry_type: INTEGER
gen_parameters: BLOB
grid_job_id: VARCHAR
prod_type: VARCHAR

MERGE_JOB
merge_id: INTEGER
merged_file_type: VARCHAR
merged_file_name: VARCHAR
location_site: VARCHAR
location_path: VARCHAR
size: INTEGER

FASTJOB_MERGE
prod_series: VARCHAR (FK)
runnum: INTEGER (FK)
merge_id: INTEGER (FK)

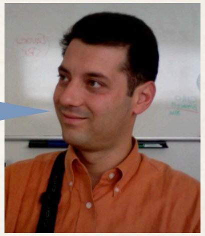
Fast Jo

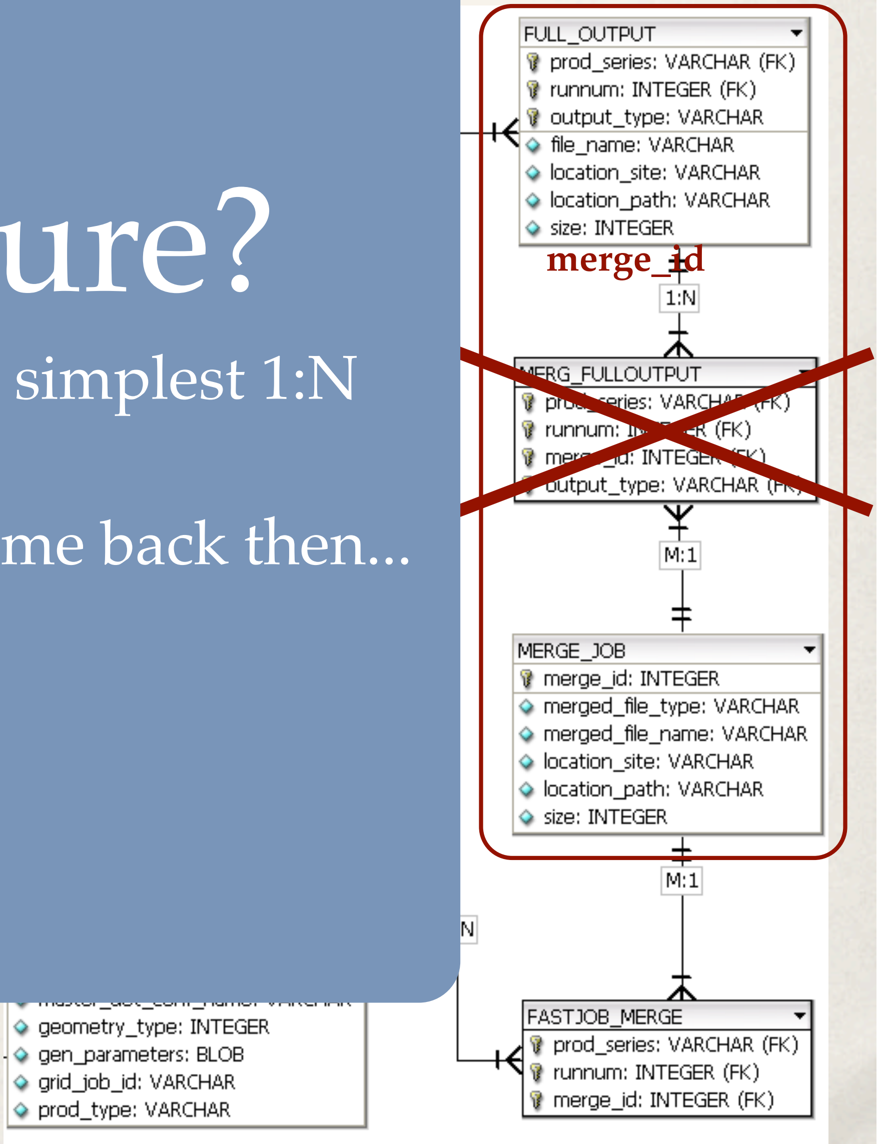
- A Merge fi Fullsim Ou (of the sam by the sam
- ~~A Fullsim (used in ma~~
- A Fullsim (used in on files

Are You sure?

