[Xray DC] Lost of Pre-condition links on Xray 7.x

Before performing these actions, we recommend creating a backup of the table AO_8B1069_TEST_PRE_COND_LINK at least, and ideally, all the database.

Xray 7. x introduced a major bug (

~~XRAY-9359~~ - When having two or more preconditions linked to the same test, integrity checker will return invalid links and removed all preconditions from the test CLOSED

) that would result in the loss of precondition links in specific conditions.

To recover the majority of the links, we'll be using the mementos from past test runs created.

E.g Test ta was linked to Precond pa, a user created a Test Run, and this data is saved in the Test Run.

This solution is not perfect and will not consider some situations:

- Scenario A (no solution other than each user analyzing this information)
  - A Test **ta** was linked to a Pre-condition **pa**.
  - A Test Run was created.
  - The link has been intentionally removed, and the Test Run data was never reset or merged.
  - This query will re-insert the link.
- Scenario B (a second query will be provided to have the list of links to be fixed)
  - A Test **ta** was linked to a Pre-condition **pa**
  - A Test Run was created
  - The **ta's** test type or **pa's** precondition type changed have been changed
  - This query will re-insert the link

Based on the Test Runs data, the following query will insert the missing links into the Test Pre-condition link table. We recommend executing only the select first to validate the data and add more filters in case. you would like to recover the precondition links by project, date of test run or other.

```
  INSERT INTO "AO_8B1069_TEST_PRE_COND_LINK" ("CREATED_DATE", "RANK", "SOURCE_ISSUE", "TARGET_ISSUE",
"TEST_VERSION_ID")
  select distinct CURRENT_DATE, rank, source_issue, target_issue, version_id from (
  select ji.id as target_issue,concat(p.pkey, '-', ji.issuenum) as key, pr."RANK" as rank, pr."TEST_RUN_ID",
tr."TEST_ISSUE_ID" as source_issue, tv."ID" as version_id
  from "AO_8B1069_PRE_COND_RESULT" pr, jiraissue ji, project p, "AO_8B1069_TEST_RUN" tr,
"AO_8B1069_TEST_VERSION" tv, issuetype it
  where ji.project = p.id and concat(p.pkey, '-', ji.issuenum) = pr."KEY" and tr."ID" = PR."TEST_RUN_ID"
          and tv."ENTITY_ID" = tr."TEST_ISSUE_ID" and tv."IS_DEFAULT" = true and it.id=ji.issuetype and it.id
in (SELECT propertyvalue FROM PROPERTYENTRY pe
        LEFT JOIN PROPERTYSTRING string on string.id = pe.id
        WHERE property_key like '%issue-type.pre-condition%')) as precondata
          WHERE  (source_issue, target_issue) not in (SELECT "SOURCE_ISSUE", "TARGET_ISSUE" FROM
"AO_8B1069_TEST_PRE_COND_LINK");
```

This query helps to find the tests linked to preconditions of different types; Please be aware that Cucumber types can be called behave, automated [cucumber], etc.
The cleaning up should be done manually on Jira based on the ids listed on the below query.

```
select pt."ID", pt."SOURCE_ISSUE" as test_id, pt."TARGET_ISSUE" as precond_id, CFO.CUSTOMVALUE as precondtype,
CFO2.customvalue as test_type
from "AO_8B1069_TEST_PRE_COND_LINK" pt
join customfieldvalue cfv on cfv.issue= pt."TARGET_ISSUE"
join customfield cf on cfv.customfield= cf.id and cf.CUSTOMFIELDTYPEKEY like '%com.xpandit.plugins.xray:
precondition-test-type-custom-field%'
JOIN customfieldoption CFO ON CFO.ID=CAST (cfv.stringvalue AS INTEGER)
join customfieldvalue cfv2 on cfv2.issue= pt."SOURCE_ISSUE"
join customfield cf2 on cfv2.customfield= cf2.id and cf2.CUSTOMFIELDTYPEKEY like '%com.xpandit.plugins.xray:
test-type-custom-field%'
JOIN customfieldoption CFO2 ON CFO2.ID=CAST (cfv2.stringvalue AS INTEGER)
where CFO.CUSTOMVALUE <> CFO2.customvalue;
```

These queries are built for PostgreSQL.

The resolution for this was implemented starting from version **7.1.1 and subsequent releases**.