

# Proposal for restoring foreign keys to pipeline database tables

On June 17th, the foreign keys from all core pipeline tables were lost. In order to restore and maintain referential integrity between parent and child tables, these keys need to be restored. Any dangling child records (orphaned when a parent record was deleted due to the absence of the foreign-key cascade-delete mechanism) need to be identified and removed before the foreign keys can be built. Many of the missing foreign keys have already been built, and for those that are still missing, the foreign key fields remain indexed, so the process of creating the keys will be reasonably fast.

**New:** I consulted with Ian MacGreggor (the SLAC DBA) and he recommended a way of re-introducing foreign keys that does not require table locks and therefore does not strictly require a pipeline outage. It also does not require any special handling of the Log table. I have updated the SQL code snippets, below, to reflect this new method. Here is an outline of the new method:

1. Create foreign key constraints, enabled but not validated. This requires no locking of the target table (or rows in the master) to validate the existing data. However, it does validate any new data inserted to the table; ie: It immediately prevents further orphaned records.
2. Examine, and clean up existing, orphaned records. This only requires row-level locking of the records to be removed.
3. Modify foreign key constraints and enable full-validation. This requires no locking.

Since no locking of "ACTIVE" records is required, none of the steps above will interfere with pipeline operation. This means the pipeline can be restarted even if the clean-up process has not completed.

See "Oracle Database SQL Reference" (10gR2 version) pages 8-14 - 8-16 and "Oracle Administrator's Guide" (10gR2 version) pages 13-9 - 13-11 for information on the method and details of why concurrent activity is permitted.

**Note:** I have already performed step 1, creating the constraints in "novalidate" mode.

## To Do List:

### The following constraints need to be created:

- Log.ProcessInstance (on delete of parent, ->NULL)
- DatasetVersion.ProcessInstance (on delete of parent, ->NULL)
- ProcessInstance.Stream (cascade delete)
- Process.Task (cascade delete)
- Stream
  - .Task (cascade delete)
  - .ParentStream (cascade delete)
- Task.ParentTask (cascade delete)

## SQL Code:

### Log.ProcessInstance

This guy needs an "on delete set null" foreign key constraint:

```

alter table Log
  add constraint FK_Log_PI
  foreign key (ProcessInstance)
  references ProcessInstance (ProcessInstance)
  on delete set NULL
  enable novalidate;

select count(*) from Log;
select count(*) from Log L
  where not exists (select 1 from ProcessInstance PI where PI.ProcessInstance = L.ProcessInstance);

-- sanity check: Make sure the second count is much smaller than the first!

update Log L
  set ProcessInstance=NULL
  where not exists (select 1 from ProcessInstance PI where PI.ProcessInstance = L.ProcessInstance);

-- sanity check: make sure number of records updated is same as the 2nd count from above before committing

commit;

alter table Log
  modify constraint FK_Log_PI
  validate;

```

## DatasetVersion.ProcessInstance

This guy needs an "on delete set null" foreign key constraint:

```

alter table DatasetVersion
  add constraint FK_DSV_PI
  foreign key (ProcessInstance)
  references ProcessInstance (ProcessInstance)
  on delete set NULL
  enable novalidate;

select count(*) from DatasetVersion;
select count(*) from DatasetVersion DSV
  where not exists (select 1 from ProcessInstance PI where PI.ProcessInstance = DSV.ProcessInstance);

-- sanity check: Make sure the second count is much smaller than the first!

update DatasetVersion DSV
  set ProcessInstance=NULL
  where not exists (select 1 from ProcessInstance PI where PI.ProcessInstance = DSV.ProcessInstance);

-- sanity check: make sure number of records updated is same as the 2nd count from above before committing

commit;

alter table DatasetVersion
  modify constraint FK_DSV_PI
  validate;