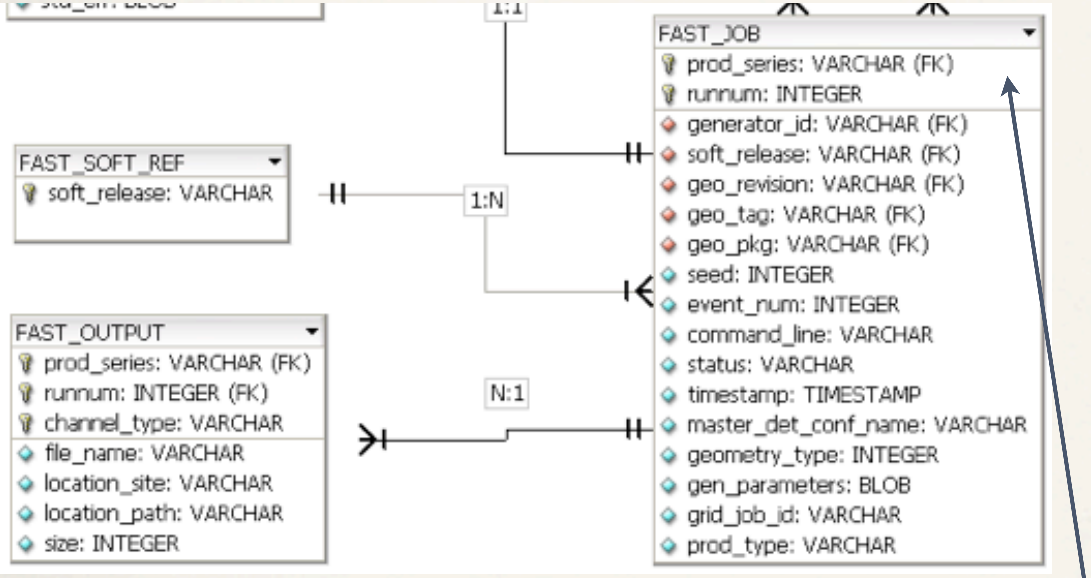
The N:M case includes the simplest 1:N

It will be more difficult to come back then...





Fast Job I/O – Output files



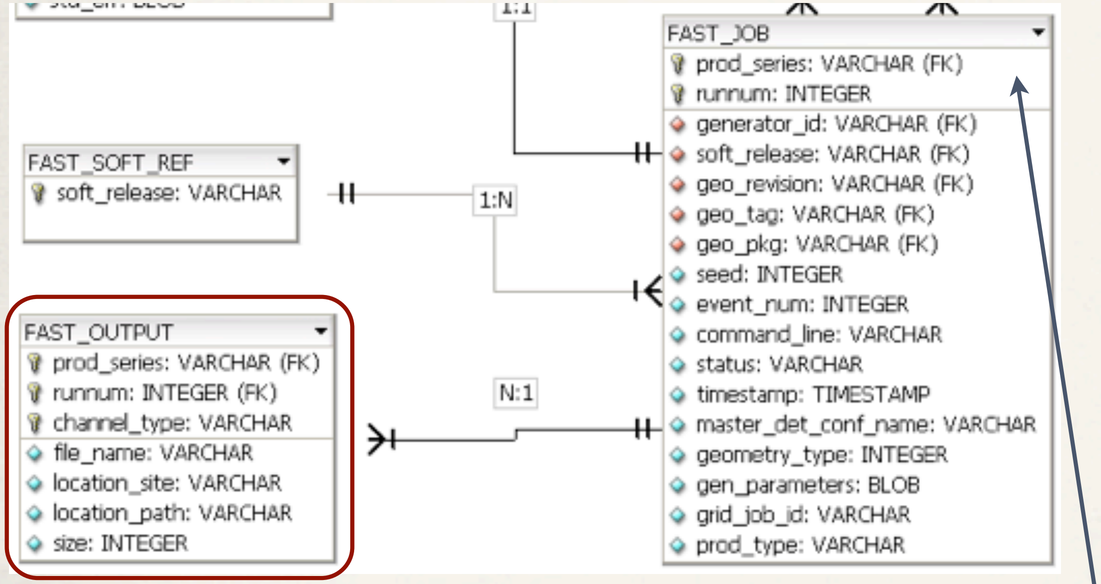
FAST_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- generator_id: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- status: VARCHAR
- timestamp: TIMESTAMP
- master_det_conf_name: VARCHAR
- geometry_type: INTEGER
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

FAST_SOFT_REF
- soft_release: VARCHAR

FAST_OUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- channel_type: VARCHAR
- file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

- Output files:
  - Many per job (1,N), *at most one per physical channel*
  - A given output file is produced by one and only one job
- One table:
  - `FAST_OUTPUT`
    `channel_type, prod_series, runnum`

Fast Job I/O – Output files



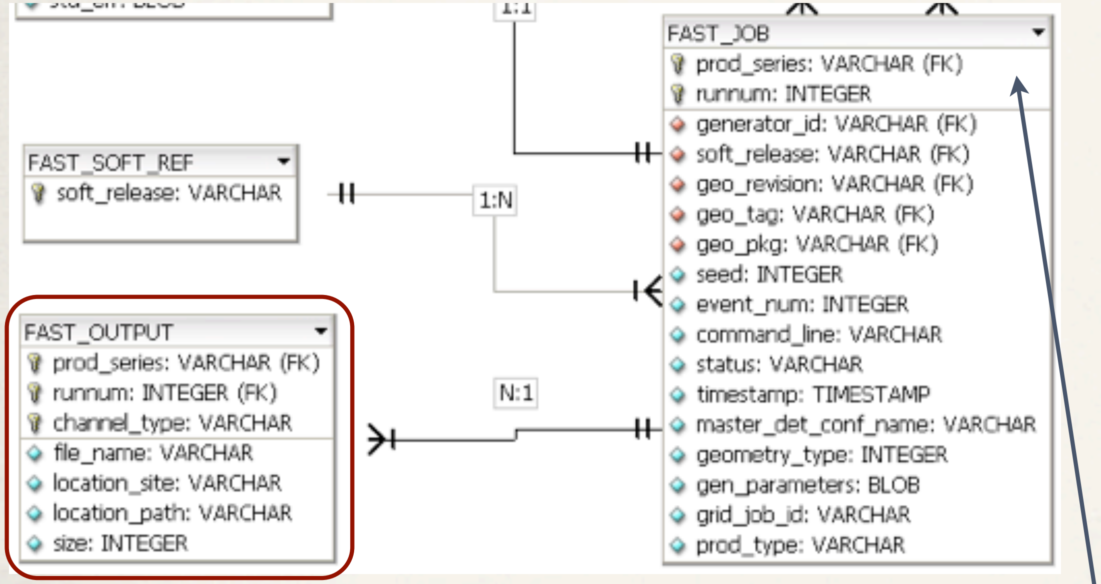
FAST_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- generator_id: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- status: VARCHAR
- timestamp: TIMESTAMP
- master_det_conf_name: VARCHAR
- geometry_type: INTEGER
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

FAST_SOFT_REF
- soft_release: VARCHAR

FAST_OUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- channel_type: VARCHAR
- file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

- Output files:
  - Many per job
    (1,N), *at most one per physical channel*
  - A given output file is produced by one and only one job
- One table:
  - `FAST_OUTPUT`
    `channel_type, prod_series, runnum`