```

## Processinstance.Stream

This guy needs an "on delete cascade" foreign key constraint:

```

alter table ProcessInstance
  add constraint FK_PI_Stream
  foreign key (Stream)
  references Stream (Stream)
  on delete cascade
  enable novalidate;

select count(*) from ProcessInstance;
select count(*) from ProcessInstance PI
  where not exists (select 1 from Stream S where S.Stream = PI.Stream);

-- sanity check:  Make sure the second count is much smaller than the first!

delete from ProcessInstance PI
  where not exists (select 1 from Stream S where S.Stream = PI.Stream);

-- sanity check:  make sure number of records updated is same as the 2nd count from above before committing
commit;

alter table ProcessInstance
  modify constraint FK_PI_Stream
  validate;

```

## Process.Task

This guy needs an "on delete cascade" foreign key constraint:

```

alter table Process
  add constraint FK_Process_Task
  foreign key (Task)
  references Task (Task)
  on delete cascade
  enable novalidate;

select count(*) from Process;
select count(*) from Process P
  where not exists (select 1 from Task T where T.Task = P.Task);

-- sanity check:  Make sure the second count is much smaller than the first!

delete from Process P
  where not exists (select 1 from Task T where T.Task = P.Task);

-- sanity check:  make sure number of records updated is same as the 2nd count from above before committing
commit;

alter table Process
  modify constraint FK_Process_Task
  validate;

```

## Stream.Task

This guy needs an "on delete cascade" foreign key constraint:

```

alter table Stream
  add constraint FK_Stream_Task
  foreign key (Task)
  references Task (Task)
  on delete cascade
  enable novalidate;

select count(*) from Stream;
select count(*) from Stream S
  where not exists (select 1 from Task T where T.Task = S.Task);

-- sanity check: Make sure the second count is much smaller than the first!

delete from Stream S
  where not exists (select 1 from Task T where T.Task = S.Task);

-- sanity check: make sure number of records updated is same as the 2nd count from above before committing
commit;

alter table Stream
  modify constraint FK_Stream_Task
  validate;

```

## Stream.ParentStream

This guy needs an "on delete cascade" foreign key constraint:

```

alter table Stream
  add constraint FK_Stream_ParentStream
  foreign key (ParentStream)
  references Stream (Stream)
  on delete cascade
  enable novalidate;

select count(*) from Stream;

select count(*) from Stream S1
  where not exists (select 1 from Stream S2 where S1.ParentStream = S2.Stream);

select count(*) from Stream S1
  where Stream != 0
  and not exists (select 1 from Stream S2 where S1.ParentStream = S2.Stream);

-- sanity check: Make sure the second count is much smaller than the first!

delete from Stream S1
  where Stream != 0
  and not exists (select 1 from Stream S2 where S1.ParentStream = S2.Stream);

-- sanity check: make sure number of records updated is same as the 2nd count from above before committing

-- Make sure we didn't delete the dummy root:
select count(*) from Stream;

commit;

alter table Stream
  modify constraint FK_Stream_ParentStream
  validate;

```

## Task.ParentTask

This guy needs an "on delete cascade" foreign key constraint:

```

alter table Task
  add constraint FK_Task_ParentTask
  foreign key (ParentTask)
  references Task (Task)
  on delete cascade
  enable novalidate;

select count(*) from Task;

select count(*) from Task T1
  where not exists (select 1 from Task T2 where T1.ParentTask = T2.Task);

select count(*) from Task T1
  where Task != 0
    and not exists (select 1 from Task T2 where T1.ParentTask = T2.Task);

-- sanity check: Make sure the second count is much smaller than the first!

delete from Task T1
  where Task != 0
    and not exists (select 1 from Task T2 where T1.ParentTask = T2.Task);

-- sanity check: make sure number of records updated is same as the 2nd count from above before committing

-- Make sure we didn't delete the dummy root:
select count(*) from Task;

commit;

alter table Task
  modify constraint FK_Task_ParentTask
  validate;

```