Fast Job I/O – Output files



FAST_JOB
- prod_series: VARCHAR (FK)
- runnum: INTEGER
- generator_id: VARCHAR (FK)
- soft_release: VARCHAR (FK)
- geo_revision: VARCHAR (FK)
- geo_tag: VARCHAR (FK)
- geo_pkg: VARCHAR (FK)
- seed: INTEGER
- event_num: INTEGER
- command_line: VARCHAR
- status: VARCHAR
- timestamp: TIMESTAMP
- master_det_conf_name: VARCHAR
- geometry_type: INTEGER
- gen_parameters: BLOB
- grid_job_id: VARCHAR
- prod_type: VARCHAR

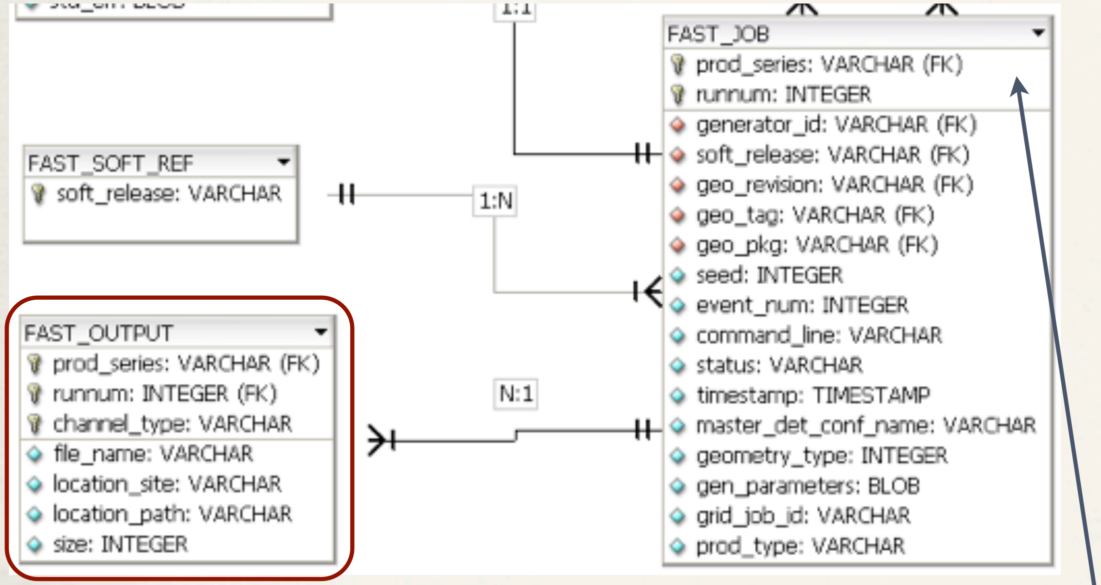
FAST_SOFT_REF
- soft_release: VARCHAR

FAST_OUTPUT
- prod_series: VARCHAR (FK)
- runnum: INTEGER (FK)
- channel_type: VARCHAR
- file_name: VARCHAR
- location_site: VARCHAR
- location_path: VARCHAR
- size: INTEGER

- Output files:
  - Many per job (1,N), *at most one per physical channel*
  - A given output file is produced by one and only one job
- One table:
  - `FAST_OUTPUT`
    `channel_type, prod_series, runnum`

physical channels
naming, list, ...

Fast Job I/O – Output files



- Output files:
  - Many per job (1,N), *at most one per physical channel*
  - A given output file is produced by one and only one job
- One table:
  - `FAST_OUTPUT`
    `channel_type, prod_series, runnum`

Software Release

- Packaging of Fullsim and Production will move to a release based schema (Fastsim already is)

- Geometry packages are “inside” the releases;
self-consistent packages independent from the sim pkg.

- We just need to know which release has been used (for fast/full)
⇒ geometry package determined

- At the moment (no release, yet) is necessary to specify which geometry package has been used on a per job basis.

Open Questions

Generators

- We have many (possible) generators
- A job uses one and only one generator
- A generator has a set of parameters (and their values can vary)
- The generator's parameters values are job dependent

➡ `gen_parameters` is in the Job table

➡ we may store default values in the Generator table
and when needed the new values in the Job table

Machine / Generators / Input Files

- Background's simulation with FullSim is already modeled
- Something can be improved in the logic and/or schema in order to reduce redundancy
  - mad_output_file in the Machine table
  - Input files for the Fullsim
- Discussion with Andrea Di Simone & Manuela Boscolo in progress

Uniqueness

- `prod_series` ➟ identifier of a Production:
  who's giving names? Besides strictly checking it, how to avoid duplicates?

- `runnum` ➟ identifier of a Job within a Production:
  who assign it? Cross numbering between Full & Fast, yes or no?

- `full_input_id` ➟ identifier of an input file for Fullsim
  autoincrement, time-related? only if db centralized; ?

- `merge_id` ➟ identifier of a merged file
  how to assign it? merge job doesn't exist yet... we must take care of it

First Implementation & Queries

MySQL testing

- The presented Relational Schema has been implemented with MySQL RDBMS at Ferrara [Cinzia Luzzi made the job!]

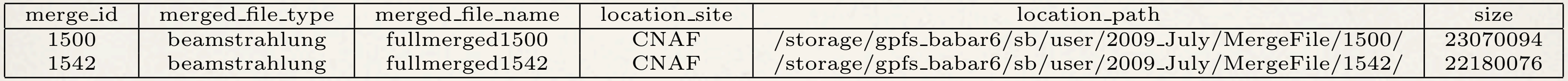
- Scripts have been used to populate the database with data “taken” from previous production (July test)

- Queries of interest have been developed and executed on the schema as a functionality test

Queries

- Retrieve all merge files used by a specific Fastsim Job
  (so we must identify the job by its runnum and prod_series)

```
SELECT MERGE_JOB.* FROM MERGE_JOB NATURAL JOIN FASTJOB_MERGE
NATURAL JOIN FAST_JOB AS FAST WHERE FAST.runnum = 1021 AND
FAST.prod_series = ‘2009_July’;
```

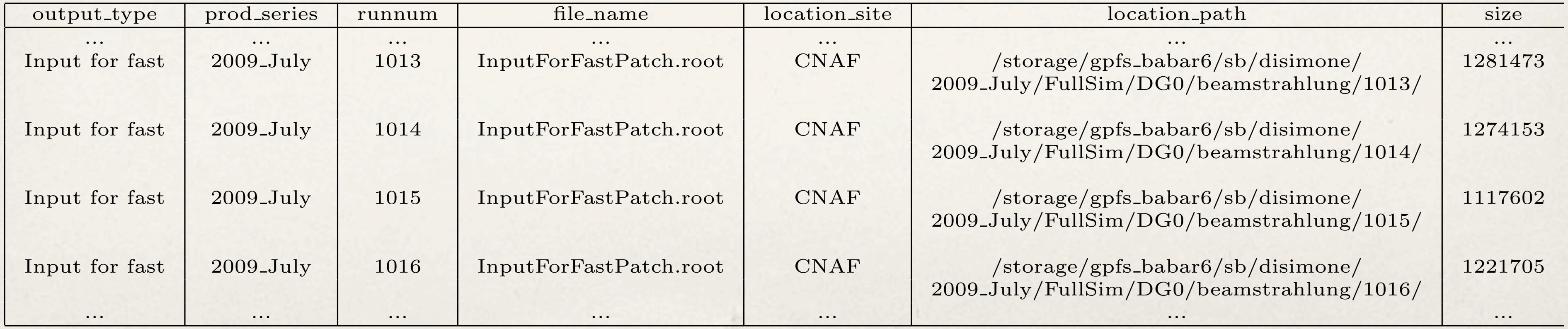


merge_id	merged_file_type	merged_file_name	location_site	location_path	size
1500	beamstrahlung	fullmerged1500	CNAF	/storage/gpfs_babar6/sb/user/2009_July/MergeFile/1500/	23070094
1542	beamstrahlung	fullmerged1542	CNAF	/storage/gpfs_babar6/sb/user/2009_July/MergeFile/1542/	22180076

Queries

- Retrieve all Fullsim Output files used (via merge files) by a specific Fastsim Job

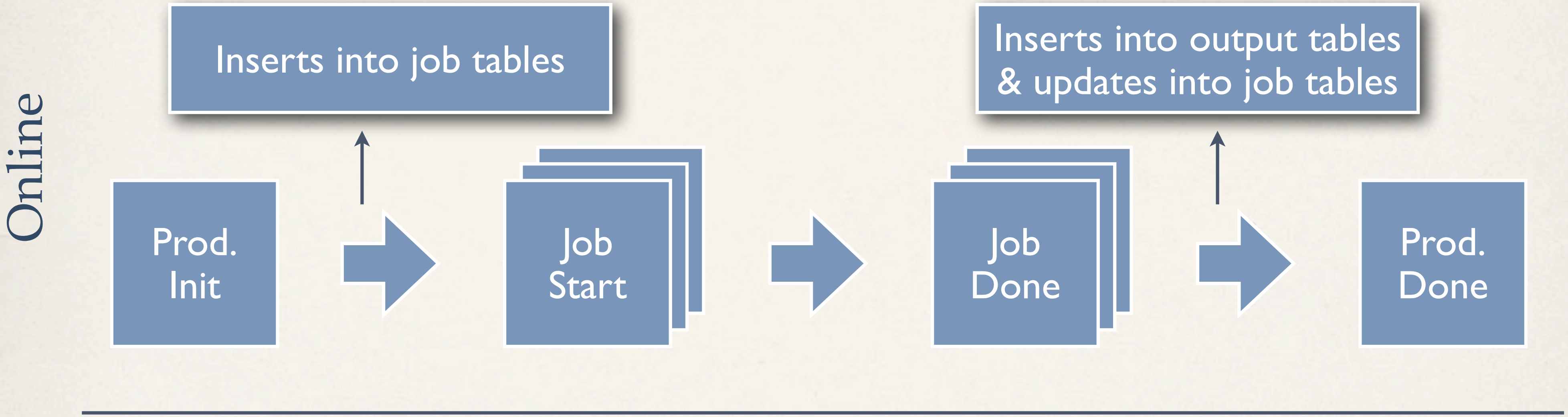
```
SELECT FO.* FROM MERGE_JOB AS M, MERGE_FULLOUTPUT AS MFO,
FULL_OUTPUT AS FO, FASTJOB_MERGE AS FJM, FAST_JOB AS F WHERE
M.merge_id = MFO.merge_id AND MFO.runnum = FO.runnum AND
MFO.prod_series = FO.prod_series AND MFO.output_type =
FO.output_type AND M.merge_id = FJM.merge_id AND FJM.runnum =
F.runnum AND FJM.prod_series = F.prod_series AND F.runnum = 1200
AND F.prod_series = ‘2009_July’;
```



output_type	prod_series	runnum	file_name	location_site	location_path	size
...	...	...	...	...	...	...
Input for fast	2009_July	1013	InputForFastPatch.root	CNAF	/storage/gpfs_babar6/sb/disimone/2009_July/FullSim/DG0/beamstrahlung/1013/	1281473
Input for fast	2009_July	1014	InputForFastPatch.root	CNAF	/storage/gpfs_babar6/sb/disimone/2009_July/FullSim/DG0/beamstrahlung/1014/	1274153
Input for fast	2009_July	1015	InputForFastPatch.root	CNAF	/storage/gpfs_babar6/sb/disimone/2009_July/FullSim/DG0/beamstrahlung/1015/	1117602
Input for fast	2009_July	1016	InputForFastPatch.root	CNAF	/storage/gpfs_babar6/sb/disimone/2009_July/FullSim/DG0/beamstrahlung/1016/	1221705
...	...	...	...	...	...	...

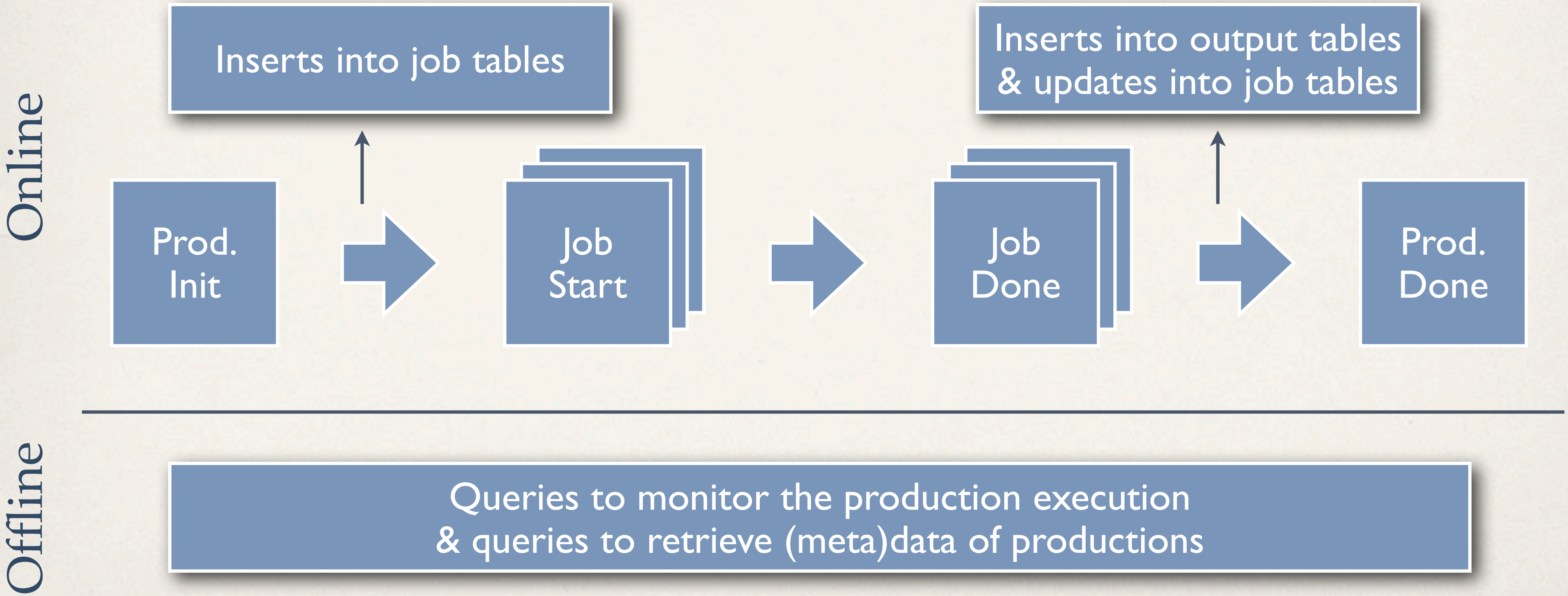
Queries

- Queries of that type have to be included in the production software layer in order to populate the database and monitor the production



Queries

- Queries of that type have to be included in the production software layer in order to populate the database and monitor the production



Online

Inserts into job tables

Inserts into output tables & updates into job tables

Prod. Init

Job Start

Job Done

Prod. Done

Offline

Queries to monitor the production execution & queries to retrieve (meta)data of productions

Future Developments

Deployment at CNAF

- Schema will be deployed at CNAF in the next few weeks
- Basic Web interface

  offline

  - A prototype of production monitor
  - parametric queries to retrieve (meta)data of productions
  - Mysql + PHP + Apache
- Production Web-UI

  online

  - Web form + PHP
  - Executes the production initialization
  - Provides the interface to the database

Deployment at CNAF

- Schema will be deployed at CNAF in the next few weeks

offline

- Basic Web interface
  - A prototype of production monitor
  - parametric queries to retrieve (meta)data of productions
  - Mysql + PHP + Apache

it may be postponed

online

- Production Web-UI
  - Web form + PHP
  - Executes the production initialization
  - Provides the interface to the database

we are already going towards
a distributed production software

Distributed Production

Works to do

[where the Database is involved]

Web-UI

- Production Initialization Script
  - Web form to be used by the production manager
  - Strict check on user input:
    - Production data (e.g. prod_series, prod_root, prod_software)
    - Production type, Number of jobs
    - Jobs data (e.g. seeds, generators, number of events, geometry_type, input files, ...)
    - TAG, ARCH, RELEASE_WORKDIR, ...
  - Mysql + PHP + Apache at CNAF
  - It will populate the database init
  - It will provide a macro for jobs submission (GANGA) start

- Pre- & Post- Job Scripts
  - They take care of updates into the database before and after job execution
    - status changes, ...
    - output files metadata
  - HTTP based service
  - Exception detection
  - Python

Works to do

[where the Database is involved]

Fullsim should be ok
Fastsim has inputs "hard coded"
It will be necessary to separate things

Web-UI

- Production Initialization Script
  - Web form to be used by the production manager
  - Strict check on user input:
    - Production data (e.g. prod_series, prod_root, prod_software)
    - Production type, Number of jobs
    - Jobs data (e.g. seeds, generators, number of events, geometry_type, input files, ...)
    - TAG, ARCH, RELEASE_WORKDIR, ...
  - Mysql + PHP + Apache at CNAF
  - It will populate the database init
  - It will provide a macro for jobs submission (GANGA) start

- Pre- & Post- Job Scripts
  - They take care of updates into the database before and after job execution
    - status changes, ...
    - output files metadata
  - HTTP based service
  - Exception detection
  - Python

Works to do

[where the Database is involved]

Web-UI

- Production Initialization Script
  - Web form to be used by the production manager
  - Strict check on user input:
    - Production data (e.g. prod_series, prod_root, prod_software)
    - Production type, Number of jobs
    - Jobs data (e.g. seeds, generators, number of events, geometry_type, input files, ...)
    - TAG, ARCH, RELEASE_WORKDIR, ...
  - Mysql + PHP + Apache at CNAF
  - It will populate the database init
  - It will provide a macro for jobs submission (GANGA) start

- Pre- & Post- Job Scripts
  - They take care of updates into the database before and after job execution
    - status changes, ...
    - output files metadata
  - HTTP based service
  - Exception detection
  - Python

Works to do

[where the Database is involved]

Web-UI

- Production Initialization Script
  - Web form to be used by the production manager
  - Strict check on user input:
    - Production data (e.g. prod_series, prod_root, prod_software)
    - Production type, Number of jobs
    - Jobs data (e.g. seeds, generators, number of events, geometry_type, input files, ...)
    - TAG, ARCH, RELEASE_WORKDIR, ...
  - Mysql + PHP + Apache at CNAF
  - It will populate the database init
  - It will provide a macro for jobs start submission (GANGA)

- Pre- & Post- Job Scripts
  - They take care of updates into the database before and after job execution
    - status changes, ...
    - output files metadata
  - HTTP based service
  - Exception detection
  - Python

See Armando’s talk for details

Conclusions

- Database schema is ready, validated, implemented and tested
- Some (minor) refinements are under discussion
- Deployment will be ready by the end of October
- Interactions with Production Software have been modeled (see Armando's talk)
- Coding will start soon...
- ...ready for January 2010 production!